

# Learning Higher-order Transition Models in Medium-scale Camera Networks

Ryan Farrell, David Doermann, Larry S. Davis

Department of Computer Science  
Institute for Advanced Computer Studies  
University of Maryland  
College Park, MD 20742

{farrell, lsd}@cs.umd.edu, doermann@umiacs.umd.edu

## Abstract

We present a Bayesian framework for learning higher-order transition models in video surveillance networks. Such higher-order models describe object movement between cameras in the network and have a greater predictive power for multi-camera tracking than camera adjacency alone. These models also provide inherent resilience to camera failure, filling in gaps left by single or even multiple non-adjacent camera failures.

Our approach to estimating higher-order transition models relies on the accurate assignment of camera observations to the underlying trajectories of objects moving through the network. We address this data association problem by gathering the observations and evaluating alternative partitions of the observation set into individual object trajectories. Searching the complete partition space is intractable, so an incremental approach is taken, iteratively adding observations and pruning unlikely partitions. Partition likelihood is determined by the evaluation of a probabilistic graphical model. When the algorithm has considered all observations, the most likely (MAP) partition is taken as the true object trajectories.

From these recovered trajectories, the higher-order statistics we seek can be derived and employed for tracking. The partitioning algorithm we present is parallel in nature and can be readily extended to distributed computation in medium-scale smart camera networks.

## 1. Introduction

While traditional CCTV surveillance systems are generally limited to archival and operator monitoring, the recent proliferation of Network Cameras and Smart Cameras [1] heralds a new generation of intelligent surveillance architectures. Future surveillance devices will be endowed with substantial computational and communication

resources. The challenge is to provide them with commensurate algorithms to collectively interpret activity within the network.

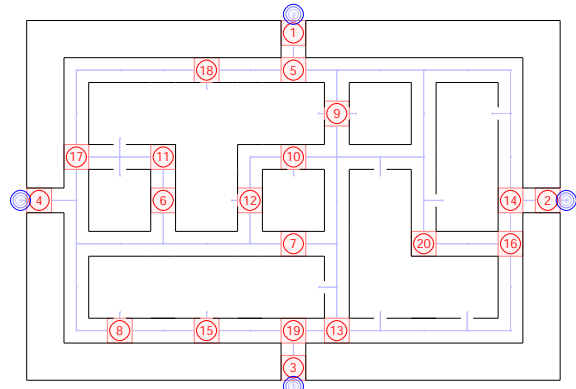


Figure 1. **Simulated Camera Network.** One of several camera networks used in our experiments. This one contains 20 cameras: four peripheral cameras (numbered 1-4), with the others (5-20) placed randomly within the halls of the office environment.

A fundamental challenge in surveillance is tracking objects and individuals throughout the network in spite of occlusion and lapses in observation, changing illumination, etc. Stauffer and Tieu [15] provide an excellent description of the general tracking problem, suggesting that an ideal tracking system should produce “only as many tracking sequences as there were independently moving objects in an environment, regardless of the number of cameras or their overlap”. Successful tracking requires the maintenance of object identity, typically relying both on an understanding of the camera network topology and the ability to match properties such as appearance and dynamics across observations [17].

We focus on the problem of recovering topology in uncalibrated medium-scale (10-1000 camera) surveillance networks. While previous work has focused mainly on first-order relationships (*i.e.* adjacency), our focus is on higher-

order topological relationships and transition models.

Consider the case of a newsstand located in an airport terminal. We might generally expect most of the newspapers and magazines sold to be purchased by departing passengers to read on their flight or while they wait for boarding. Suppose we know, however, that the few people who come from a gate and stop by the newsstand almost always continue in the direction of the main terminal. Contrary to the general expectation that people go from the newsstand to a gate, we have a strong prior belief that people tracked from a gate to the newsstand will continue toward the main terminal when they leave.

The example illustrates that learning higher-order topological relationships can potentially improve tracking performance. Other benefits of recovering higher-order relationships include resilience to camera/node failure. If second-order topological data is available, then we can overcome loss of any given camera, or even multiple non-adjacent cameras.

## 2. Related Work

In the computer vision literature, previous work on uncalibrated camera network topologies has focused primarily on pairwise camera relationships. Stauffer and Tieu [15] illustrate the possible types of camera overlap: (i) non-overlapping views (no mutually observable volume); (ii) partially overlapping views (views are connected, adjacent cameras have regions of overlap); (iii) completely overlapping views (existence of a spatial volume mutually observable by all cameras); and (iv) the general case combining the other types. The problem they tackle is that of modelling regions of overlap for groups of cameras with at least partial overlap (type (ii)), where the overlapping regions lie on or near a ground plane. The solution they propose is to consider cameras pairwise and use temporally co-occurring observations sequences in the two views to estimate a homography between them and the region of overlap or mutual observability.

