Tracking many objects with many sensors

Hanna Pasula and **Stuart Russell**
Computer Science Division
University of California, Berkeley
{pasula,russell}@cs.berkeley.edu

Michael Ostland and **Ya'acov Ritov***
Statistics Dept.
University of California, Berkeley
{ostland,ritov}@stat.berkeley.edu

Abstract

Keeping track of multiple objects over time is a problem that arises in many real-world domains. The problem is often complicated by noisy sensors and unpredictable dynamics. Previous work by Huang and Russell, drawing on the data association literature, provided a probabilistic analysis and a threshold-based approximation algorithm for the case of multiple objects detected by two spatially separated sensors. This paper analyses the case in which large numbers of sensors are involved. We show that the approach taken by Huang and Russell, who used pairwise sensor-based *appearance probabilities* as the elementary probabilistic model, does not scale. When more than two observations are made, the objects' *intrinsic* properties must be estimated. These provide the necessary conditional independencies to allow a spatial decomposition of the global probability model. We also replace Huang and Russell's threshold algorithm for object identification with a polynomial-time approximation scheme based on Markov chain Monte Carlo simulation. Using sensor data from a freeway traffic simulation, we show that this allows accurate estimation of long-range origin/destination information even when the individual links in the sensor chain are highly unreliable.

1 Introduction

The problem of tracking multiple objects over time has long been studied in the literature on *data association* [Bar-Shalom and Fortmann, 1988; Bar-Shalom, 1992]. The problem is defined as that of associating a set of current observations with a set of existing "tracks" or object trajectories, creating new tracks as needed. Radar tracking of multiple aircraft is the canonical application. In AI, the problem of *object identification* is essentially the same: deciding if some newly observed object is the same as some previously observed object. Solving this problem is essential for any intelligent agent that reasons about individual objects. Huang and Russell [1997; 1998] provide a fairly general formulation of the problem and describe an application to traffic surveillance. Other possible applications range from removing "duplicate" entries from databases to re-recognizing locations during exploratory map-building.

The object identification problem is difficult because sensors are noisy, objects look similar, and object behaviors are unpredictable. This leads to a large number of possible *assignments* specifying identities among observed objects. For example, in the traffic surveillance application studied by Huang and Russell, the sensors are cameras at various locations on a freeway network and the objects are vehicles. Many thousands of vehicles pass each camera, and the system must decide whether each vehicle is a new vehicle or the same as one previously observed at a different location. Over time, these decisions give rise to hypothetical vehicle *trajectories*. Deriving a set of trajectories is the first step of many traffic surveillance applications, such as the average *link travel time* between locations or *origin-destination counts* along different routes through the system. Moreover, sudden changes in these quantities can be indicators of highway incidents, such as accidents or breakdowns.

Adopting the notation used by Huang and Russell, let $\omega \in \Omega$ denote an assignment placing pairs (or, more generally, sets) of observed objects into equivalence classes, where each class represents an existing object, and let $\mathbf{O}$ denote all observations made to date. Then the posterior probability that two objects a and b are the same is given by

$$\begin{aligned} P(a=b|\mathbf{O}) \\ &= \sum_{\omega \in \Omega} P(a=b|\omega, \mathbf{O}) P(\omega|\mathbf{O}) \\ &= \sum_{\omega \in \Omega : (a,b) \in \omega} P(\omega|\mathbf{O}) \qquad (1) \end{aligned}$$

In evaluating terms such as $P(\omega|\mathbf{O})$, we will make use of the fact that the prior $P(\omega)$ can be assumed uniform.

*Permanent address: Statistics Dept., Hebrew University of Jerusalem.

This is because the probability of an assignment ω, in the absence of observations linking the objects, must be invariant under renaming of the objects. This it the *exchangeability* assumption of Huang and Russell.

Other quantities can also be calculated by summing over ω. For example, in freeway surveillance, ω specifies the correspondence between vehicles observed at upstream and downstream sensor locations. For a given ω, the average link travel time $LTT(\omega)$ between the two locations can be calculated directly if the observations include the arrival time at each location. Then the posterior expectation of the link travel time is

$$E(LTT|\mathbf{O}) = \sum_{\omega \in \Omega} LTT(\omega)P(\omega|\mathbf{O}).$$

This paper addresses the two principal difficulties that arise in putting such equations into practice.

Section 2 deals with the computation of the $P(\omega|\mathbf{O})$ terms—in particular their decomposition into tractable local models that can be estimated from data. We show that the decomposition proposed by Huang and Russell using *appearance probabilities*, while adequate for the case of two sensor locations, does not scale up to handle the decomposition of a global model for many sensors. In fact, this appears to require the estimation of intrinsic parameters of the observed objects, which render successive observations conditionally independent.

