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, and this is often exploited by visual tracking algorithms. A local dynamic model is typically used to extract image features that are then used as input 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 are incorporated into an 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 background subtraction.*

1. Introduction

Raw image data is rarely used as input to object tracking algorithms. It is usually preprocessed with low-level feature extractors, whose output is then treated as observations by high-level algorithms. The most popular of these low-level features are foreground silhouettes, produced by background subtraction, and feature-point locations, obtained by point-tracking methods. These algorithms operate on small image patches (a single pixel in some background subtraction frameworks), and often use strong assumptions about local behavior.

Feature-point trackers assume that the image patch about the point of interest has 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 do not

hold, feature extractors fail, leaving high-level algorithms to pick up the pieces. The alternative—a monolithic system jointly modeling local and global dynamics—is also not appealing. Monolithic systems are less efficient than two-stage approaches, and the latter also have the advantage of modularity, since object trackers and feature extractors may be designed independently.

Both object tracking and feature extraction are often formulated as inference in probabilistic generative models. We would like to combine them in a framework that also has a clear probabilistic interpretation, with feedback behavior arising from the structure of the model. Preserving modularity is also important, since we would like to base inference on existing feature-extraction and object-tracking algorithms.

Our motivation in building such a framework is based on the fact that while features are *observed* variables in object-tracking models, they may be made *latent* by explicitly modeling their dependency on the images. Feature values are already latent variables in feature-extraction models. The models may then be combined by sharing these variables, in a manner similar to that used in Product of Hidden Markov Models (PoHMMs) approach [1].

Our framework is based on the multichain probabilistic model. The main difference between this approach and PoHMMs is that individual chains share a *latent* rather than an observed variable. This enables a flow of information between states of the trackers, realizing the desired feedforward/feedback property. Since trackers are coupled only through features, modularity is preserved with only minimal modification to the algorithms.

We demonstrate the advantages of our framework by applying it to structure from motion recovery and adaptive background subtraction problems.

2. Related Work

Reliability of feature extraction has long been a concern in computer vision. In the absence of high-level information, the best that can be done is to compute not only feature values but also uncertainty about the measurement. For ex-
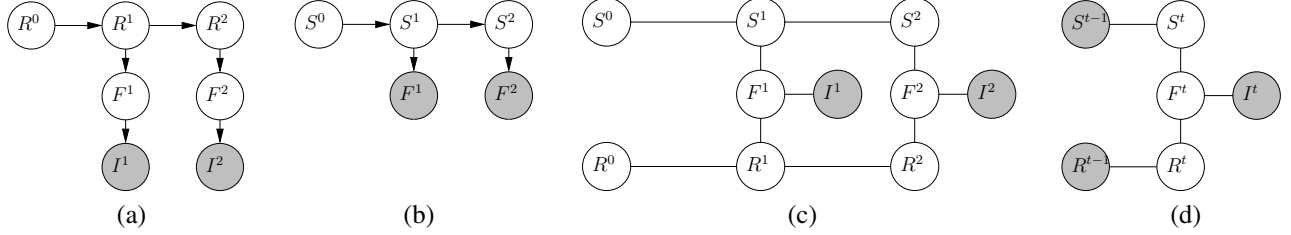
Figure 1: Combining local and global dynamics for object tracking. (a) A generative model used in feature extraction algorithms. The state $R = \{R_k\}$ evolves according to the local dynamic model, $p(R^{t+1}|R^t) = \prod_k p(R_k^{t+1}|R_k^t)$. At time t, the observed image is based on $p(I^k|F^k)$, where the feature set, $F^t = \{F_k^t\}$, is generated from 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 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.).(d) A tree-shaped subgraph on which a single step of approximate inference 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.

ample, dissimilarity computations [11] and Kalman filtering [10] have been used to estimate uncertainty of feature-point tracking. Kalman-filter based approaches such as MHT [2] and JPDAF [3] have been used to disambiguate feature tracks based on individual motion models.