Recently there has been an increased interest in non-overlapping camera networks. Makris *et al.* [11] attempted to recover the network topology to facilitate tracking between spatially adjacent cameras by estimating the transition delay between two cameras using cross-correlation on large numbers (thousands) of departure/arrival observations. Tieu *et al.* [16] suggested the use of mutual information as a measure of pairwise statistical dependence, using Markov Chain Monte-Carlo (MCMC) to simultaneously recover the correspondence between departures/arrivals and the transition delay distribution. In contrast to these entirely unsupervised techniques, Javed *et al.* [8] use labeled ground-truth trajectories to generate a nonparametric model (using Parzen windows) of transition probability between cameras. The parameters they use in building the model are

exit location, entrance location, exit velocity and time delay. After generating this model, they employ an offline tracking algorithm based on bipartite graph matching and an online approach which updates the KDE in real-time.

A vast literature addresses the problem of data association. Among the earliest work in this field was Reid's algorithm [14], a Bayesian formulation for multiple target tracking in a single view (*e.g.* associating radar tracks). Where Reid outlined a multiple-hypothesis tracker (MHT) to deal with the intractable space, Cox and Hingorani [4] later provided an efficient implementation of Reid's algorithm based on bipartite graph matching. Huang and Russell [7] applied this approach to the problem of highway traffic monitoring, offering an improved matching algorithm which scores the quality of each association.

Recent research on multiple-camera surveillance continues to use the Bayesian formulation. Given certain constraints, Kettner and Zabih [10] are able to frame the multi-camera problem as a Linear Program. Pasula *et al.* [13] and Oh *et al.* [12] use MCMC-based approaches. Zaidel *et al.* [18, 19, 20] employ an approach similar to Pasula *et al.*, but use Dynamic Bayes Networks to evaluate hypothesis likelihoods and an EM-based algorithm for learning model parameters.

Another body of relevant work is found in the sensor networks literature. The ad hoc and inherently distributed nature of wireless sensor networks has led researchers to a focus on distributed inference techniques. Funiak *et al.* [6] presented an online approach for localizing a network of cameras, essentially employing distributed probabilistic inference to approximately calibrate a network of sensors based on observations of an object moving through the network. Distributed inference is also employed for multi-target tracking. Examples include Chen *et al.* [2] and Chu *et al.* [3] though both apply distributed inference to tracking in dense networks of calibrated, non-visual sensors.

## 3. Learning Camera Network Topology

Our objective is to learn as much as possible about the camera network, while assuming as little as possible. The primary purpose for learning network topology is to improve the models used by tracking algorithms. Understanding the probabilities of potential events that could follow an object's departure from a given camera provides information which should be helpful in tracking the object.

Particularly difficult scenarios arise when multiple objects with similar appearances are simultaneously present in the network. A network topology model can help discriminate between ambiguous objects when appearance alone cannot. For example, suppose we know that a particularly object  $x$  cannot get from camera 1, where it was last seen, to camera 3 without passing through camera 2. If an object with the appearance of  $x$  is then seen at camera 3 before one

is seen at camera 2, we deduce that the object in camera 3 cannot be object  $x$ .

To recover the topological relationships, we focus not just on first-order ‘‘adjacency’’. While most previous work only considered where an object leaving camera  $i$  could appear next, we are interested in higher-order transition models which provide a richer description of object movement tendencies. As we recover complete trajectories, the range of queries that can be addressed are broader, *e.g.* ‘‘What fraction of objects passing through cameras 5 and 7 will, at some time later reach camera 4?’’.

We make only one assumption about the spatial distribution of the cameras, requiring non-overlapping fields of view. The only information we require a priori is a labelled set of *peripheral cameras*, the subset of cameras where objects may enter or exit the network. We also assume the network is initially empty. Without these two constraints, we would need to consider the possibility that each observation is of a unique, previously unseen object.

Our task is then to recover these underlying trajectories, using what little information we know. Given the peripheral-labeled cameras and the full set of observations, we aim to simultaneously determine how many objects have passed through, learn their respective appearances and associate which observations belong to which objects. The probabilistic approach we use to partition the observations into individual object trajectories is described next, in Section 4.

## 4. Bayesian Observation Partitioning

Several Bayesian approaches for problems such as data association and tracking are described in Section 2. Our solution closely follows the Bayesian framework presented in Zajdel [18] for multi-camera tracking (similar approaches used in [12, 13]). This approach learns model parameters incrementally by accumulating observations into consideration incrementally and probabilistically evaluating proposed partitionings of these observations into objects.