Section 3 deals with the intractability of the summation in Eq. (1), which includes an exponential number of terms. Whereas Huang and Russell describe a heuristic scheme that seems to work well in practice, we apply the Markov chain Monte Carlo (MCMC) method, a general-purpose approximation algorithm for probabilistic inference that can be shown to converge in polynomial time for the specific inference problem involved in object identification. Furthermore, the algorithm can be adapted easily to incorporate online updating of the probability models required for computing $P(\omega|\mathbf{O})$. The overall scheme is in fact an online EM algorithm with MCMC as an approximate E-step.[1]

Section 4 describes an application of the new approach to data extracted from a freeway simulation. We show that the estimation of intrinsic parameters, as described in Section 2, successfully handles some multi-camera scenarios for which the appearance probability models of Huang and Russell are not applicable. We also show that the MCMC method allows accurate estimation of long-range origin/destination information even when the individual links in the sensor chain are highly unreliable.

2 Scaling up to multiple sensors

As mentioned in the Section 1, calculation of assignment probabilities $P(\omega|\mathbf{O})$ is crucial for object identification. The calculation will be done using probability models that, in some way, capture the properties of the sensors and the behavior of the objects being tracked. We hope to find models that allow decomposition of the global assignment probability in much the same way that local causal models allow decomposition of joint probabilities in Bayesian networks.

We begin by describing the models used by Huang and Russell, explaining how they work only for two sensor locations. We then describe an alternative approach based on estimation of intrinsic properties of objects.

[1] The use of MCMC with EM is well-known in statistics [Wei and Tanner, 1990]; to our knowledge, its use for online learning and for object identification is novel.

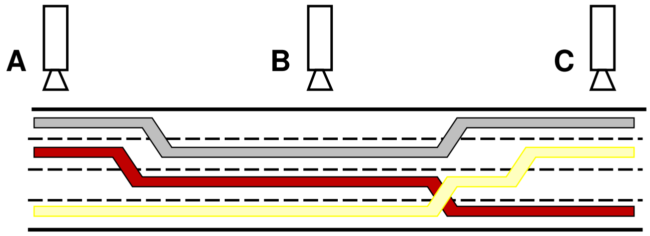

Figure 1: Schematic diagram showing three consecutive camera sites, A, B, and C, and three vehicle trajectories.

2.1 Problems with appearance probabilities

We begin with the case of two consecutive cameras A and B as considered by Huang and Russell. Let the observations at each be $\mathbf{O}^A = \{o_1^A, \ldots o_m^A\}$ and $\mathbf{O}^B = \{o_1^B, \ldots o_n^B\}$, and let ω_{AB} be an assignment pairing up objects observed at A with objects observed at B.[2] Then

$$\begin{aligned} P(\omega_{AB}|\mathbf{O}) &= P(\omega_{AB}|\mathbf{O}^A, \mathbf{O}^B) \\ &= \alpha P(\mathbf{O}^B|\mathbf{O}^A, \omega_{AB}) P(\omega_{AB}|\mathbf{O}^A) \\ &= \alpha P(\mathbf{O}^B|\mathbf{O}^A, \omega_{AB}) \\ &\approx \alpha \prod_{(a,b) \in \omega_{AB}} P(o_b^B|o_a^A, a=b) \end{aligned} \tag{2}$$

where α is, again, a normalizing constant. The term $P(\omega_{AB}|\mathbf{O}^A)$ is dropped by exchangeability—conditioning on the initial observations provides no information about matching with the subsequent objects. The approximate equality in the last line arises from the assumption of approximate independence among vehicle trajectories.[3]

In Eq. (2), the terms $P(o_b^B|o_a^A, a=b)$ are called *appearance probabilities* since they describe "how an object can be expected to appear at subsequent observations given its current appearance" [Huang and Russell, 1997]. The appearance probability models for freeway vehicles are composed of factors such as the arrival time at B given the arrival time at A, the measured colour at B given the measured colour at A, and so on. Huang and Russell show how these models can be estimated online from

[2] Since in general $m \neq n$ and conservation of objects is not assumed, ω_{AB} may include unpaired objects from either or both sensors.

[3] Trajectory independence is a reasonable assumption only if the individual models are conditioned on some global context variables such as the current link travel time.

matched vehicles in a very straightforward way, avoiding the need for camera calibration.