It has long been observed that incorporating some feedback from high-level algorithms into feature extraction in many cases improves overall performance. Indeed, many heuristic-based methods for providing such a feedback have been incorporated into real-life systems. Knowledge about global dynamics has been involved in feature extraction on multiple levels. On the basic level, robust methods such as Least Median Squares have been used to reject feature locations that are deemed to be outliers [7]. The complete integration of feature extraction and object motion is achieved in monolithic systems [14, 5], which jointly model foreground and background processes.

The framework proposed in this paper is most closely related to the intermediate integration approaches of [8] and [6]. 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.

3. Formulation of Dual-chain Model

Feature extraction algorithms can often be seen as inference in a generative model with a structure similar to the one in Figure 1(a). The feature set at time t, $F^t = \{F_k^t\}$ (e.g. foreground/background labels and pixel values), is generated based on the hidden 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 inde-

pendent, with 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.

Similarly, probabilistic object tracking algorithms may be formulated as inference in the model shown in Figure 1(b). The hidden state, S^t, evolves according to global dynamics, $p(S^{t+1}|S^t)$. 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.

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

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

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

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

As we showed in [13], for the case of articulated tracking, this interfeature dependency may be accounted for by introducing feature prediction available from the global dynamic model. We can define an approximation

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

to $q(F^t, R^t; I^{0..t-1})$ that combines information from both systems via a product

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

This is equivalent to a dual-chain model shown in Figure 1(c). 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).

4. Inference Methods in Dual-Chain Model

The dual-chain model proposed in the previous section is loopy, and inference may in general be complicated. Single-chain inference algorithms developed for object and feature tracking cannot be directly used in our framework, since it introduces dependencies unrepresented in individual models. Fortunately, they can be easily adapted, due to the fact that individual models become decoupled if the features are known. The general strategy is for each model to incorporate the features predicted by the other model and the observed images. Both online (filtering) and batch (smoothing) algorithms can be thus adapted.

4.1. Approximate Filtering
While exact filtering in the dual-chain model requires modeling the joint distribution, $p(S^t, R^t|I^{0..t-1})$, in [13] we have shown an approximate filtering algorithm which

uses only marginal state distributions $p(S^t|I^{0..t-1})$ and $p(R^t|I^{0..t-1})$.

Consider the model in Figure 1(c). If the initial states S^0 and R^0 are independent (as shown), then posterior distributions $p(S^1|I^1)$ and $p(R^1|I^1)$ can be (exactly) computed using Belief Propagation [9] on a tree-shaped subgraph:

$$p(S^1|I^1) = \frac{1}{Z} \int dS^0 \phi(S^1, S^0) p(S^0) \int dF^1 \Big[\phi(F^1) \qquad (4)$$
$$\phi(F^1, S^1) \int_{R^1} \Big[\phi(F^1, R^1) \int R^0 \phi(R^1, R^0) p(R^0) \Big] \Big],$$

where $\phi(F^1) \equiv \phi(I^1, F^1)$. The equivalent expression of $p(R^1|I^1)$ 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 $p(S^1, R^1|I^1)$:

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

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:

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

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 1(d), using the marginal state distribution obtained at time $t - 1$ as priors at time t.

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.

4.2. Batch Optimization

While filtering is appropriate for online tasks, some object-tracking problems are formulated as global optimization in single-chain models such as one in Figure 1(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. The summary of out method is presented in Algorithm 2.

5. Experiments

We demonstrate an adaptive background algorithm based on the dual-chain filtering, and structure from motion estimation based on the dual-chain batch optimization.

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 while keeping feature states ($R^{0..T}$) fixed.
    COMPUTE feature predictions from the object model for each time-step
    APPLY feature-extraction algorithm to all available observations, keeping object states ($S^{0..T}$) fixed and incorporating feature predictions from the object model
end while

5.1. Background Modeling

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). Unfortunately, they also adapt to foreground objects that remain stationary for extended periods of time. After these objects "fade" into the background, their locations are no longer labeled as foreground.

Common adaptive background algorithms similar to [12] 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 1(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 1(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 [12]. 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 models, we 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 the background probability value at every pixel, and extracted a set of connected components.

