Computer Science and Artificial Intelligence Laboratory

Technical Report

MIT-CSAIL-TR-2005-016
AIM-2005-008

March 2, 2005

Combining Object and Feature Dynamics in Probabilistic Tracking

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

massachusetts institute of technology, cambridge, ma 02139 usa — www.csail.mit.edu

Combining Object and Feature Dynamics in Probabilistic Tracking

Leonid Taycher, John W. Fisher III, and Trevor Darrell
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA, 02139
{lodrion,fisher,trevor}@csail.mit.edu

Abstract

Objects can exhibit different dynamics at different 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 entire 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—but constraints from the global model can make them more robust. We propose a probabilistic framework for incorporating global dynamics knowledge into the local feature extraction processes. A global tracking algorithm can be formulated as a generative model and used to predict *feature values that influence the observation process of the feature extractor. We combine such models in a multichain graphical model framework. We show the utility of our framework for improving feature tracking and thus 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 problems such as background subtraction, structure from motion and articulated body tracking.*

1 Introduction

Motion analysis algorithms are often structured in a multistage fashion, with each stage operating at a particular spatio-temporal scale and exploiting different model of scene dynamics. 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. It would then use these features to determine poses of the objects moving in the scene, and analyze object interaction based on the individual object poses. Low-level methods account for local scene dynamics while ignoring global spatial coherence. High-level algorithms use models that are often too coarse (and/or approximate) for local motion estimation, but take into account global spatial relationships. 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.

Low-level algorithms ignore global spatial relationships by modeling the evolution of each image patch (in feature extraction [16, 18]) or object (in object tracking [13]) as independent, 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 the foreground object does not remain stationary for extended periods of time. When these assumptions are violated, the resulting errors (e.g. so-called "sleeping man problem", Figure 1), are passed to higher-level modules, and these are not always able to correct them. Furthermore, 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-hoc fashion and do not have a sound probabilistic interpretation—e.g., the uncertainty information is propagated only one way, from low- to high-level models.

The need to incorporate a feedback mechanism into multistage systems has long been recognized [14, 1]. There are three important 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 of course critical to have correct propagation of uncertainty from high- to low-level processing.

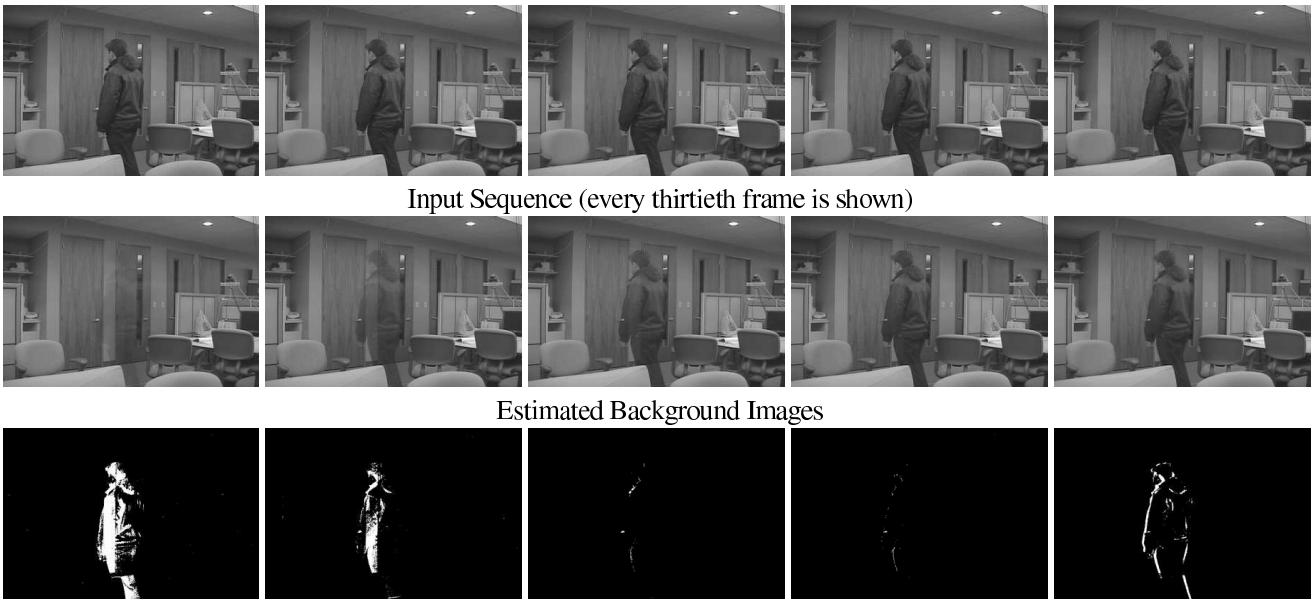


Figure 1: An example of the "sleeping man" problem in adaptive background subtraction. Adaptive background maintenance systems make an implicit assumption that foreground objects do not remain stationary. When this is not the case (as in the sequence shown in the top row), the background model (middle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

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 we, with some abuse of terminology, refer to as "features"). The features (e.g., foreground labels or individual object positions) are *latent* variables in the lower-level module, but are *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) [2].

The resulting framework may be thought of as performing probabilistic model-based regularization of low-level algorithms, similar to deterministic model-based regularization of [1]. Since methods operating at different levels are coupled only through 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 compute not only a feature value but also an uncertainty about the measurement. For example, dissimilarity computations [15] and Kalman filtering [14] 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 multihypothesis tracking [4] and probabilistic data association filters [5] can be used to disambiguate feature tracks. These methods are quite computationally intensive and cannot correct feature drift or deal with missing observations.

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

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

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 [11]. 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 [20, 8], which jointly model foreground and background processes.

The framework proposed in this paper is most closely related to intermediate integration approaches of [12] and [10]. These methods update both global and local models based on the feature match deterministically selected 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 [19] that use a single dynamical model to interpret multiple streams of observations.

We combine individual models by using the product system proposed in [7] and [2]. Our framework uses renormalized products of tractable probability distributions to model data that satisfies constraints arising from different models

The main difference between the resulting multichain model and PoHMMs [2] is that individual chains 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 adaptive background subtraction.

In this work, we assume that constituent stochastic trackers are completely specified, and that the appearance feature hierarchy (if any) is known; we concern ourselves with inference on the combined model, rather than learning its structure.

3 Developing a Dual-chain Model

As has been alluded earlier, 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 Figure 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 the consequence of the independence assumption, state prediction, $p(R^t|R^{t-1})$, and the prior over features, $p(F^t|I^{0..t-1}) = \int dR^t p(F^t|R^t) \int dR^{t-1} p(R^t|R^{t-1}) p(R^{t-1}|I^{0..t-1})$ are overly broad, which makes 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 Figure 2(b).

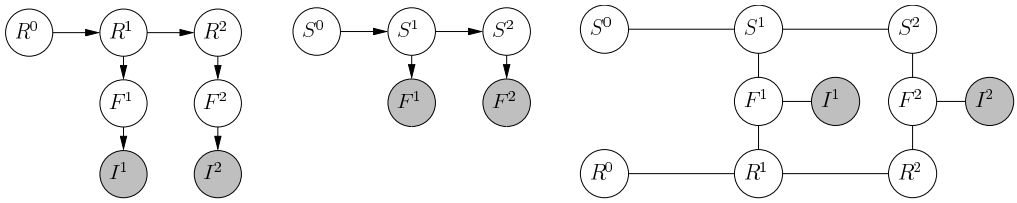


Figure 2: Combining local and global dynamics for object tracking. (a) A generative model used in feature extraction algorithms. The low-level state, $R = \{R_k\}$, evolves according to the local dynamic model, $p(R^{t+1}|R^t) = \prod_k p(R_k^{t+1}|R_k^t)$. At time t, the observed image is based on $p(I^k|F^k)$, 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.).

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 considers features to be directly observed, 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.