Let us now extend this approach to three cameras, using the scenario of (Figure 1) as an example. An assignment ω_{ABC} now specifies sequences of three observations that belong to a single object, and can be decomposed into two pairwise assignments ω_{AB} and ω_{BC}. As in Eq. (2), we can apply Bayes' rule and eliminate $P(\omega_{ABC}|\mathbf{O}^A)$; then we can apply the chain rule:

$$\begin{aligned} P(\omega_{ABC}|\mathbf{O}) &= \alpha P(\mathbf{O}^B, \mathbf{O}^C|\mathbf{O}^A, \omega_{ABC})P(\omega_{ABC}|\mathbf{O}^A) \\ &= \alpha P(\mathbf{O}^C|\mathbf{O}^A, \mathbf{O}^B, \omega_{ABC})P(\mathbf{O}^B|\mathbf{O}^A, \omega_{ABC}) \\ &= \alpha P(\mathbf{O}^C|\mathbf{O}^A, \mathbf{O}^B, \omega_{ABC})P(\mathbf{O}^B|\mathbf{O}^A, \omega_{AB}) \quad (3) \end{aligned}$$

where ω_{ABC} is replaced in the last line by ω_{AB} because assignments of vehicles at C carry no information about A and B. Now the last term on the RHS of Eq. (3) can be written as the product of appearance probabilities between A and B, as in Eq. (2). However, the first term *cannot* be simplified to give the appearance probabilities between B and C; that is,

$$P(\omega_{ABC}|\mathbf{O}) \neq \alpha P(\mathbf{O}^C|\mathbf{O}^B, \omega_{BC})P(\mathbf{O}^B|\mathbf{O}^A, \omega_{AB})$$

To see why, consider the extreme case in which cameras A and C can read the license plate of each vehicle, but camera B is broken. Then the posterior distribution for assignments should have all its mass on assignments ω_{ABC} that correctly match up vehicles at A and C; whereas both the pairwise models will be uninformative and hence will fail to propagate information from A to C.

Two possible fixes are 1) use multicamera models, e.g., $P(\mathbf{O}^C|\mathbf{O}^A, \mathbf{O}^B, \omega_{ABC})$, and 2) estimate models for all camera pairs, e.g., $P(\mathbf{O}^C|\mathbf{O}^A, \omega_{AC})$. The first fix enlarges the model dimension and scales exponentially with the number of cameras. The second fix requires a quadratic number of models; moreover, it is unclear how to combine the predictions of these models. In summary, it seems that, despite their many advantages, appearance probability models apply only to the two-camera case.

2.2 Spatial decomposition via intrinsic properties

The example of the broken camera at B raises the following problem: we wish to propagate information between nonadjacent sensors, yet we do not want to have to employ nonlocal probability models, since such models result in a combinatorial explosion in the number of parameters to be estimated. The solution is, essentially, to let the objects themselves carry the necessary information. As with Kalman filters and hidden Markov models, hidden state variables can render current observations conditionally independent of previous observations. This provides a decomposition of the global model.

Let $\mathbf{H}^l$ represent the hidden state of the objects observed at location l, and let h^l range over possible values of $\mathbf{H}^l$. Notice that, given the hidden state, the observations at a camera are independent of all the other observations. More formally, $P(\mathbf{O}^j|\mathbf{O}^k, \mathbf{H}^j) = P(\mathbf{O}^j|\mathbf{H}^j)$

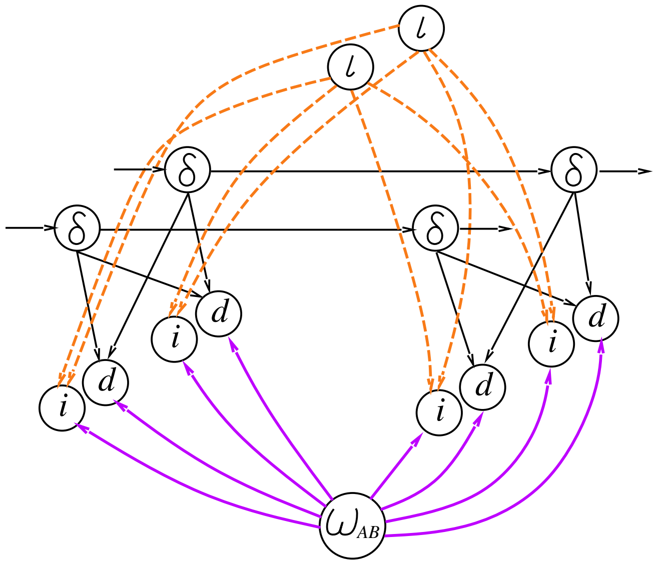

Figure 2: Graphical model for object identification inference, showing two objects at two sensor locations.

for all j and k. Now, we can introduce hidden state into Eq. (3) by summing over h^A, h^B, and h^C and simplifying using conditional independence, to yield a nested sum exactly as in the derivation of the forward equations for HMMs. The only significant difference is that the relationship between successive hidden state variables is only meaningful if we condition on the assignment so we know which vehicle is which:

$$\begin{aligned} &P(\omega_{ABC}|\mathbf{O}) = \\ &\quad \alpha \sum_{h^A} P(h^A)P(\mathbf{O}^A|h^A) \\ &\quad\quad \sum_{h^B} P(h^B|h^A, \omega_{AB})P(\mathbf{O}^B|h^B) \\ &\quad\quad\quad \sum_{h^C} P(h^C|h^B, \omega_{BC})P(\mathbf{O}^C|h^C) \qquad (4) \end{aligned}$$

Thus, the introduction of hidden state solves the problem of Section 2.1. Unlike Eq. (3), Eq. (4) calls only for two-camera models. Moreover, these models need be estimated only for neighbouring camera pairs.