### 4.1. Finding the Optimal Partition

In this approach, we first consider the entire set of observations  $\mathbf{O} = \{o_1, o_2, \dots, o_N\}$ . These observations represent the observable portions of the trajectories of  $K$  (value unknown) objects moving within the network. Each observation represents an object passing through a given camera at a particular time. The observations could have been generated by a single object ( $K = 1$ ),  $N$  distinct objects with just one observation each ( $K = N$ ), or, some number of objects in between ( $1 < K < N$ ). Our goal is to select a partition  $\omega \in \Omega_N$  of the observations  $\mathbf{O}$

$$\mathbf{O}_\omega \stackrel{e}{=} \mathbf{O}_1 \cup \mathbf{O}_2 \cup \dots \cup \mathbf{O}_{K_\omega} \quad (1)$$

such that each set  $\mathbf{O}_k = \{o_1^{(k)}, o_2^{(k)}, \dots, o_{n_k}^{(k)}\}$  contains all  $n_k$  observations of the  $k^{\text{th}}$  object, the temporal sequence  $o_1^{(k)}, o_2^{(k)}, \dots, o_{n_k}^{(k)}$  describing object  $k$ ’s trajectory or path through the network.

Since the true number of objects is unknown, we consider various partitionings (see Figure 2) and in estimating the most likely one, hopefully recover the correct set of objects with their respective trajectories. Formally, we consider the space  $\Omega_N$  of all partitions of the  $N$  observations, evaluating each partition’s likelihood in the context of established priors and the evidence (observations) collected. However, for any nontrivial observation size  $N$ , considering all such partitionings exhaustively is intractable<sup>1</sup>. We therefore use a procedure reminiscent of Reid’s multiple hypothesis tracking approach [14], to prune the partition space.

We begin with a small initial observation set consisting of the first  $m$  observations,  $\mathbf{O}_0 = \{o_1, o_2, \dots, o_m\}$ . We exhaustively enumerate all partitions in  $\Omega_m$  and evaluate the likelihood of each partition  $\omega$  using the inference method described below in Section 4.2. At this point we discard unlikely partitions, retaining only the  $B$  best (most probable) partitions, associating with each retained partition an updated model reflecting the properties of its respective trajectories. Formally, we denote this initial set of hypotheses as  $\mathbf{H}_0 = \{h_1^{(0)}, h_2^{(0)}, \dots, h_B^{(0)}\}$  where  $h_i^{(0)}$  is comprised of its partition  $\omega_i^{(0)}$  and its resulting transition model  $T_i^{(0)}$  (the transition model is covered more fully in section 4.2.2).

With our initial set of hypotheses  $\mathbf{H}_0$  formed, we begin an incremental search process akin to Fox’s beam search [5]. At each iteration we add a few,  $s$ , additional observations and again consider the resulting partitions and prune all but the best. For the  $\tau^{\text{th}}$  iteration, to extend each hypothesis  $h_i^{(\tau-1)} \in \mathbf{H}_{\tau-1}$  with  $s$  additional observations, we must evaluate  $O(k^s)$  amended partitions<sup>2</sup>, where  $k$  is the number of trajectories in  $h_i^{(\tau-1)}$ . Due to the exponential complexity  $O(B \cdot k^s)$ , small values of  $s$  are used in practice. After these amended partitions are evaluated, the unlikely partitions are again pruned and we form  $\mathbf{H}_\tau$  by retaining the  $B$  most likely amended partitions, each with its updated model. This incremental process is continued until all of the observations have been brought into consideration and the most likely partition in the final hypothesis set is taken as the final MAP estimate (this is described in greater detail below).

<sup>1</sup> The number of ways to partition a set of  $n$  elements is given by the  $n^{\text{th}}$  Bell number,  $B_n$ . The first 10 Bell numbers are 1, 2, 5, 15, 52, 203, 877, 4140, 21147, 115975 and, in general,  $B_{n+1} = 1 + \sum_{k=1}^n \binom{n}{k} B_k$ .

<sup>2</sup> For  $s = \{1, 2, 3, \dots\}$ , adding  $s$  additional observations to a hypothesis of size  $k$  will produce  $\{k+1, k^2+2k+2, k^3+3k^2+6k+5, \dots\}$  amended partitions to evaluate. In essence, each observation added can go into any one of the existing trajectories or be considered as a new object. The combinatorial complexity of adding  $s$  observations to a partition with  $k$  trajectories is  $O(k^s)$ , independent of the total number of observations.

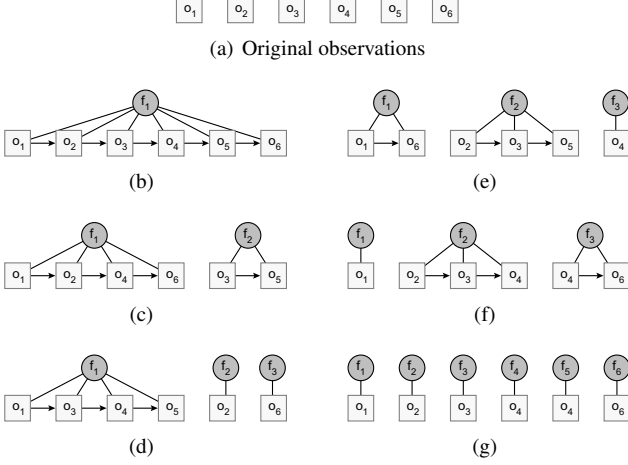


Figure 2. **The Space of Partitions.** In (b)-(g) we show a few of the 203 possible ways to partition the six observations shown in (a). Semantically, (b) refers to the hypothesis that a single object generated all six observations. Similarly, (g) depicts the scenario where each observation was generated by a unique object. The difference between (e) and (f) is the object to which observation  $o_6$  is attributed. Note that in any given trajectory the observations must form a temporally-increasing sequence.