Both of the models described above are approximate. 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

$$p(F^t, R^t|I^{0..t-1}) = p(F^t|R^t) \int dR^{t-1} p(R^t|R^{t-1}) p(R^{t-1}|I^{0..t-1}), \tag{1}$$

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

$$p(F^t, R^t|I^{0..t-1}) = q(F^t, R^t; I^{0..t-1}) \int dR^{t-1} p(R^t|R^{t-1}) p(R^{t-1}|I^{0..t-1}) \tag{2}$$
$$q(F^t, R^t; I^{0..t-1}) \neq p(F^t|R^t).$$

That is, when the true dynamics model is used, F^t (and I^t) are independent from prior observations conditioned on R_i^t, but this is not the case for the approximate dynamics actually used. Modeling the dependency between F^t and prior observations that remains unrepresented by feature extraction model would allow for better estimation of the state posterior. We choose the approximation to $q(F^t, R^t; I^{0..t-1})$ that incorporates the information available to the object tracking model via a product

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

This is equivalent to the dual-chain model shown in Figure 2(c), with potentials corresponding to conditional distributions from constituent models (that is, $\phi(S^t, S^{t-1}) = p(S^t|S^{t-1}), \phi(F^t, S^t) = p(F^t|S^t), \phi(I^t, F^t) = p(I^t|F^t)$, 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. And visa 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., reduce the possibility of 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 we discuss 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

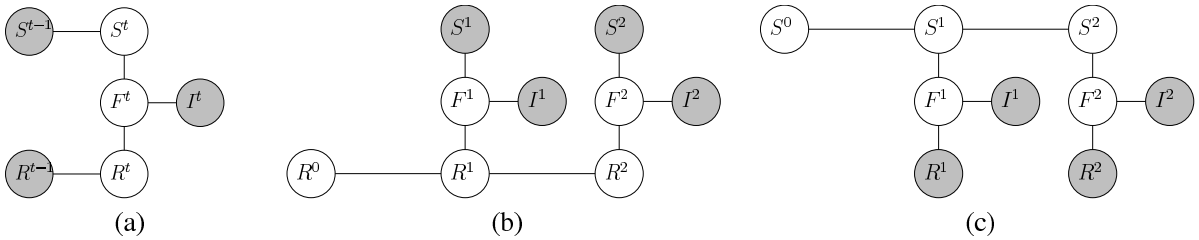


Figure 3: Graph structures used in inference algorithms in the dual-chain model. (a) A tree-shaped subgraph on which a single step of approximate filtering is performed. The marginal distributions, $p(S^{t-1}|I^{0..t-1})$ and $p(R^{t-1}|I^{0..t-1})$, have been computed at the previous iteration, and are not modified; I^t is observed. (b, c) Subgraphs for coordinate ascent in the dual-chain model. By fixing values of states $S^{0..T}$, 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 $R^{0..T}$ are fixed (c) an existing high-level optimization algorithm can be applied.

sets, and the likelihood potentials are extended to produce uniform likelihoods for features that are not part of an original submodel. In general, when the feature sets are disjoint, the model would reduce to a PoHMMs model with noninteracting chains. Since we are interested in combining models that correspond interacting stages of a feed-forward algorithm, this will never be the case.

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 multichain model is loopy (Figure 3(a)), making inference complicated 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 multichain model.

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

$$p(S^1|I^1) = \frac{1}{Z} \int dS^0 \phi(S^1, S^0) p(S^0) \int dF^1 \left[\phi(F^1)\phi(F^1, S^1) \int dR^1 \left[\phi(F^1, R^1) \int dR^0 \phi(R^1, R^0) p(R^0) \right] \right], \quad (4)$$

where $\phi(F^1) \equiv \phi(I^1, F^1)$. The equivalent expression for $p(R^1|I^1)$ is not shown.

Filtering at the next timestep ($t = 2$) is more complex since the model now contains loops and the exact inference would require representing the joint $p(S^1, R^1|I^1)$:

$$p(S^2|I^1, I^2) = \frac{1}{Z} \int dF^2 \left[\phi(F^2)\phi(F^2, S^2) \int dR^2 \left[\phi(F^2, R^2) \int dS^1 dR^1 \phi(S^2, S^1)\phi(R^2, R^1) p(S^1, R^1|I^1) \right] \right]. \quad (5)$$

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 is 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 Equation 5, we obtain an approximate inference equation:

$$p(S^2|I^2) \approx \frac{1}{Z} \int dS^1 \phi(S^2, S^1) p(S^1|I^1) \int dF^2 \left[\phi(F^2)\phi(F^2, S^2) \int dR^2 \left[\phi(F^2, R^2) \int dR^1 \phi(R^2, R^1) p(R^1|I^1) \right] \right]. \quad (6)$$

The similarity between Equations (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 Figure 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 computation of *all* of the marginal distributions.

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

for all $t \geq 0$ **do**

PREDICT the current state of the object and states of individual features, compute messages:

$\mu_{S^{t-1} \to S^t} = \int dS^{t-1} \phi(S^t, S^{t-1}) p(S^{t-1} | I^{0..t-1})$ and

$\mu_{R^{t-1} \to R^t} = \int dR^{t-1} \phi(R^t, R^{t-1}) p(R^{t-1} | I^{0..t-1})$.

ESTIMATE feature distributions based on predicted states and current observations, compute messages:

$\mu_{S^t \to F^t} = \int dS^t \phi(F^t, S^t) \mu_{S^{t-1} \to S^t}$,

$\mu_{R^t \to F^t} = \int dR^t \phi(F^t, R^t) \mu_{R^{t-1} \to R^t}$,

$\mu_{F^t \to S^t} = \int dF^t \mu_{R^t \to F^t} \phi(I^t, F^t)$, and

$\mu_{F^t \to R^t} = \int dF^t \mu_{S^t \to F^t} \phi(I^t, F^t)$.

UPDATE object state using features predicted by feature extractor, and state of the feature extractor using features predicted by object model:

$p(S^t | I^{0..t}) \propto \mu_{S^{t-1} \to S^t} \mu_{F^t \to S^t}$ and

$p(R^t | I^{0..t}) \propto \mu_{R^{t-1} \to R^t} \mu_{F^t \to R^t}$.

end for

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 partially estimation steps of the approximate algorithm; consequently, combining models introduces very little additional complexity to the inference process.

3.2 Batch Optimization in the Multichain 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 Figure 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 $\arg\max_{S^{0..T}} p(F^{1..T} | S^{0..T})$. Once again, the algorithms developed for single-chain models need to be modified to be of use in the dual-chain setting.

We base our optimization approach on a coordinate ascent algorithm that alternates between optimizing one set of states (either $R^{0..T}$ or $S^{0..T}$) while keeping the other one fixed. The dual-chain structure, with latent feature nodes separating states, 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, Figure 3(b, c). The summary of out method is presented in Algorithm 2.

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

4 Applications

We demonstrated the utility of our dual-chain framework in three different domains. Adaptive background subtraction and feature-point tracking are ubiquitous low-level methods used in many high-level motion-analysis algorithms (e.g, [21, 16, 3, 9, 1]) and could benefit from spatial coherence information available to those algorithms. In particular, we demonstrated that the results can be dramatically improved by using a dual-chain formulation combining adaptive background subtraction and multi-object (blob) tracking. Similarly, structure-from-motion estimates in a dual-chain framework that includes a feature-point tracker are superior to those in a feed-forward system. Finally, we presented a dual-chain articulated-body tracking

approach combining rigid 2D head and hand motion model with articulated body dynamics. This method improves upon likelihood-sampling approach [17], and compares favorably with the well-known CONDENSATION algorithm in two ways. First, a monolithic approach using CONDENSATION required a significantly greater number of samples in order to explore the configuration space sufficiently as compared to the multi-chain method. 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.

4.1 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 object 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 (Figure 1).

Common adaptive background algorithms similar to [16] can be represented as inference in a generative model that can then be incorporated into a dual-chain 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 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 Figure 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, in particular, 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 Figure 2(b)) in the dual-chain framework.