As with appearance probabilities, we can decompose the expressions in Eq. (4) into models for individual vehicles by making the appropriate independence assumptions (again, assuming the models depend on some global context variables). The models we obtain are the *transition models* such as $P(h_b^B|h_a^A, a=b)$ and the *sensor models* such as $P(o_a^A|h_a^A)$. As with HMMs, these models can be learned online using EM [Dempster *et al.*, 1977], as we show below. The process is complicated by the fact that we must simultaneously estimate both the hidden state variables of the observed objects and the global assignment saying which object is which.

For traffic surveillance and many other applications, some aspects of the hidden state do not change over time.

We call these *intrinsic* variables; for traffic surveillance, these include colour, length, width, and so on. Intrinsic variables have no transition model but often have very noisy sensor models, specific to each sensor location. *Dynamic* variables such as lane, speed, and arrival time must be tracked as the vehicle progresses through the freeway network. Often they have relatively noiseless sensor models. Thus, the hidden variables h can be divided into the intrinsic variables, ι, and the dynamic variables, δ. Similarly, the observed variables O can be divided into i, observations of ι, and d, observations of δ. Figure 2 illustrates all the independence assumptions we have made. It is relatively easy to augment this model to include dependencies between, for example, vehicle size and lane.

3 Approximate inference and online model updating

As mentioned in Section 1, the expressions for the probability of identity (Eq. 1) and other quantities involve exponentially many terms. It can be shown that the inference problem is equivalent to computing the permanent of a matrix and hence is #P-complete. It is possible to compute the *most probable* assignment for n vehicles in $O(n^3)$ time using the Hungarian algorithm [Cox and Hingorani, 1994]; as Huang and Russell point out, however, this assignment may be of little interest if the individual matches therein are highly unreliable. They developed a heuristic "leave-one-out" algorithm with runtime $O(n^4)$ that alleviates this problem and works well in practice. The algorithm identifies individual matches that exceed a reliability threshold and then treats those matches as if true. The matches are used to compute link travel time and to update the appearance probability models. A major drawback of this approach is that the if the fraction of reliably matched vehicles on each link is significantly below 100%, as often happens, the number of vehicles that can be tracked across a multi-link freeway network is vanishingly small [Huang and Russell, 1998].

In this section, we describe an alternative approach to the inference problem based on Markov chain Monte Carlo (MCMC). Roughly speaking, MCMC approximates sums of probabilities such as Eq. (1) by sampling a small number of high-probability terms; thus, for object identification, it considers a small number of likely assignments and likely values for the hidden state variables. Given any particular assignment and set of values for the hidden variables, estimating the transition and sensor models is trivial because the data is complete. This suggests an online EM scheme for learning the models, as shown in Figure 3.

3.1 Introduction to MCMC inference

MCMC inference is a general-purpose method for approximating $E_\pi(f(x))$, the expected value of the function f when its argument x is drawn from a probability distribution π. Typically, $\pi(x) = P(x|e)$, a posterior distribution over x given evidence e.

```
For each newly detected vehicle
  Augment Markov chain state to include the vehicle
  Repeat until models converge
    E: Run MCMC, sampling from Ω and H
    M: Update models from sampled values
```

Figure 3: The overall inference scheme: online EM using MCMC for the E-step and updating the models directly from the states sampled by MCMC.

Ideally, $E_\pi(f(x))$ can be approximated simply by *sampling* from π; if each sample can be drawn in constant time, then from Chernoff bounds we will have a polynomial-time approximation method. For general π, we know this cannot exist; sampling from an arbitrary distribution in constant time is not always possible.

MCMC inference [Metropolis *et al.*, 1953; Gilks *et al.*, 1996] provides a general method to generate samples from π by defining a Markov chain whose states are the objects x and whose *stationary distribution* is $\pi(x)$. Samples are produced by simulating the Markov chain and selecting states from among those visited.

In the *Metropolis-Hastings* method, transitions in the Markov chain are constructed in two steps:

- Given the current state x, a candidate next state is sampled from the *proposal distribution* $q(x'|x)$, which may be (more or less) arbitrary.
- The transition to x' is not automatic, but occurs with an *acceptance probability* defined by

$$\alpha(x'|x) = min\left(1, \frac{\pi(x')q(x|x')}{\pi(x)q(x'|x)}\right)$$

  Notice that to use this rule we need only be able to compute *ratios* $\pi(x')/\pi(x)$, which conveniently avoids the need to normalize π.

Provided q is defined in such a way that the chain is ergodic, this transition mechanism defines a Markov chain whose stationary distribution is $\pi(x)$, and hence the average value of f over the sampled states will converge to the desired value $E_\pi(f(x))$.