## 4.2. Partition Likelihood

To determine which partitioning of the observations is the most likely, we wish to find the partitioning  $\omega_{MAP} \in \Omega_N$  which maximizes the posterior  $P(\omega|\mathbf{O})$ . Assuming a uniform prior  $P(\omega)$ , we use Bayes' rule to express this posterior in terms of the likelihood

$$P(\omega|\mathbf{O}) = \alpha P(\mathbf{O}|\omega)P(\omega) = \alpha P(\mathbf{O}|\omega). \quad (2)$$

where  $\alpha$  represents normalization terms.

Recall that  $\mathbf{O}_\omega$ , defined in Eq (1), represents the division by  $\omega$  of the complete set of observations,  $\mathbf{O}$ , into  $K_\omega$  disjoint trajectories,  $\mathbf{O}_1, \mathbf{O}_2, \dots, \mathbf{O}_{K_\omega}$ . Assuming independence amongst the object trajectories, the likelihood  $P(\mathbf{O}|\omega) = P(\mathbf{O}_\omega)$  can be factored as a product of the individual trajectory likelihoods

$$P(\mathbf{O}_\omega) = \prod_{k=1}^{K_\omega} P(\mathbf{O}_k) \quad (3)$$

The likelihood of a given trajectory is dependent on various parameters including the object's intrinsic appearance and the camera topology/transition model. As in Zajdel [18], we use a Dynamic Bayes Net (DBN) to evaluate the likelihood of each given trajectory.

We first describe the graphical model representing a single trajectory, illustrated in Figure 3. The intrinsic appearance of object  $k$  is described by the hidden variable  $f_k$ . Each observation  $o_i$  in the trajectory's observation set  $\mathbf{O}_k = \{o_1, o_2, \dots, o_{n_k}\}$  is represented by the observable

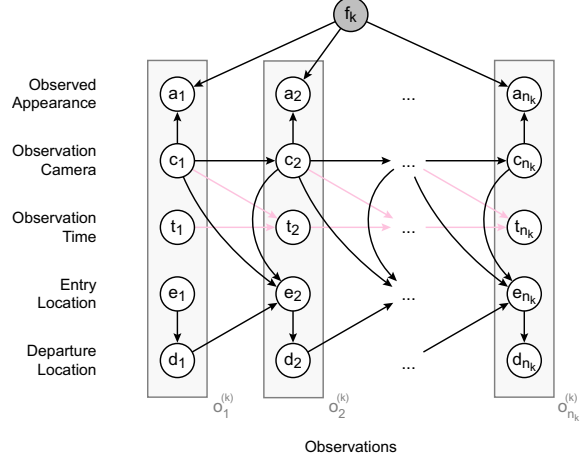


Figure 3. **Graphical Model (Dynamic Bayes Net)** for computing trajectory likelihood and estimating object  $k$ 's intrinsic appearance  $f_k$ .

variables  $a_i$ ,  $c_i$ ,  $t_i$ ,  $e_i$ , and  $d_i$ , described on the left of Figure 3. This graphical model facilitates representing the joint distribution over all variables, by describing conditional dependencies (arrows) between them. The conditional dependencies, represented by PDFs, are described below together with priors for those variables that are not conditioned on others:

- $P(f_k)$  - the prior probability on the intrinsic appearance of an object. See Section 4.2.1 for details.
- $P(a_i|f_k, c_i)$  - the appearance model. The observed appearance depends on both the intrinsic object properties and camera-specific factors such as illumination and occlusion.
- $P(e_i, c_i|d_{i-1}, c_{i-1})$  - the [first-order] transition model/topology. In practice, we approximate this distribution by the product  $P(e_i|c_i, d_{i-1}, c_{i-1}) \cdot P(c_i|c_{i-1})$ .
- $P(t_i|c_i, c_{i-1}, t_{i-1})$  - the distribution of transition times between cameras. While of great utility when object dynamics are highly predicable (see [11, 16]), in other applications where objects stop or disappear for long, uncorrelated periods of time, this term may be of lesser value. This term is presently neglected but is mentioned here for completeness.
- $P(d_i|c_i, e_i)$  - the typical paths within a camera's field of view. As this information is clearly observed within each camera, this term is computed directly from the data.
- $P(c_1)$  - the cameras where an object may enter (or exit) the network. The *peripheral cameras* are incorporated into the model in this manner.