We were interested in three common classes of errors: (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 2. Sample frames from the first sequence with corresponding estimated background images and foreground components are shown in Figure 3.

Replacing feed-forward tracking algorithm with a dual-chain framework did not result in large performance penalty. In our experiments, the difference between running times of dual-chain algorithm and feed-forward system was less than 4%. Semioptimized 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).

5.2. Structure from Motion

We have evaluated the global optimization algorithm (section 4.2) by applying it to the problem of extracting structure from motion. For the object-level optimization algorithm we use the factor-analysis based factorization methods of [4] (both with and without temporal coherence). Dual-chain variants were obtained by combining them with a Kalman-filter based feature-point tracker. 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 4. 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 5. 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 6. Shape recovered by stand-alone factor analysis contains many spurious points, but the dual-chain framework succeeded in approximately estimating the correct shape.

6. Conclusions

We have proposed a method for combining probabilistic feature extraction and object tracking systems, and demonstrated significantly improved tracking results. The approach was motivated by the simple 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.

Inference algorithms that have been designed for single-chain feature extraction and object-tracking modules are not directly applicable in this framework. Inference on the dual-chain model is further complicated due to the introduction of loops in the graphical structure representing the combined models. However, 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 inference appropriate for online tracking (filtering). An coordinate-ascent based algorithm has been designed for the batch inference case, and applied to structure-from-motion estimation. In both cases, our method has been demonstrated to compare favorably to the pure feed-forward approaches.

Videos and additional results can be found at http://people.csail.mit.edu/people/lodrion/Research/multichain/.

References

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

[2] 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.

---

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

[2] We used part of an original sequence from http://www.cs.ucla.edu/~hljin/research/voi.html
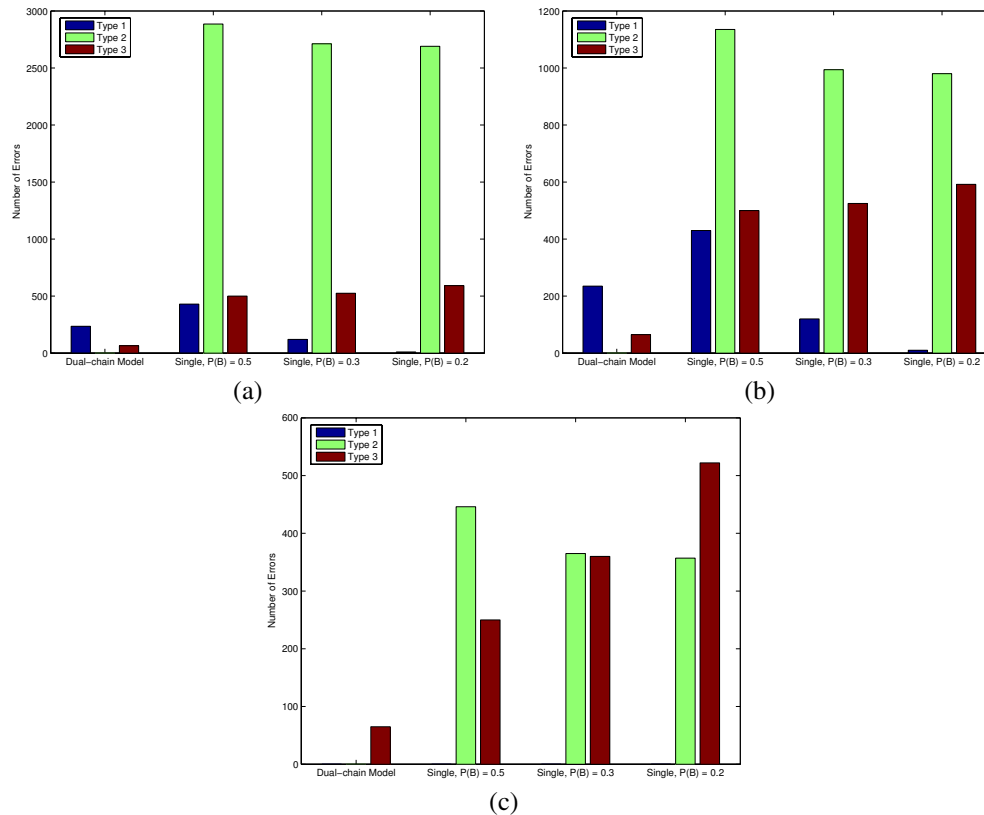
Figure 2: Quantitative evaluation of background subtraction performance on PETS 2001 image sequences. Three error classes were differentiated. 1: no foreground components corresponding to a **pedestrian** have been detected. 2: no foreground components corresponding to a **vehicle** have been detected. 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.