In our experiments, we have used an object (blob) tracker with first-order linear dynamics similar to the one described in [16]. In this case high-level S^t contained 2D positions and velocities and appearances of the moving objects. 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 dual-chain algorithm and of the stand-alone background subtraction modules with different values of $P(B_k^t = 1)$.

The systems were evaluated on datasets provided for the PETS 2001 workshop. [1] Algorithms were evaluated as follows: at every frame, we have computed a raw foreground map by thresholding (at 0.5) the background probability value at every pixel, and extracted a set of connected components.

We were interested in three common classes of coarse 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 covered by extracted components; and (3) a foreground component was detected in a location where no moving objects were present. The quantitative comparison results are summarized in Figure 5. Sample frames from the first sequence with corresponding estimated background images and foreground components are shown in Figure 6.

Replacing the feed-forward tracking algorithm with a dual-chain framework did not result in a large performance penalty. In our experiments, the difference between running times of the dual-chain algorithm and the feed-forward system was less than 4%. Partially optimized code on a 2.8GHz workstation was able to achieve 9.6fps for sequential processing and 9.3fps for dual-chain processing on 768×576 images (this time included reading images from the hard drive).

4.2 Structure from Motion Estimation

We have evaluated our batch optimization algorithm by applying it to the problem of extracting structure from motion. We have based our dual-chain variant on the factor-analysis based method of [6]. 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 homogeneus projection matrix at time t as

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

$m^t = (m_1^t, \ldots, m_8^t)$, the noisy projection equations for P points in T frames are written by [6] as

$$\begin{pmatrix} u_1^1 & \ldots & u_P^1 & v_1^1 & \ldots & v_P^1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ u_1^T & \ldots & u_P^T & v_1^T & \ldots & v_P^T \end{pmatrix} = \begin{pmatrix} m_1^1 & \ldots & m_8^1 \\ \vdots & \ddots & \vdots \\ m_1^P & \ldots & m_8^P \end{pmatrix} A + \eta_{[T\times 2P]}, \text{ where} \tag{7}$$

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

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 Figure 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

$$p(S^t|S^{t-1}) = \delta(\mathfrak{S}^t - \mathfrak{S}^{t-1}) N\left(\begin{pmatrix} m^t \\ \dot{m}^t \\ \ddot{m}^t \end{pmatrix}; \begin{pmatrix} \mathbf{1} & \mathbf{1} & \mathbf{\frac{1}{2}} \\ 0 & \mathbf{1} & \mathbf{1} \\ & & \mathbf{1} \end{pmatrix} \begin{pmatrix} m^{t-1} \\ \dot{m}^{t-1} \\ \ddot{m}^{t-1} \end{pmatrix}, \Sigma_\epsilon\right), \tag{8}$$

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

$$p(F^t|S^t) = N(F^t; m^t A, \Sigma_\eta), \tag{9}$$

with A defined in Equation 7. By using a Kalman-filter based feature-point tracker in the low-level chain (that modeled evolution of each (u_i, v_i) feature independently), we have obtained a dual-chain extension of both the pure factor-analysis based algorithm and its 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 distributed 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 5 frames with probability $P(drift)$.

Shapes recovered for $P(loose) = 0.1, P(drift) = 0.3$ are shown in Figure 7. As can be seen, 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 dual-chain 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 Figure 8. Note that the number of occlusions (related to $P(loose)$) 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 $P(loose)$.

The results of applying factor analysis with temporal coherence and its dual-chain variant to a fifty-frame video sequence[2] of a rotating box are shown in Figure 9. The Shape recovered by stand-alone factor analysis contains many spurious points, but the dual-chain framework succeeded in approximately estimating the correct shape.

4.3 Articulated Body Tracking

We have used the 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 choose to use a sample-based density representation. Common sample-based particle-filtering approaches (e.g., CONDENSATION) compute posterior state distribution at each timestep by sampling from the distribution at the previous timestep propagated by dynamics and reweighting samples by their likelihood. If the configuration space is complex, then, unless the dynamics are well known, this procedure results in many samples falling into areas of zero likelihood, and thus increasing the number of samples that need to be drawn. An alternative is *likelihood sampling*[17], when pose samples are drawn from the pose likelihood function and are then reweighted based on the propagated prior. Although this method results in greater per-sample complexity, it enables us to use fewer samples, since they are in general 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 sample from observation likelihood definitions for the head and hands (in this case mixtures of Gaussians corresponding to face detector outputs and to 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 has 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 sample from the a pose likelihood.

We define our proposal distribution as in [17]. 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 [17].

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 dual-chain model.

We have applied our dual-chain tracker to three sample sequences, with results shown in Figures 10, 11, and 12. For each frame in the sequence, we have rendered forty 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 Figure 13, we compare the performance of the enhanced dual-chain tracker using 1,000 samples per frame (first column), likelihood-sampling tracker using 1,000 samples (second column), CONDENSATION tracker with 5,000 samples that runs as fast as the dual-chain tracker (third column), and finally CONDENSATION tracker with 15,000 samples (the smallest number of samples that enables CONDENSATION to perform with accuracy approaching dual-chain tracker performance). The results are presented using the same method as in Figure 10, the frontal view is shown overlayed on top of the input image, with the side view below.

The dual-chain 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 22, 60, etc.). The CONDENSATION variant with 5,000 samples failed after 30 frames due partly 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 more of the sequence.

[2] We used part of an original sequence from http://www.cs.ucla.edu/~hljin/research/voi.html

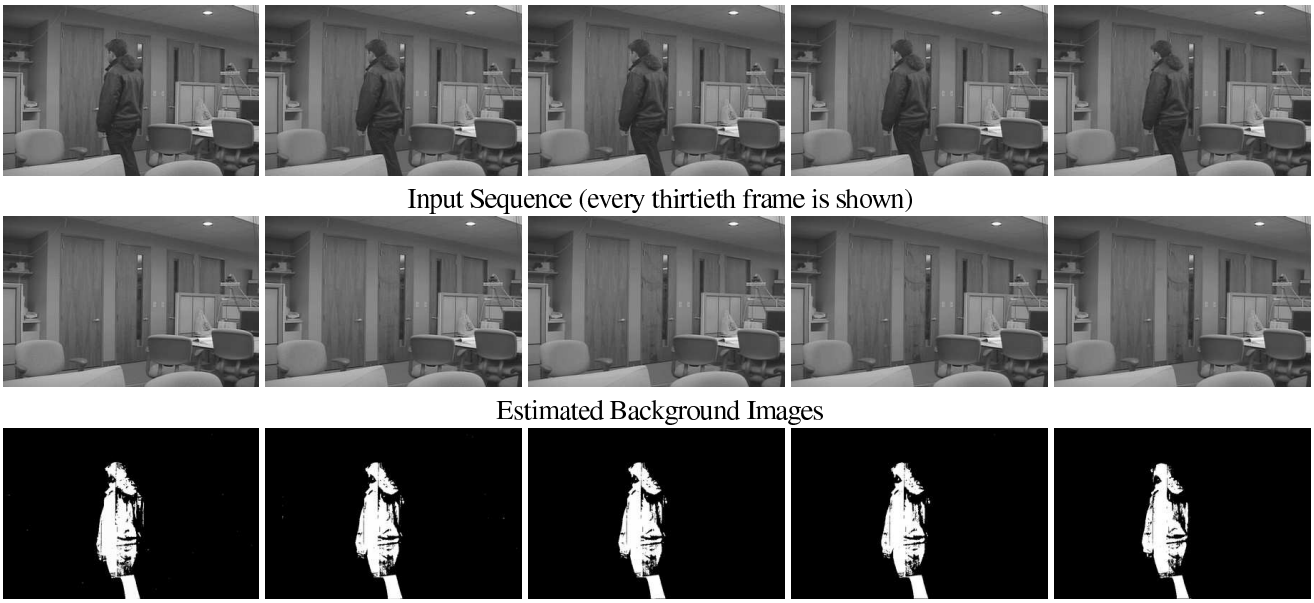


Figure 4: Fixing "sleeping man" problem. Performance of the dual-chain system on the sequence shown in Figure 1. Note that the correct background model and foreground maps are maintained while the person is stationary.