The complexity of the original inference problem is reflected in the *mixing rate* of the Markov chain, which determines the speed at which the sample average converges. Jerrum and Sinclair [1997] have shown that a Markov chain defined on assignments, as described below, yields a *fully randomized approximation scheme* for estimating expectations over the probability distribution on assignments. This means that, if we want to approximate some such function f to within ratio $1+\epsilon$ in a world with n objects, the chain will run in time polynomial in n and $1/\epsilon$ to approximate f with probability > 0.75. The probability may be boosted to $> (1-\delta)$ by running the chain $O(\log \delta^{-1})$ times and taking the median value. Of course, this assumes that proposals and acceptance probabilities can be computed in time constant in n, which we show below. Therefore, MCMC provides an efficient approximation scheme for object identification.

3.2 Application to traffic surveillance

As we have seen in Section 1, traffic surveillance, as defined here, involves taking the expected value of a quantity f over all assignments ω, given the observations $\mathbf{O}$. The calculation of this expected value, $E_{P(\omega|\mathbf{O})}(f(\omega))$, can be approximated by sampling from an ergodic Markov chain with state space Ω and stationary distribution $P(\omega|\mathbf{O})$. The quantities to be estimated, which include LTTs and origin destination counts, are very simple to derive for any given assignment ω. An assignment specifies trajectories through the sensor network with known times at each sensor location, enabling us to read off the desired quantity directly.

The transitions in the Markov chain may be set up in many ways, as long as ergodicity is ensured. In our approach, each transition is simply a swap between two assignment pairs across one pair of sensors. For example, in Figure 4 observed object w at sensor j is originally matched with observed object y at sensor $j+1$, while object x at j is matched with z at $j+1$. A transition leads to w being matched with z, and x with y. The

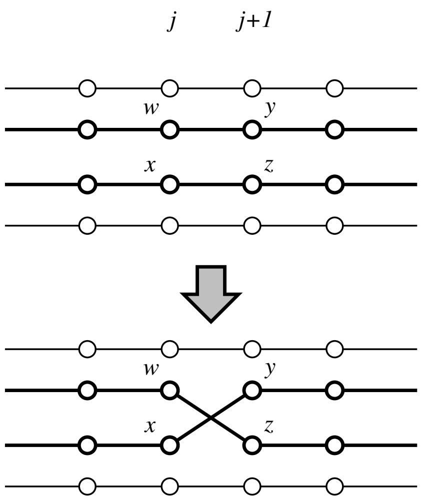

Figure 4: Figure demonstrating a single transition from one assignment with four trajectories to another.

observation pairs to be switched are suggested in a very simple manner. The algorithm cycles through all pairs of adjacent sensors, and all matched pairs currently at each sensor. For each such pair, a plausible second pair to swap with is then chosen uniformly at random. Such a chain is provably ergodic.

Once a pair is chosen, it is accepted or rejected based on the acceptance probability as Eq. (3.1). This involves the computation of the ratio

$$\frac{P(\omega'|\mathbf{O})q(\omega|\omega')}{P(\omega|\mathbf{O})q(\omega'|\omega)}$$

Because of the simple form of the swapping proposals, the q values are trivial to compute. We can also derive a general expression for $P(\omega|\mathbf{O})$, which will permit us to simplify the $P(\omega'|\mathbf{O})/P(\omega|\mathbf{O})$ ratio and make its calculation more efficient. Let us demonstrate this using the transition in Figure 4.

Generalizing Eq. (4) for the observation sequence $\mathbf{O}^1, \mathbf{O}^2 \ldots \mathbf{O}^k$, we express $P(\omega_{1\ldots k}|\mathbf{O})$ as a nested sum

$$\begin{aligned} P(\omega_{1\ldots k}|\mathbf{O}) &= \alpha \sum_{h^1} P(h^1)P(\mathbf{O}^1|h^1) \times \ldots \\ &\ldots \times \sum_{h^{j+1}} P(h^{j+1}|h^j, \omega_{j,j+1})P(\mathbf{O}^{j+1}|h^{j+1}) \times \ldots \\ &\ldots \times \sum_{h^k} P(h^k|h^{k-1}, \omega_{k-1,k})P(\mathbf{O}^k|h^k) \end{aligned} \tag{5}$$