- $P(e_1|c_1)$  - the entry points in the peripheral cameras where an object can first appear; this is learned from the data.

Using these priors and conditional distributions, the DBN allows computation of the trajectory likelihood as

$$P(\mathbf{O}_k) = P(c_1) P(e_1|c_1) \cdot \overbrace{\left( \prod_{i=2}^{n_K} P(e_i, c_i | d_{i-1}, c_{i-1}) \right)}^{\text{Inter-Camera}} \cdot \underbrace{\left( \prod_{i=1}^{n_K} P(d_i | c_i, e_i) \right)}_{\text{Intra-Camera}} \cdot \underbrace{\left( \prod_{i=1}^{n_K} P(a_i | \hat{f}_k, c_i) \right)}_{\text{Appearance}} \quad (4)$$

where  $\hat{f}_k$  is the estimated intrinsic appearance for object  $k$ .

#### 4.2.1 Estimating Intrinsic Appearance

A given partitioning  $\omega$  splits the set of observations  $\mathbf{O}$  into  $K_\omega$  individual trajectories  $\{\mathbf{O}_k\}$ . In Eq (4), the appearance term  $P(a_i | \hat{f}_k, c_i)$ , expresses the measurement likelihood that camera  $c_i$  measures the appearance  $a_i$  (color, etc.) from object  $k$  where object  $k$ 's actual appearance is given by (estimated as)  $\hat{f}_k$ . At present, estimation of the camera-specific influence on measured appearance is not considered.

To facilitate parameter estimation, we model the intrinsic appearance as a Gaussian distribution,  $\hat{f}_k = \mathcal{N}(\mu_k, \Sigma_k)$ , though more descriptive models could be employed. The appearance of each observation is represented by a point in RGB color-space. We compute  $\mu_k$  as the maximum likelihood estimate  $\mu_{ML}$ , equal to the sample mean. We assume a known covariance  $\Sigma_k$  derived from the complete observation set.

#### 4.2.2 Transition Model Parameters

The transition model consists of a known prior  $P(c_1)$  and the conditional dependencies  $P(e_1|c_1)$ ,  $P(e_i, c_i | d_{i-1}, c_{i-1})$ , and  $P(d_i | c_i, e_i)$  which are learned. As we learn the transition model incrementally, starting with just a few observations (see section 4.1), we want to dynamically model the uncertainty, which begins high but gradually decreases as we consider additional observations. To model this uncertainty, we represent the transition model by combining a uniform prior  $T_{unif}$  with the model constructed from the current partitioning of the observations  $T_{data}$

$$T(\tau) = \beta \cdot T_{unif} + (1 - \beta) \cdot T_{data} \quad (5)$$

where  $\beta$  is the exponentially decaying interpolation parameter defined as  $\beta = e^{-4 \frac{m+s\tau}{N}}$  and, as previously,  $\tau$  represents the iteration number ( $0 \leq \tau \leq \lceil \frac{N-m}{s} \rceil$ ). In a given

iteration, the transition model used for the inter- and intra-camera conditional probabilities is  $T^{(\tau-1)}$ , the model resulting from the previous iteration. The updated model  $T^{(\tau)}$  is computed after completing iteration  $\tau$ , only on the partition hypotheses which are retained.

## 5. Experimental Results

We created randomly-generated medium-scale camera networks comprised of 20 cameras placed in the hallways of an indoor office environment (see example in Figure 1). The simulator, implemented in MATLAB, can control the number of objects (people) in the network as well as their behavior: whether they stay primarily in their own office, visit colleagues, how quickly they leave, etc. Each time an object passes through a camera's field of view an observation is recorded, noting the time and image location of the object's entrance and exit, and the measured appearance for the object. Ground truth appearance values are perturbed for each observation by additive Gaussian noise with parameters  $\mathcal{N}(0.5, \sigma_a)$  in each color-space (RGB) dimension.

All observations made within the network are gathered into a single observation set  $\mathbf{O}$ , sorted by entrance time. We then follow the incremental estimation procedure outlined in section 4.1. After beginning with a small initial set of the first  $m$  observations, we iteratively add  $s$  observations, evaluating and keeping only the  $B$  best partitions at each iteration. The iteration continues until the entire set  $\mathbf{O}$  has been considered yielding a final maximum a posteriori partition estimate  $\omega_{MAP}$  and the corresponding transition model  $T_{MAP}$ .

### 5.1. Trajectory Reconstruction

It is critical to accurately reconstruct the original object trajectories. As we will show in sections 5.2 and 5.3, accurate reconstruction of the trajectories ensures accurate estimation of both first- and higher-order topological relationships.