5 Conclusions

We have proposed a method for combining probabilistic feature extraction and object tracking systems that significantly improves tracking results. 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 in our approach, we obtained a straightforward means of mediating between the constituent models. The resulting model has a clear probabilistic interpretation of reconciling multiple generative models (each corresponding to a particular set of independence assumptions and dynamical model) that describe the same observations. 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.

We note that using the proposed framework requires some extra modeling in order to combine existing low- and high-level vision algorithms. Specificially, an integrated model is enabled by the introduction of an explicit latent appearance model. An integrated model is desirable for reasons of global consistency. However, exact inference on the resulting combined model is complicated by the introduction of loops. Consequently, 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 appications. A coordinate-ascent based algorithm has been designed for the batch inference case and applied to structure-from-motion estimation. Our method has been demonstrated to compare favorably to the pure feedforward approaches in such diverse applications as articulated body tracking, background subtraction, and structure from motion estimation.

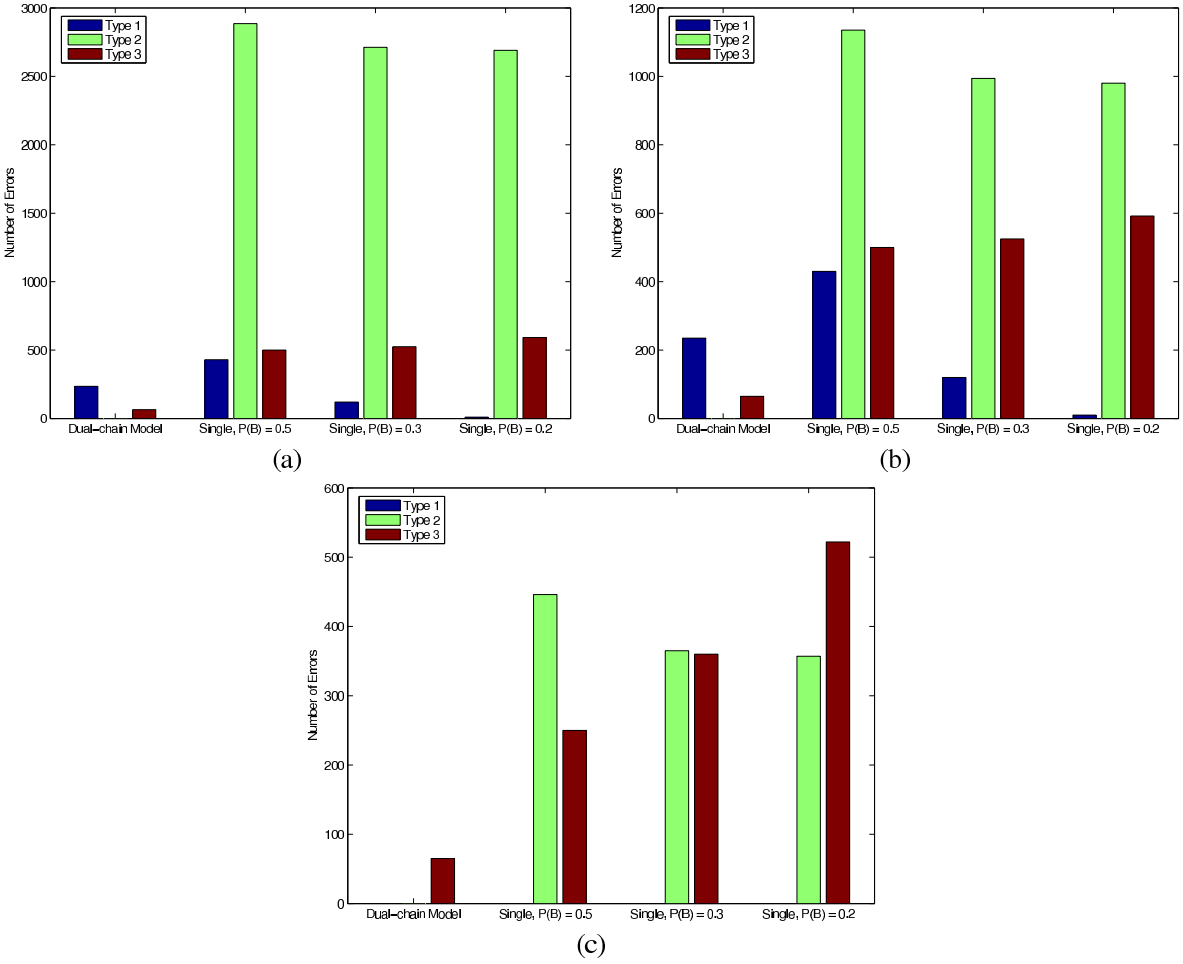


Figure 5: Quantitative evaluation of background subtraction performance on PETS 2001 image sequences. Three error classes were differentiated. 1: no foreground components corresponding to a **pedestrian** have been detected. 2: no foreground components corresponding to a **vehicle** have been detected. 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.