As was noted in Section 2.2, the hidden variables can be divided into dynamic variables δ, which change over time, and intrinsic variables ι, which do not. The observed variables are analogously divided into d and i. Here, we assume that all the dynamics can be observed reliably. Thus, assuming that ι and δ are mutually independent, $P(h^{j+1}|h^j, \omega_{j,j+1})$ becomes $P(\iota^{j+1}|\iota^j, \omega_{j,j+1})P(\delta^{j+1}|\delta^j, \omega_{j,j+1})$, and $P(O^j|h^j)$ becomes $P(i^j|\iota^j)P(d^j|\delta^j)$. Now, we can make use of the assumption that the intrinsic variables remain constant across all sensors to discard the $P(\iota^{j+i}|\iota^j, \omega_{j,j+1})$ terms. Similarly, since we assume that dynamic variables are observed with perfect reliability, we can replace δs with ds. The $P(d^j|\delta^j)$ terms can then be dropped. So, given our assumptions, Eq. (6) simplifies as follows:

$$\begin{aligned} P(\omega_{1\ldots k}|\mathbf{O}) &= \alpha \sum_{\iota^1} P(\iota^1)P(i^1|\iota^1)P(d^1) \times \ldots \\ &\ldots \times \sum_{\iota^{j+1}} P(d^{j+1}|d^j, \omega_{j,j+1})P(i^{j+1}|\iota^{j+1}) \times \ldots \\ &\ldots \times \sum_{\iota^k} P(d^k|d^{k-1}, \omega_{k-1,k})P(i^k|\iota^k) \end{aligned} \tag{6}$$

Finally, note that the only hidden variables we are summing over now are the intrinsics ι. Moreover, ι is the same across all cameras by assumption, so we can combine all the summations. The $P(d^j|d^{j-1}, \omega_{j-1,j})$ terms are independent of ι, and so can be moved outside the summation

$$\begin{aligned} &P(\omega_{1\ldots k}|\mathbf{O}) \\ &\quad = \alpha P(d^1)P(d^2|d^1, \omega_{1,2}) \ldots P(d^k|d^{k-1}, \omega_{k-1,k}) \\ &\qquad \sum_{\iota} P(\iota)P(i^1|\iota)P(\iota)P(i^2|\iota) \ldots P(i^k|\iota) \\ &\quad = \alpha P(d^1) \left(\prod_{j=1}^{k-1} P(d^{j+1}|d^j, \omega_{j,j+1}) \right) \\ &\qquad \sum_{\iota} P(\iota) \left(\prod_{j=1}^{k} P(i^j|\iota) \right) \end{aligned} \tag{7}$$

We can now use trajectory independence to factor this equation into separate terms, one for each trajectory. Let the variable t represent a trajectory within a specified ω and let ι_t represent the hidden intrinsic variables for the putative object that follows trajectory t. Define i_t and d_t analogously.

Eq. (7) may now be factored into a product of probabilities for individual trajectories:

$$P(\omega_{1\ldots k}|\mathbf{O}) = \alpha \prod_{t \in \omega} S(t) P(d_t^1) \prod_{j=1}^{k-1} P(d_t^{j+1}|d_t^j, \omega_{j,j+1})$$

where

$$S(t) = \sum_{\iota_t} P(\iota_t) \prod_{j=1}^{k} P(i_t^j|\iota_t)$$

We can now evaluate the ratio $P(\omega'|\mathbf{O})/P(\omega|\mathbf{O})$ in a simplified form. As shown in Figure 4, each Markov chain transition affects only two trajectories, those that include w, x, y, and z. Clearly, the probabilities along the remaining trajectories remain unchanged. Thus, $P(\omega'|\mathbf{O})$ and $P(\omega|\mathbf{O})$ will share many common factors. All these factors, including the normalization constant α, cancel out. If we let xy signify the trajectory which includes x and y, the ratio simplifies to

$$\frac{P(d_z|d_w, z=w)P(d_y|d_x, y=x)S(wz)S(xy)}{P(d_y|d_w, y=w)P(d_z|d_x, z=x)S(xz)S(wy)}$$

This ratio can be computed in time proportional to the longest network trajectory, which is constant for any given network. Note that the bottleneck of the computation is the evaluation of the $S(t)$s, integrals over products of intrinsic noise models. Currently, all of these models are Gaussian, and so the products are Gaussian and integration is not difficult. However, some care is required when introducing other models, to ensure that the computation is still efficient. Similar problems may be introduced if the assumption of noise-free dynamics is abandoned. Without this assumption, we will need to integrate over the $P(d|\delta)$s as well as over the $P(i^j|\iota)$s. This can still be efficient, as long as the models are chosen well.

3.3 Model updating

The dynamic variable models at each link and the intrinsic variable models at each sensor are all learnt using EM. A model update is performed whenever a new object is observed at a sensor, and the MCMC process has been given time to converge. Each EM iteration proceeds as follows.

The E-step requires computing joint expectations for the hidden variables ω and I; with MCMC, this is approximated by sampling ω from the Markov chain and extending each sample by calculating $E(I|\omega, \mathbf{O})$. Again, the fact that all our intrinsic models are Gaussian yields a relatively straightforward calculation that requires time proportional to trajectory length. For example, consider a sequence of vehicle length measurements with a uniform prior and a sequence of Gaussian noise models with a common variance. In this situation, the expected true vehicle length is just the mean of all the observations. If other, non-Gaussian, models are used, the computation may become much more complicated—in which case, sampling over I as well as over ω may be the best approach.