To quantitatively assess trajectory reconstruction, we use two measures: partition *accuracy* and partition *recall* (see [18]). Suppose the true (ground-truth) partition  $\bar{\omega}$  divides the full observation set  $\mathbf{O}$  into  $K_{\bar{\omega}}$  trajectories  $\bar{\mathbf{O}}_i$ ,  $1 \leq i \leq K_{\bar{\omega}}$ . Similarly, the partition estimated by our algorithm  $\hat{\omega}$  produces  $K_{\hat{\omega}}$  trajectories  $\hat{\mathbf{O}}_k$ ,  $1 \leq k \leq K_{\hat{\omega}}$ . The partition *accuracy* denotes the [average] fraction of each recovered trajectory's observations that actually belong to some ground-truth trajectory

$$q_{\hat{\omega}} = \frac{1}{K_{\hat{\omega}}} \sum_{k=1}^{K_{\hat{\omega}}} \frac{\max_i |\bar{\mathbf{O}}_i \cap \hat{\mathbf{O}}_k|}{|\hat{\mathbf{O}}_k|} \cdot 100\% \quad (6)$$

Similarly, the partition *recall* indicates the fraction of each ground-truth trajectory's observations that are partitioned

together in the estimated partition

$$\rho_{\hat{\omega}} = \frac{1}{K_{\hat{\omega}}} \sum_{i=1}^{K_{\hat{\omega}}} \frac{\max_k |\bar{\mathbf{O}}_i \cap \hat{\mathbf{O}}_k|}{|\bar{\mathbf{O}}_i|} \cdot 100\% \quad (7)$$

After using our algorithm to recover object trajectories in several simulated camera networks, we apply these two metrics to the results. Table 1 shows how performance varies with changes in  $B$ , the number of hypotheses retained at each iteration and with  $\sigma_a$ , the appearance noise parameter (see Figure 4 for a visual noise comparison). These results represent average performance across 20 randomly-generated camera networks. Each 20-camera network accumulated observations from ten objects moving through the network with a mean of 32.8 observations collected per object (per network).

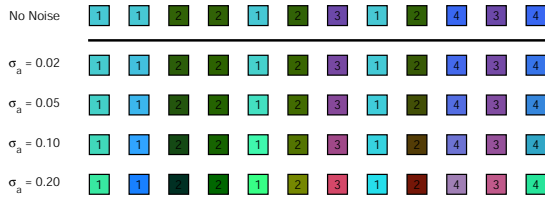


Figure 4. **Visual Noise Comparison.** Twelve initial observations labeled by object identity (the object number is displayed inside each observation’s square). The top row shows the true appearance (color) of each observation as measured without any noise. The four lower rows show how noise of varying levels ( $\sigma_a = \{0.02, 0.05, 0.10, 0.20\}$ ) can change the measured appearance. Note how some objects (*e.g.* the first and last of the twelve observations) can begin to appear similar when measured with high noise, making appearance a less effective discriminant between objects.

## 5.2. First-Order Topology

We next evaluate the algorithm’s recovery of the first-order topology, as done in previous work on topology [11, 16]. Our comparison is based on a stochastic adjacency matrix we call the *topology matrix*. The entries of row  $i$  form a probability distribution, indicating the probability that an object last seen at camera  $i$  will next appear at a particular camera. In theory, the binary matrix formed by replacing the non-zero transition probabilities in the topology matrix with ones would be symmetric (if an object can move from camera  $a$  to  $b$  it should be able to return from  $b$  to  $a$ ). However, while some domains might exhibit true “one-way” paths, in practice there simply may not be any objects taking the reverse path despite its availability.

Results from a simulated 20-camera network (that shown in Figure 1) are presented in Figure 5. This simulation has  $10^3$  objects collectively producing 314 observations.

<sup>3</sup>For real-world observations, one would, of course, need far more than ten tracks to construct a useful model.

Both the ground-truth and recovered topology matrices are shown, together with an error matrix displaying discrepancies between the two. The results show the recovery of almost every camera-camera transition made, and with the correct probabilities in all but a few cases. With the exception of cameras 1 and 4 (the top and fourth rows), all of the spurious estimated transitions are of negligible probability.

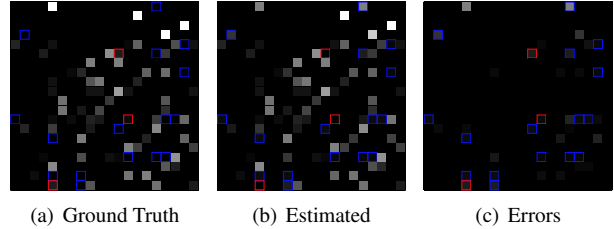


Figure 5. **First-order Topology.** Shown here are the topology matrices induced by (a) the ground truth partition, (b) the estimated best partition, and in (c) the error between the ground truth and estimated topology matrices. The observations used had an appearance noise of  $\sigma_a = 0.05$ , and the parameters used in estimation were  $m = 8, s = 2$ , and  $B = 10$ . Entries framed in red denote non-zero entries in the ground-truth which were entirely lost in the recovery process. Blue-framed entries denote spurious transitions due to estimation errors. Both are generally very low-probability errors.