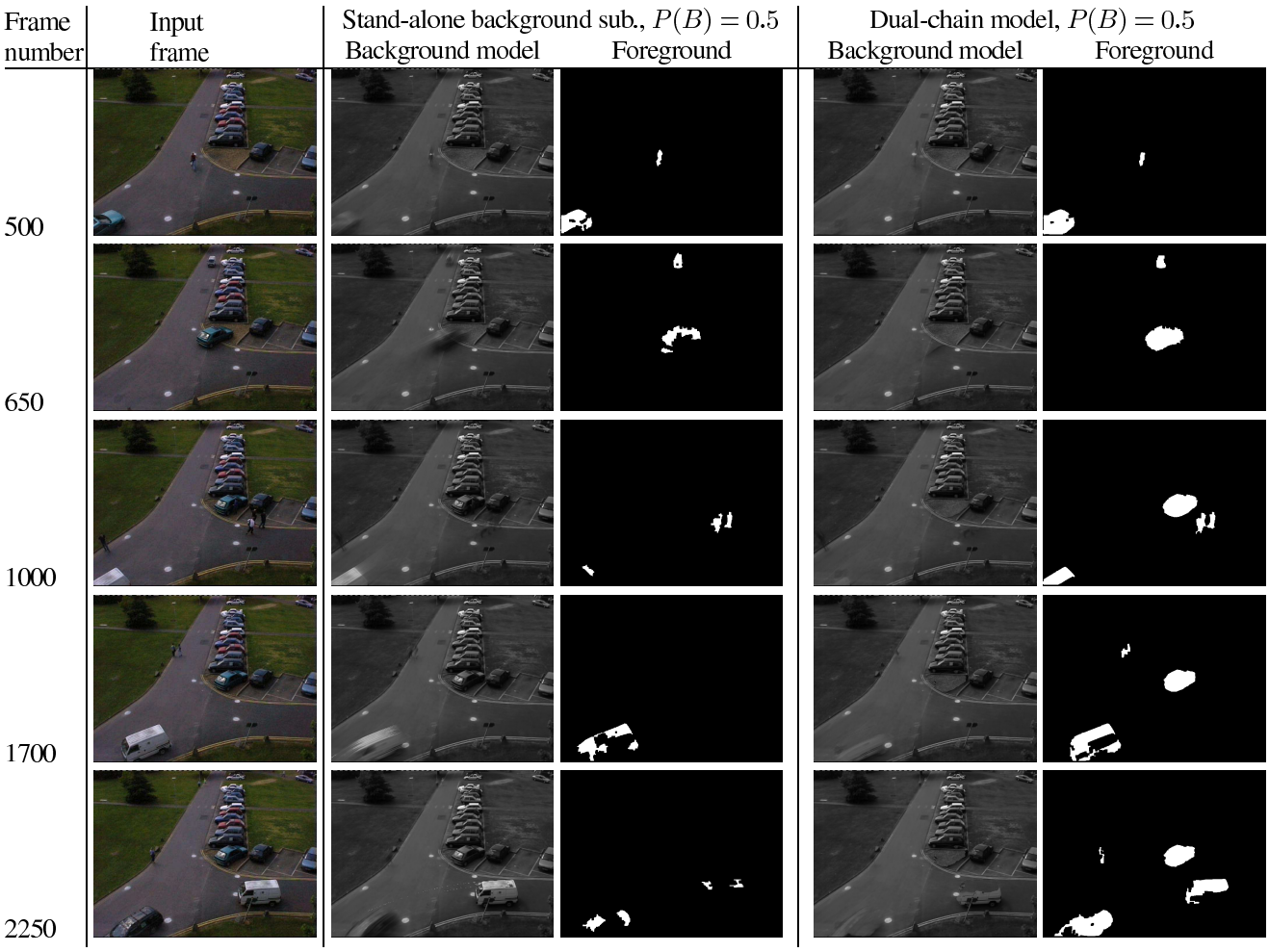


Figure 6: Qualitative comparison of background subtraction performance on one PETS2001 image sequence. Second column holds input frames. Estimated background model and the computed foreground components are presented in the third and fourth columns for stand-alone background subtraction and in the fifth 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.

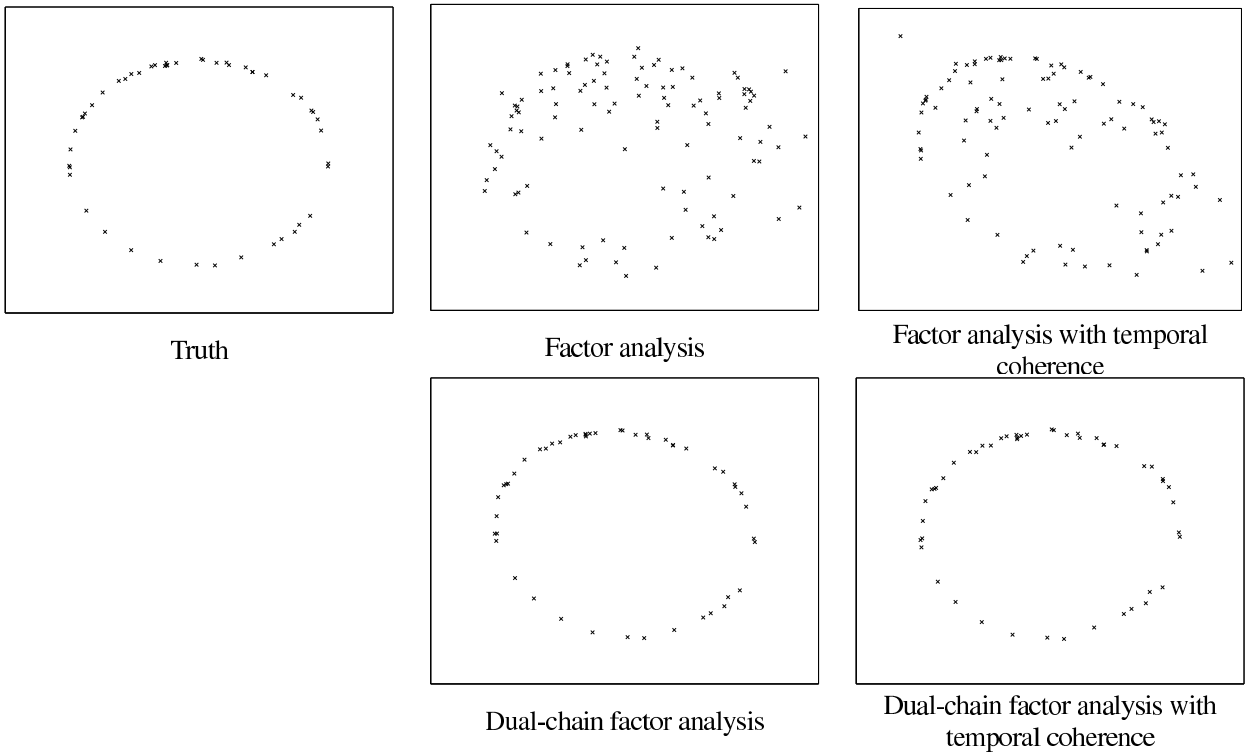


Figure 7: Comparison of typical performance of four structure-from-motion algorithms on a synthetic sequence ($P(loose) = 0.1$, $P(drift) = 0.3$, see text for details). Single-chain methods produce much worse results in the presence of occlusions due to their inability to establish correspondences between partial tracks of the same point.

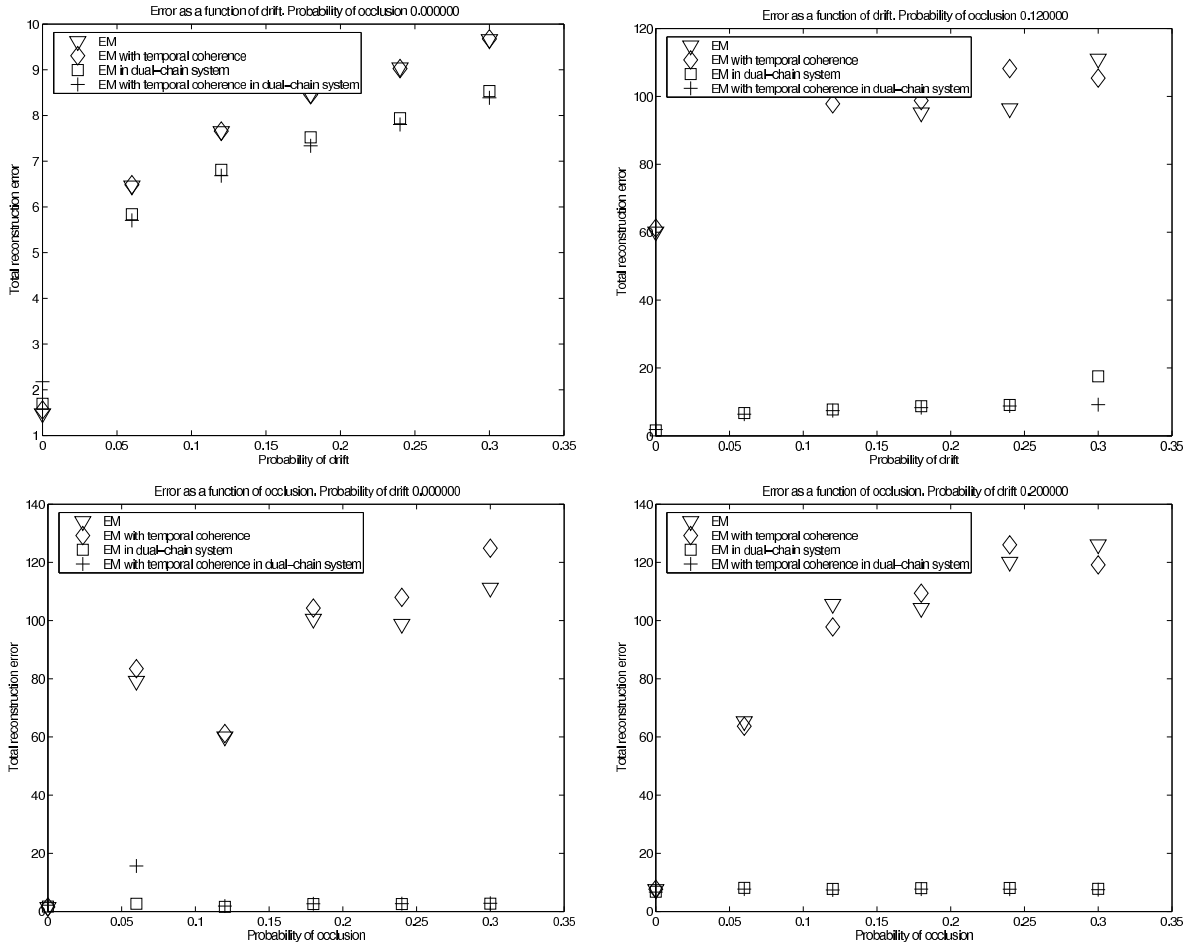


Figure 8: Quantitative of structure-from-motion recovery algorithms on the synthetic sequence with varying amounts of drift and occlusion. Top row—total reprojection error as a function of drift with no occlusion, .i.e $P(loose) = 0$ (left); with 12% chance of occlusion, i.e., $P(loose) = 0.12$ (right). Bottom row—total reprojection error as a function of occlusion for $P(drift) = 0$ (left) and $P(drift) = 0.2$ (right). Dual-chain algorithms were able to approximately reconstruct shape in all cases. Single-chain methods failed for even small values of $P(loose)$.