The M step is exact. It uses the ω and I values of each sample as if they were observed variables to perform conventional parameter learning.

Running EM to its convergence requires a Markov chain run for each EM iteration, and we run EM whenever a new observation is made, resulting in many Markov chain runs. In practice, the chain converges very quickly, as the models change only slowly with each incoming observation. We are currently investigating the use of online EM, where only a single iteration is performed for each new available data point. Nowlan [1991] has proved that this approach should lead to locally maximum likelihood estimates in the limit, and our preliminary experimental results are encouraging.

4 Experimental results

We have performed two experiments comparing our approach to that of Huang and Russell. Our data sets were created using a freeway simulator [Forbes *et al.*, 1995] that allowed us to control road configurations, camera characteristics, and complex vehicle behaviour. Each set included the observations generated by approximately one hundred cars, and the models used in the experiments had been learnt by our EM-MCMC algorithm using data generated with the same parameter settings.

4.1 Overcoming faulty sensors

In this experiment, the algorithms were applied to a three-camera network such as that in Figure 1, set up in a manner analogous to the license-plate example of Section 2.1. The only intrinsic attribute used was colour, and every car had a unique colour. The outer two cameras, A and C, measured colour exactly. Gaussian noise was added to the measured colour at B. Thus, the data set contained enough data for a complete matching between cameras A and C, but a pairwise appearance-based model could be expected to lose accuracy due to its hasty independence assumptions.

The results of the experiment, averaged over three runs, can be seen in Figure 5. The y–axis requires a little explanation. In the case of the Huang and Russell algorithm, which results in discrete matches, it shows simply the percentage of correct A to C matches: in the case of the MCMC, it shows the average percentage of correct matches across all the samples drawn. The advantage of MCMC is clear: hidden feature estimation helps to maintain an almost constant degree of accuracy, whereas the pairwise approach is highly sensitive to sensor noise.

4.2 Estimating origin/destination counts

Here, we compare the algorithms in the slightly more realistic setting shown in Figure 6. The aim is to estimate the origin-destination counts between the two entry points and the three exit points. This requires tracking each object across the entire network. In this task, the

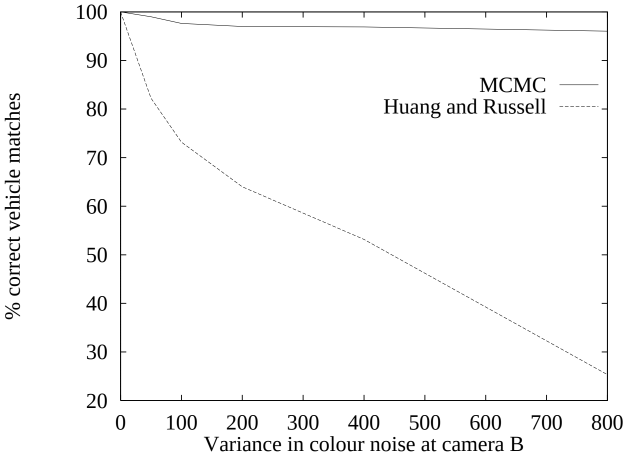

Figure 5: The vehicle matching accuracy of two algorithms as a function of the variance in the Gaussian noise at the central sensor.

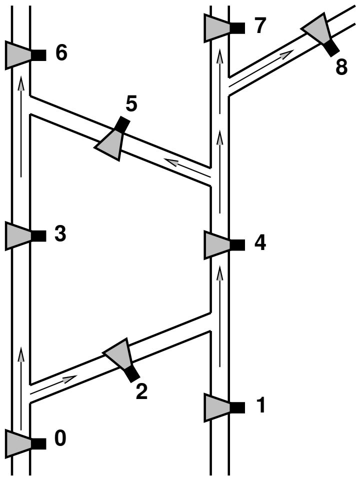

Figure 6: Schematic diagram of simulated freeway network with nine cameras.

greater number of sensors should be a liability to the pairwise algorithm, as each pair of neighbouring sensors can make its own mistakes independently of the others. Our algorithm, on the other hand, should benefit from the ability to observe each object several times to estimate intrinsics more reliably.

The colour variance was manipulated as before, but at all nine cameras simultaneously. Figure 7 shows the percentage accuracy of the origin-destination counts as a function of the colour noise variance. As expected, the MCMC algorithm substantially outperforms the Huang-Russell algorithm, holding up well even for levels of noise that essentially wipe out the colour altogether. On the other hand, Figure 8 shows that both methods are unable to find exact matches accurately for high levels of noise. The ability of the MCMC algorithm to recover reasonable counts despite the failure of individual matches suggests that its samples contain a reasonable amount of information about the ensemble behaviour of the vehicles.