## 5.3. Higher-Order Topology

Partitioning the observations into full object trajectories enables the extraction of higher-order topological relationships, simply by analyzing the estimated trajectories. With this additional information, we can answer queries such as, “If an object was first observed in camera  $a$  and next in camera  $b$ , what is the likelihood that it will next be seen in camera  $c$ ?”. As we are unaware of other work recovering higher-order transition models, we cannot provide a direct comparison with other algorithms. We therefore present results showing the extent to which our technique is able to accurately recover the second-order transition model. Example second-order transition model estimation results are shown in Figure 6. The increased expressiveness of the second-order model over first-order adjacency can be seen.

Average second-order transition model errors are shown in Figure 7. Errors are computed using sum of squared errors between the ground-truth and the estimated distributions. The probability of an object passing through cameras  $a, b$ , then  $c$  in sequence is expressed as a distribution across camera  $c$  for the given camera pair  $(a, b)$ . The presented results are the average errors over all pairs  $(a, b)$ .

## 6. Future Research

We feel that this technique holds promise for recovering the topology information for camera networks. To more

$B$	$\sigma_a = 0.00$	$\sigma_a = 0.02$	$\sigma_a = 0.05$	$\sigma_a = 0.10$	$\sigma_a = 0.20$
	acc./recall	acc./recall	acc./recall	acc./recall	acc./recall
1	70.0 / 98.4	68.2 / 97.6	63.9 / 94.4	52.7 / 83.0	40.3 / 59.6
2	69.4 / 97.9	69.0 / 97.0	65.5 / 94.1	54.1 / 82.9	39.0 / 60.0
5	69.5 / 98.5	67.7 / 96.8	65.5 / 94.2	54.4 / 84.0	38.8 / 60.3
10	69.7 / 97.5	69.2 / 97.0	65.9 / 94.1	54.2 / 84.6	40.9 / 59.5
25	- / -	69.7 / 97.2	- / -	55.4 / 84.2	- / -

Table 1. **Performance across Hypothesis and Appearance Noise Parameters.** The partition accuracy and recall vary as the number of retained hypotheses  $B$  and the appearance noise parameter,  $\sigma_a$  are changed. In these simulations  $m = 8$  and  $s = 2$  are fixed. The influence of appearance noise on accuracy and recall is substantial, while that of the hypotheses retained is negligible. We believe that accuracy values are lower than recall due to the recovery of too few trajectories. When a new object first appears, if all partitions which attribute it to a new trajectory are pruned then all of its observations will be assigned to existing trajectories.

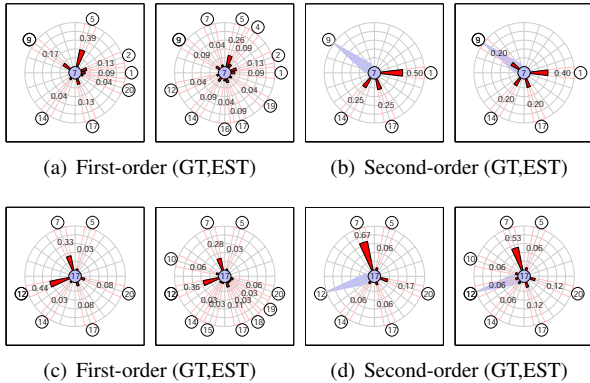


Figure 6. **Higher-order Transition Model Examples.** Two cameras, serve to illustrate the expressiveness of the second-order transition model. Each pair (a)-(d) shows the ground-truth (GT) model on the left and the estimated (EST) model on the right. In these plots, the light blue circle in the center is the camera where the object was last observed (with a blue path indicating where it came from in the second-order model). The red paths indicate probabilities of next appearing at other given cameras, with each visible radial bar proportional in length to its in respective non-zero probability. In (a) and (b) we see that while in general objects at camera 7 most often go to camera 5, the second-order model shows that objects arriving at camera 7 never go to camera 5, rather to cameras 1, 14, and 17. Similarly in (c) and (d), objects at camera 17 are more likely to go to camera 12 than camera 7, however, quite the opposite if they came from camera 12.

fully realize this potential, we propose further work on the following areas:

**Overlapping Field-of-View:** At present we make the assumption that all fields of view are non-overlapping. While it facilitates our present approach, this constraint inhibits the analysis of more general camera networks where cameras may or may not overlap.