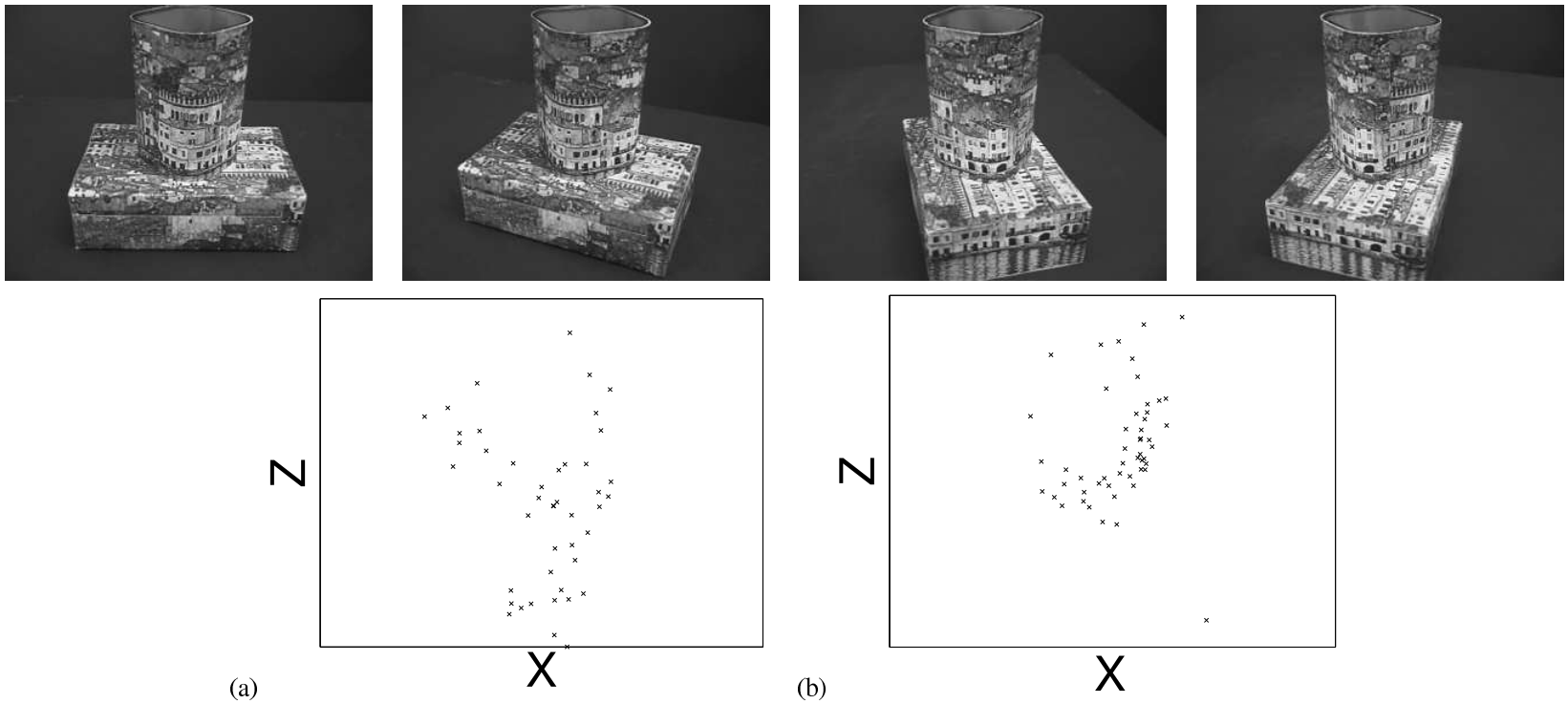


Figure 9: Comparing shape points computed by the stand-alone factor-analysis with temporal coherence and its dual-chain variant. Top row: four frames of the input video sequence. Bottom row: (a) View from above onto the top part of the shape produced by factor-analysis (b) View from above onto the top 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.

Figure 10: 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), and in the bottom row they are rendered in the side view.

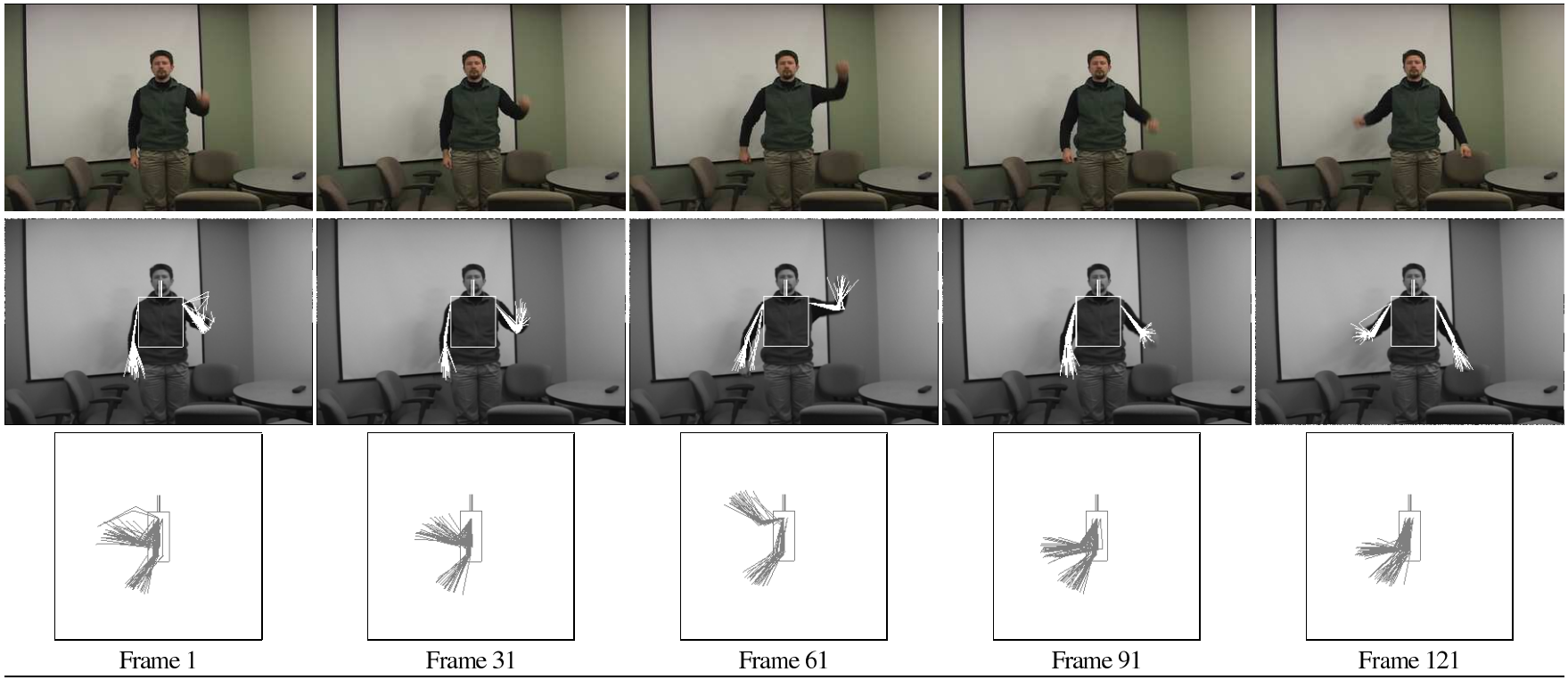


Figure 11: Applying dual-chain tracking to sample sequence 2. The top row contains input frames. Forty random particles from the estimated posterior pose distributions are shown: in the middle row, the particles are rendered onto the input image (frontal view), and in the bottom row they are rendered in the side view.