[3] 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.

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

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

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

[7] 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.

[8] 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.

[9] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1997.

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

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

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

[13] Leonid Taycher, John W. Fisher III, and Trevor Darrell. Combining simple models to approximate complex dynamics. In *Proc. Workshop on Statistical Methods in Video Processing*, May 2004.

[14] 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.

Frame number	Input frame	Stand-alone background subtraction		Dual-chain model	
		Background model	Foreground	Background model	Foreground
500					
650					
1000					
1700					
2250					

Figure 3: 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 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.

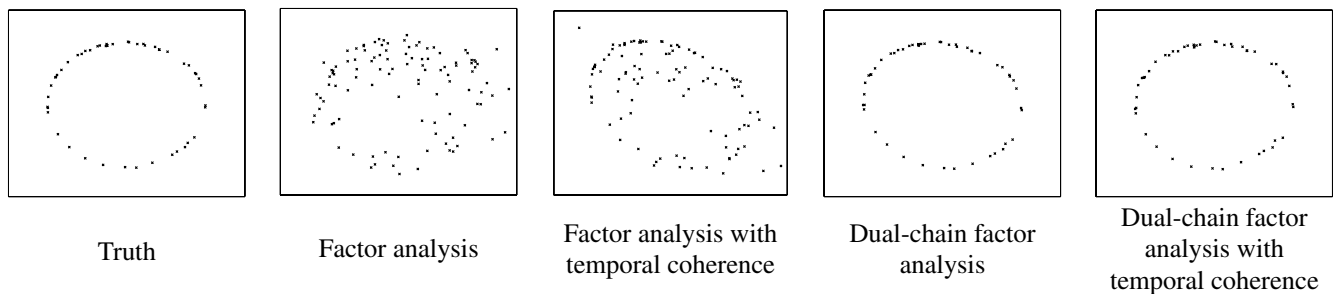| Truth | Factor analysis | Factor analysis with temporal coherence | Dual-chain factor analysis | Dual-chain factor analysis with temporal coherence |