**Scalability:** While the approach we present is described as a serial algorithm, it is inherently parallel and could be implemented on a medium-scale network of “smart cameras”, each possessing the computational

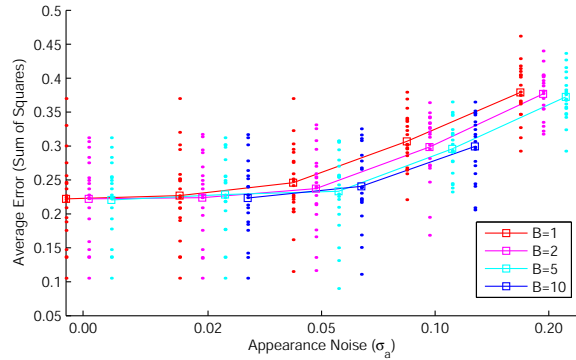


Figure 7. **Higher-order Transition Model Errors.** Second-order transition model errors computed using sum of squared errors between ground-truth and estimated distributions. The points represent the results for the 20 randomly-generated networks, boxes indicate the mean across these networks.

resources to process its own video and also collaborate in distributed topology estimation. (see Bramberger, *et al.* [1] for such a platform). At each iteration, a large number of partitions are evaluated to determine the partition likelihood. The overhead required to divvy up the partitions amongst the camera nodes and gather/prune the results would be minimal (constitutes only 3.80% of the present serial implementation runtime).

**Real Camera Network/Tracking:** We also want to test our data on an actual deployment of 20 or more cameras. We initially plan to apply our algorithm to the 9-camera Terrascope dataset [9] from the U. of Kentucky. We further wish to verify that our higher-order transition model will improve tracking performance.

## 7. Conclusion

We have presented a technique for constructing higher-order statistical transition models. The approach is based on recovering object trajectories by partitioning the observation set in a Bayesian Framework. We described the

Bayesian framework for determining partition likelihood by evaluation of a probabilistic graphical model. We adopt an incremental approach, adding observations and pruning unlikely partitions, retaining only the most probable partitions after each iteration. Having recovered the trajectories we are able to extract not only camera adjacency but also higher-order topological relationships which can improve tracking accuracy and offers topological redundancy, fortifying against camera failures.

**Acknowledgements.** We gratefully acknowledge the support and R&D collaboration with Hasan Ozdemir, and K. C. Lee from the Panasonic System Solution Development Center, USA. We also thank Aniruddha Kembhavi and Vlad Morariu for their contributions.

## References

- [1] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach. Distributed embedded smart cameras for surveillance applications. *IEEE Computer*, 39(2):68–75, 2006.
- [2] L. Chen, M. Çetin, and A. S. Willsky. Distributed data association for multi-target tracking in sensor networks. In *International Conference on Information Fusion*, 2005.
- [3] M. Chu, S. Mitter, and F. Zhao. Distributed multiple target tracking and data association in ad hoc sensor networks. In *International Conference on Information Fusion*, 2003.
- [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. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(2):138–150, 1996.
- [5] M. S. Fox. *Constraint-directed Search: A Case Study of Job-Shop Scheduling*. Morgan Kaufmann Publishers, 1987.
- [6] S. Funiak, C. Guestrin, M. A. Paskin, and R. Sukthankar. Distributed localization of networked cameras. In *IPSN*, pages 34–42, 2006.
- [7] T. Huang and S. J. Russell. Object identification in a bayesian context. In *IJCAI*, pages 1276–1283, 1997.
- [8] O. Javed, Z. Rasheed, K. Shafique, and M. Shah. Tracking across multiple cameras with disjoint views. In *ICCV*, pages 952–957, 2003.
- [9] C. Jaynes, A. Kale, N. Sanders, and E. Grossmann. The terrace dataset: Scripted multi-camera indoor video surveillance with ground-truth. In *PETS05*, pages 309–316, 2005.
- [10] V. Kettner and R. Zabih. Bayesian multi-camera surveillance. In *CVPR*, pages 2253–, 1999.
- [11] D. Makris, T. Ellis, and J. Black. Bridging the gaps between cameras. In *CVPR (2)*, pages 205–210, 2004.
- [12] S. Oh, S. Russell, and S. Sastry. Markov chain monte carlo data association for general multiple-target tracking problems, 2004.
- [13] H. Pasula, S. J. Russell, M. Ostland, and Y. Ritov. Tracking many objects with many sensors. In *IJCAI*, pages 1160–1171, 1999.
- [14] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, 24(6):843–854, 1979.
- [15] C. Stauffer and K. Tieu. Automated multi-camera planar tracking correspondence modeling. In *CVPR (1)*, pages 259–266, 2003.
- [16] K. Tieu, G. Dalley, and W. E. L. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *ICCV*, pages 1842–1849, 2005.
- [17] Y. Yu. Human appearance modeling in visual surveillance. Master’s thesis, University of Maryland, College Park, USA, 2007.
- [18] W. Zajdel. *Bayesian Visual Surveillance*. PhD thesis, Universiteit van Amsterdam, Amsterdam, Nederlands, 2006.
- [19] W. Zajdel, A. T. Cemgil, and B. J. A. Kröse. Dynamic bayesian networks for visual surveillance with distributed cameras. In *EuroSSC*, pages 240–243, 2006.
- [20] W. Zajdel and B. J. A. Kröse. A sequential bayesian algorithm for surveillance with nonoverlapping cameras. *IJPRAI*, 19(8):977–996, 2005.