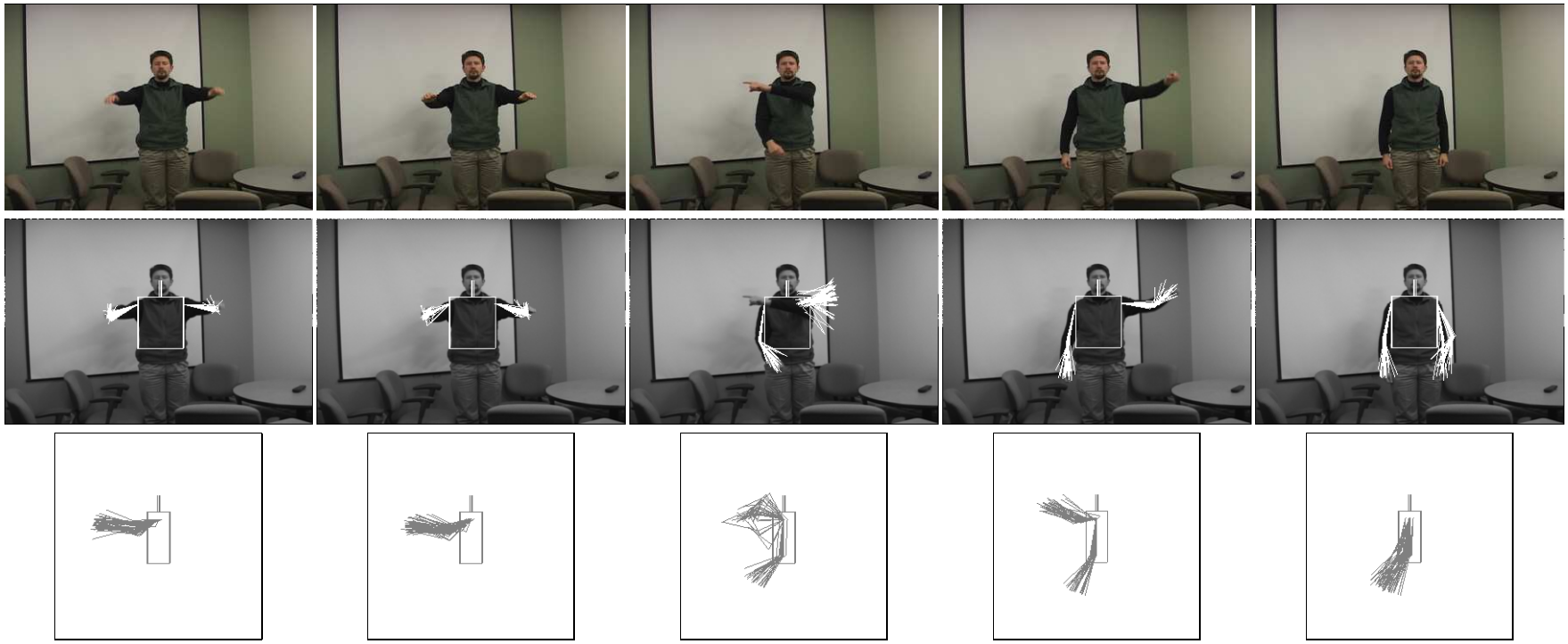
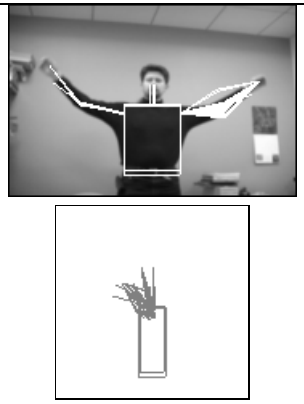
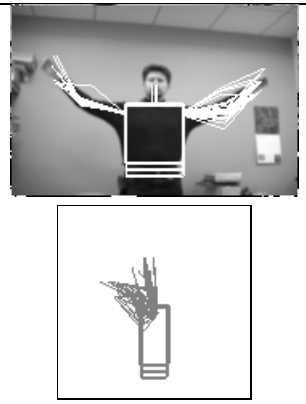
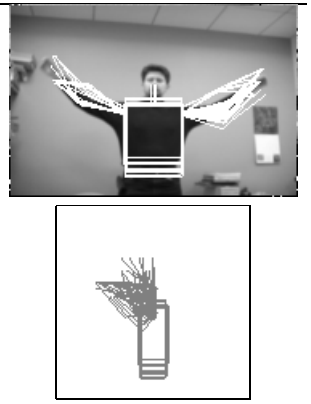
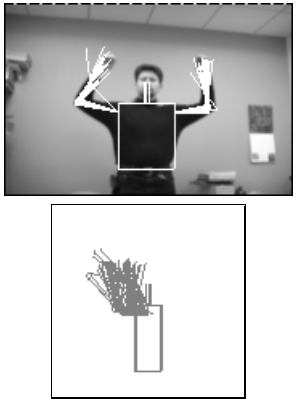
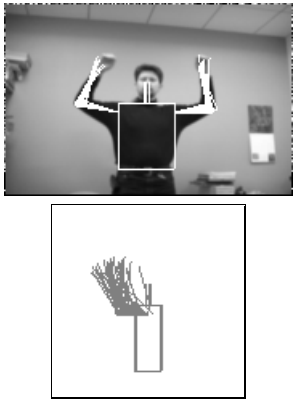
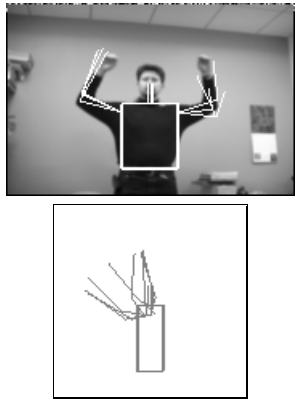
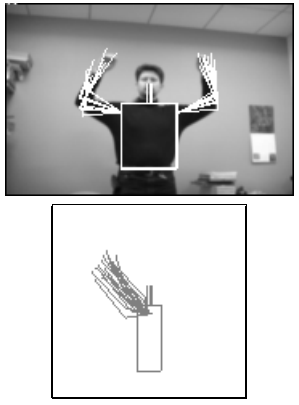
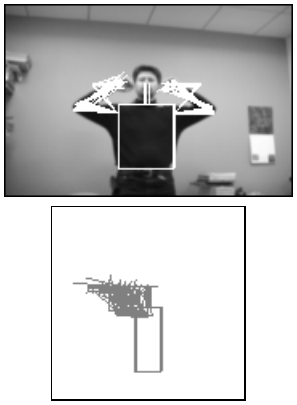
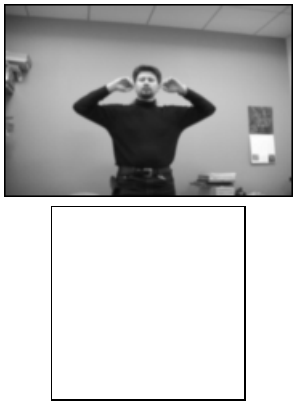
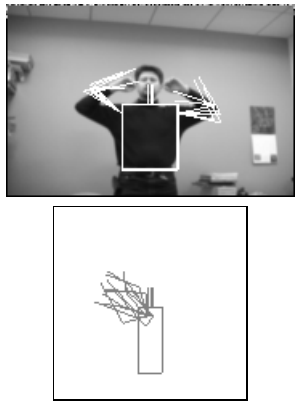
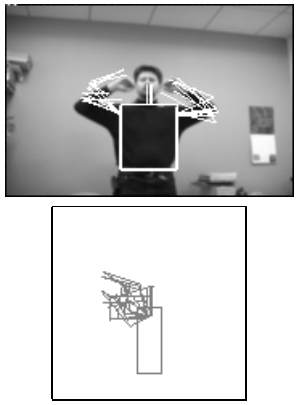


Figure 12: Applying dual-chain tracking to sample sequence 3. The top row contains input frames. Forty random particles from the estimated posterior pose distributions are shown: in the middle row, the patricles are rendered onto the input image (frontal view), and in the bottom row they are rendered in the side view. Note that while a mistrack has occurred on the third sequence near frame 61, the tracker was able to recover.