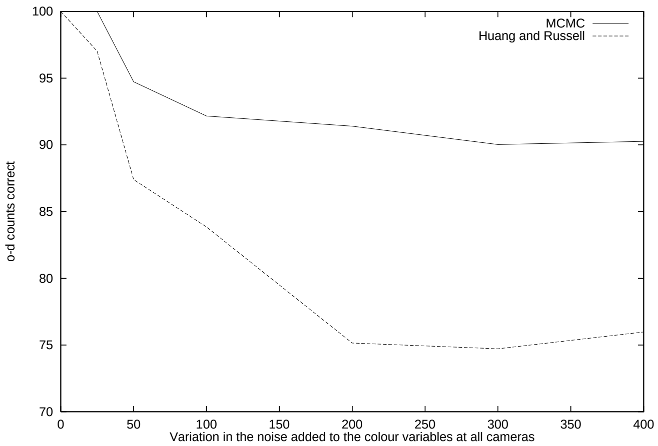

Figure 7: The origin-destination count accuracy of the two algorithms as a function of the colour noise variance.

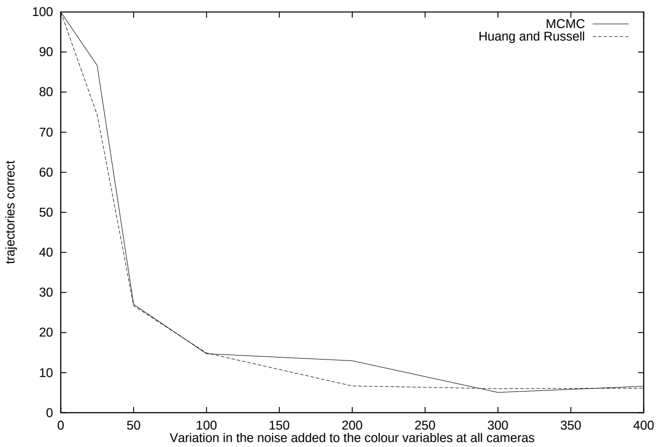

Figure 8: The origin-destination vehicle matching accuracy of the two algorithms as a function of the colour noise variance.

5 Summary and future work

We have described an improved approach to object identification based on the estimation of the intrinsic properties of objects and the use of Markov chain Monte Carlo to approximate the posterior probabilities efficiently. We have shown that this approach works on computer-generated data, and that its computational requirements are not prohibitive. We believe that this approach should be applicable in many data association applications.

We are currently working towards applying our approach in the real world. To this end, we have extended the approach to handle realistic problems such as missing observations. In the near future, we will receive data on vehicle observations made by cameras placed above the I-80 freeway as part of the Berkeley Roadwatch project.

Solving the tracking and data association problems for individual vehicles is only the first part of the solution for large, complex applications such as traffic surveillance. The next step is to connect these low-level computations to high-level, aggregated models, which will permit prediction and control for large networks containing hundreds of thousands of vehicles.

References

[Bar-Shalom and Fortmann, 1988] Yaakov Bar-Shalom and Thomas E. Fortmann. *Tracking and Data Association*. Academic Press, New York, 1988.

[Bar-Shalom, 1992] Yaakov Bar-Shalom, editor. *Multitarget multisensor tracking: Advanced applications*. Artech House, Norwood, Massachusetts, 1992.

[Cox and Hingorani, 1994] I. J. Cox and S. L. Hingorani. An efficient implementation and evaluation of Reid's multiple hypothesis tracking algorithm for visual tracking. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, volume 1, pages 437–442, Jerusalem, Israel, October 1994.

[Dempster *et al.*, 1977] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39 (Series B):1–38, 1977.

[Forbes *et al.*, 1995] Jeff Forbes, Tim Huang, Keiji Kanazawa, and Stuart Russell. The BATmobile: Towards a Bayesian automated taxi. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, Montreal, Canada, August 1995. Morgan Kaufmann.

[Gilks *et al.*, 1996] W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, editors. *Markov chain Monte Carlo in practice*. Chapman and Hall, London, 1996.

[Huang and Russell, 1997] Tim Huang and Stuart Russell. Object identification in a Bayesian context. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97)*, Nagoya, Japan, August 1997. Morgan Kaufmann.

[Huang and Russell, 1998] Tim Huang and Stuart Russell. Object identification: A Bayesian analysis with application to traffic surveillance. *Artificial Intelligence*, 103:1–17, 1998.

[Jerrum and Sinclair, 1997] M. Jerrum and A. Sinclair. The Markov chain Monte Carlo method. In D. S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*. PWS Publishing, Boston, 1997.

[Metropolis *et al.*, 1953] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1091, 1953.

[Nowlan, 1991] S. J. Nowlan. *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures*. PhD thesis, School of Computer Science, Carnegie Mellon University, 1991.

[Wei and Tanner, 1990] G. C. G. Wei and M. A. Tanner. A Monte-Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *Journal of the American Statistical Association*, 85(411):699–704, 1990.