Figure 4: Comparison of typical performance of four structure from motion algorithms on a synthetic sequence. 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.
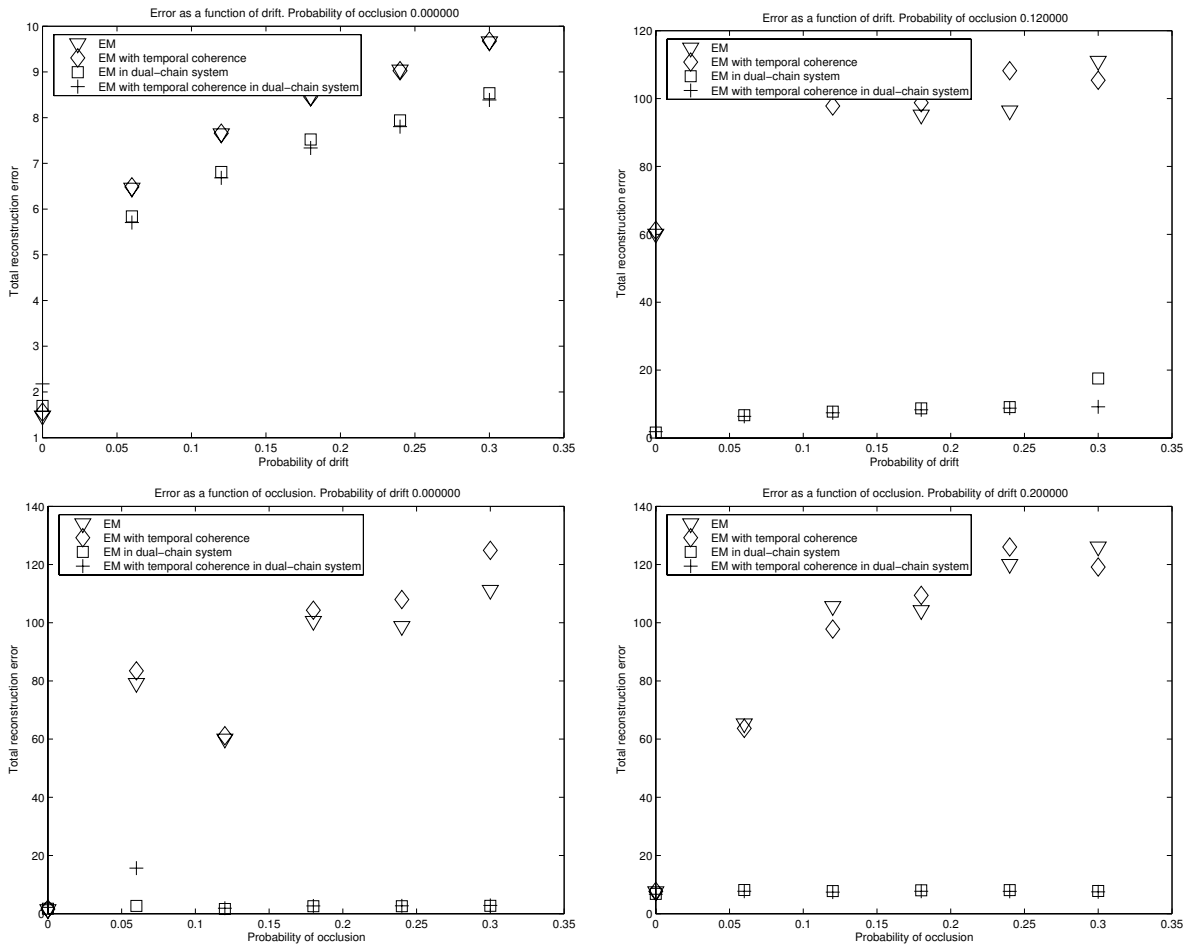
Figure 5: Quantitative of structure from motion recovery algorithms on synthetic sequence with varying amounts of drift and occlusion. Top row—total reprojection error as a function of drift with no occlusion, .i.e $P(loose) = 0$ (left) with 12% chance of occlusion, i.e $P(loose) = 0.12$ (right). Bottom 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 $P(loose)$.
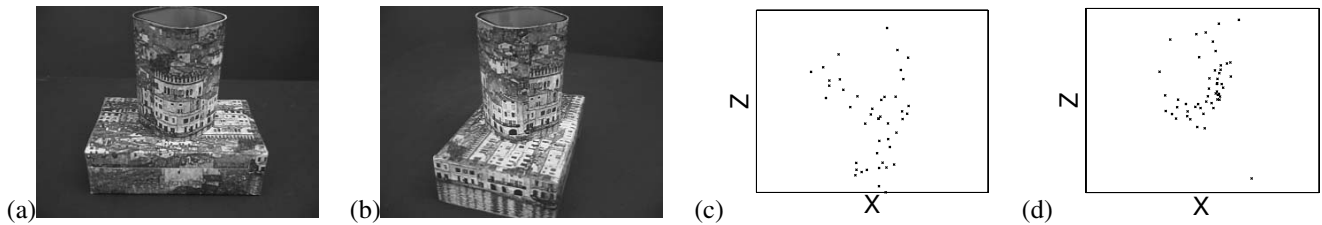

Figure 6: Comparing shape points computed by the stand-alone factor-analysis with temporal coherence and its dual-chain variant. (a, b) First and last frames of the video sequence. (c) View from above onto the top part of the shape produced by factor-analysis (d) View from above onto the top part of the shape produced by dual-chain algorithm. Note that the shape produced by factor analysis contained more than half of spurious points.