Frame	Multichain Tracker with 1,000 samples	Likelihood Sampling tracker with 1,000 samples	CONDENSATION Tracker with 5,000 samples	CONDENSATION Tracker with 15,000 samples
1				
11				
22				

Continued on the next page

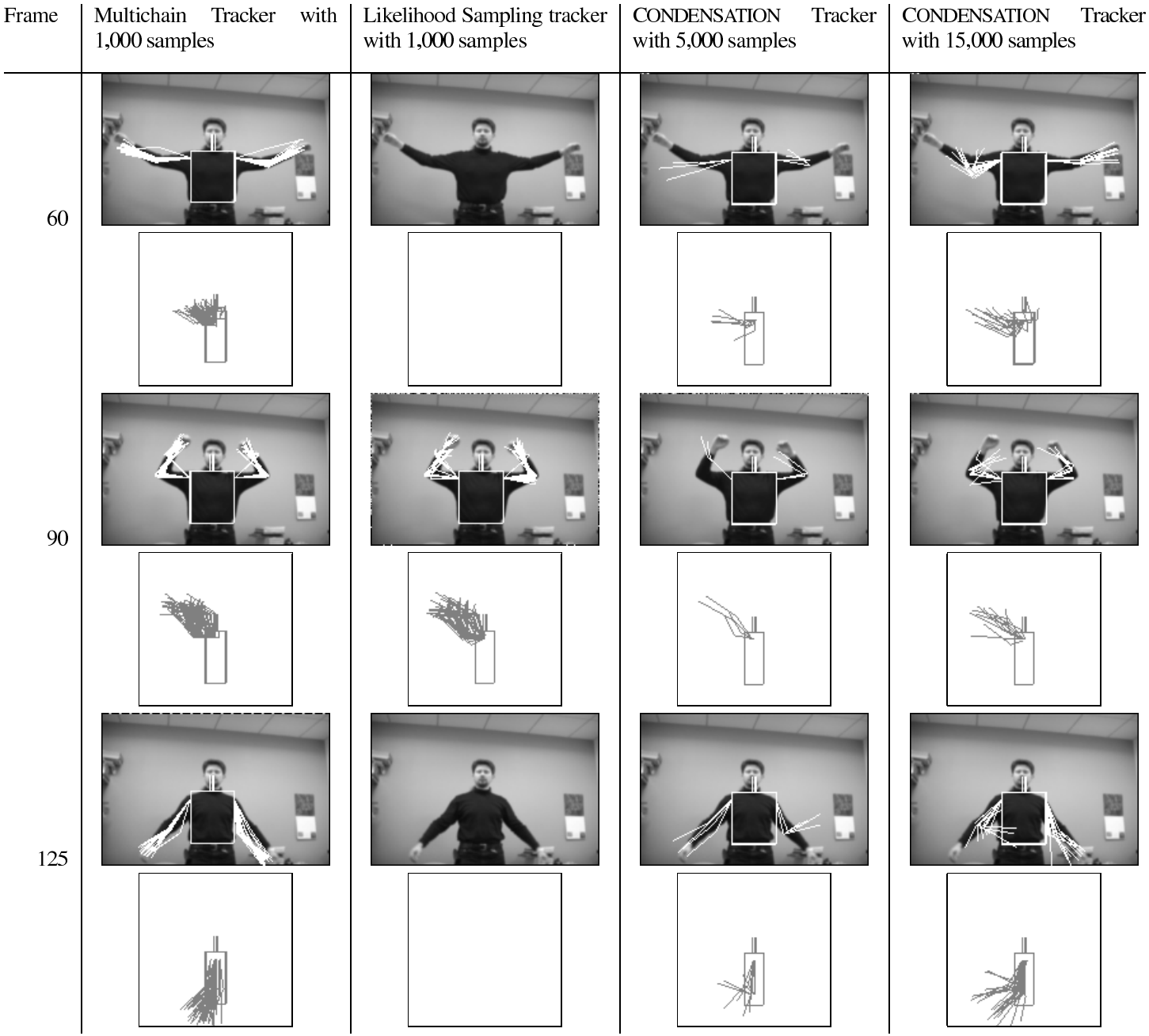


Figure 13: Applying four tracking algorithms to a sample sequence. For each frame a set of forty random pose samples were drawn from estimated posterior distribution and the corresponding skeletons were rendered (frontal view overlaid on the frame and side view below). Errors in feature detection caused likelihood-sampling tracker to fail on some of the frames (no samples were produced).

References

[1] S. Basu, I. Essa, and A. Pentland. Motion regularization for model-based head tracking. In *Intl. Conf. on Pattern Recognition (ICPR '96)*, Vienna, Austria, 1996.

[2] Andrew Brown and Geoffrey E. Hinton. Products of hidden markov models. In *Proceedings of Artificial Intelligence and Statistics*, pages 3–11, 2001.

[3] Joao Paolo Costeira and Takeo Kanade. A multibody factorization method for independently moving objects. *International Journal of Computer Vision*, 29(3):159–179, 1998.

[4] I. J. Cox and 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):138–150, Feb 1996.

[5] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journ. Oceanic Engineering*, 8(3):173–183, Jul 1983.

[6] Amit Gruber and Yair Weiss. Factorization with uncertainty and missing data: exploiting temporal coherence. In *NIPS*, 2003.

[7] Geoffrey E. Hinton. Products of experts. In *Proc, of the Ninth International Conference on Artificial Neural Networks*, pages 1 – 6, 1999.

[8] M. Isard and J.P. MacCormick. BraMBLe: A Bayesian multiple-blob tracker. In *ICCV01*, pages II: 34–41, 2001.

[9] David W. Jacobs. Linear fitting with missing data for structure-from-motion. *Computer Vision and Image Understanding: CVIU*, 82(1):57–81, 2001.

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

[11] T. Kurata, J. Fujiki, M. Kourogi, and K. Sakaue. A fast and robust approach to recovering structure and motion from live video frames. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 528–535, 2000.

[12] Philip F. McLauchlan, Ian D. Reid, and David W. Murray. Recursive affine structure and motion from image sequences. In *ECCV (1)*, pages 217–224, 1994.

[13] Nuria M. Oliver, Barbara Rosario, and Alex Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000.

[14] I. Reid and D. Murray. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision*, 18(1):41–60, 1996.

[15] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, June 1994.

[16] C. Stauffer and W.E.L Grimson. Adaptive background mixture models for real-time tracking. In *Proceedings of CVPR'99*, 1999.

[17] Leonid Taycher and Trevor Darrell. Bayesian articulated tracking using single frame pose sampling. In *Proc. 3rd Int'l Workshop on Statistical and Computational Theories of Vision*, Oct 2003.

[18] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.

[19] Kentaro Toyama and Eric Horvitz. Bayesian modality fusion: Probabilistic integration of multiple vision algorithms for head tracking. In *ACCV'00*, 2000.

[20] Christopher Richard Wren, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997.

[21] Q. Zhou and J. Aggarwal. Tracking and classifying moving objects from videos. In *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, 2001.