Tracking many objects with many sensors

Content Areas: probabilistic reasoning, perception, distributed AI
Tracking Number: A564

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 large 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 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 the same as one previously observed or a new vehicle.

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, and let $\mathbf{O}$ denote all observations made to date. Then the posterior probability that two objects a and b are the same is

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

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 large-scale 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 preceding introduction, 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, showing how they fail to scale up to more than two sensor locations. We then describe an alternative approach based on estimation of intrinsic properties of objects.

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 $\mathbf{O}^A = \{o_1^A, \dots o_m^A\}$ and $\mathbf{O}^B = \{o_1^B, \dots o_n^B\}$, and let ω_{AB} be an assignment pairing up objects observed at A with objects observed at B.[2] Then

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

where α is a normalizing constant. Notice that the $P(\omega_{AB})$ term that would normally appear in the first line disappears because the assumption of exchangeability implies a uniform prior over ω. The approximate equality in the second 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 color at B given the measured color at A, and so on. Huang and Russell show how these models can be estimated online from matched vehicles in a very straightforward way, avoiding the need for camera calibration.

Let us now extend this approach to three cameras (Figure 1). 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 $P(\omega_{AB})$; then we can apply the chain rule:

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

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 before. However, the first term *cannot* be simplified to give the appearance probabilities between B and C; that is,

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

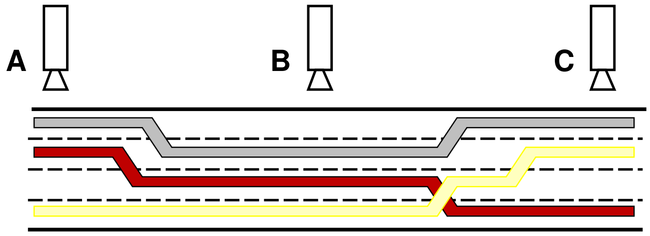

Figure 1: Schematic diagram showing three consecutive camera sites, A, B, and C, and three vehicle trajectories.

[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.

[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.

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 the pairwise models will both be uninformative and hence will fail to propagate information from A to C.

Two possible fixes are 1) use multicamera models, e.g., $P(\mathbf{C}|\mathbf{A}, \mathbf{B}, \omega_{ABC})$, and 2) estimate models for all camera pairs, e.g., $P(\mathbf{C}|\mathbf{A}, \omega_{AC})$. 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 that pass location l, and let h^l range over possible values of $\mathbf{H}^l$. Then we can introduce hidden state into Eq. (3) by summing over h^A, h^B, and h^C and simplifying using conditional independence, 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:

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

As with appearance probabilities, we can decompose the expressions in Eq. (5) 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 color, 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.

```
For each newly detected vehicle
  Repeat until models converge
    E: Run MCMC, sampling from Ω and H
    M: Update models from sampled values
```

Figure 2: The overall inference scheme: online EM using MCMC for the E-step and updating the models directly from the states sampled by MCMC.

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 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 2.

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.

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 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. For a Markov chain defined on assignments, as described below, Jerrum and Sinclair [1997] have shown that convergence occurs in time that is polynomial in n, the number of objects. Hence 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. As Figure 1 illustrates, 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 3 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

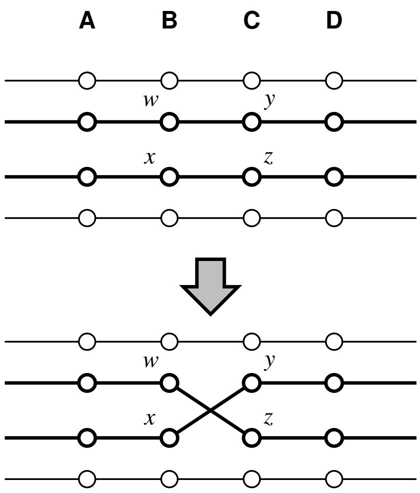

Figure 3: Figure demonstrating a single transition from one assignment with four trajectories to another.

observation pairs to be swapped are chosen uniformly at random as we cycle through all pairs of adjacent sensors. Thus, the $q(\omega|\omega')$ values are trivial to compute, and the chain is provably ergodic.

We can derive a general expression for $\pi(\omega)$, 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 3. Generalizing Eq. (5) for the observation sequence $\mathbf{O}^1, \mathbf{O}^2 \ldots \mathbf{O}^k$, we express $P(\omega_{1\ldots k}|\mathbf{O})$ as

$$P(\omega_{1\ldots k}|\mathbf{O}) = \alpha \sum_{h^1} P(h^1)P(\mathbf{O}^1|h^1)$$
$$\prod_{j=1}^{k} \sum_{h^{j+1}} P(h^{j+1}|h^j, \omega_{j,j+1})P(\mathbf{O}^{j+1}|h^{j+1}) \qquad (5)$$

As has noted in Section 2.2, the hidden variables can be divided into dynamic variables, which change over time, and intrinsic variables, which do not. Here, we assume that all the dynamics can be observed reliably.[4] Therefore, a set of observations such as O^j (all the observations made at sensor j) can be factored into two subsets: d^j, the (reliable) observations of the dynamic variables,

[4] This is not actually necessary, as the algorithm can handle dynamic variables if the graphical model is appropriately extended.

and m^j, the (unreliable) measures taken of the intrinsics. Similarly, the hidden variables can be divided into d^j, the actual values of the dynamic variables, and i^j, the actual values of the intrinsics. Thus, $P(h^{j+1}|h^j, \omega_{j,j+1})$ becomes $P(i^{j+i}|i^j, \omega_{j,j+1})P(d^{j+i}|d^j, \omega_{j,j+1})$, and $P(O^j|h^j)$ becomes $P(m^j|i^j)P(d^j|d^j)$. Now, we can note that the terms $P(d^j|d^j)$ and $P(i^{j+i}|i^j, \omega_{j,j+1})$ can be discarded, as $d^j = d^j$ and $i^{j+i} = i^j$. So, given our assumptions, Eq. (5) simplifies as follows

$$P(\omega_{1\ldots k}|\mathbf{O}) = \alpha \sum_{i^1} P(i^1)P(d^1)P(m^1|i^1)$$

$$\prod_{j=1}^{k} \sum_{i^{j+1}} P(d^{j+1}|d^j, \omega_{j,j+1})P(m^{j+1}|i^{j+1}) \qquad (6)$$

Some terms are now independent of the intrinsics we are summing over, and can be moved outside the summations.

We can now evaluate the ratio $P(\omega'|\mathbf{O})/P(\omega|\mathbf{O})$ in a simplified form. As we are assuming that all trajectories are independent of each other, each of the terms in Eq. (6) may be factored into a product of probabilities for individual objects. For example, $P(d^{j+1}|d^j, \omega j, j+1) = \prod_{(a,b)\in\omega_{j,j+1}} P(d_a^{j+1}|d_b^j, a=b))$, and $P(m^j|i^j) = \prod_{a\in m^j} P(m_a^j|i_a^j)$. Therefore, in the ratio, the probabilities for trajectories not involved in $wxyz$ will cancel, as will the alphas. Once this is done, it will be convenient to rewrite the summations so that we are summing over all the possible intrinsic values for each trajectory, rather than all the intrinsics at a sensor. Let us define one such summation $S(xy)$, over the trajectory which includes xy:

$$S(xy) = \sum_{i_{xy}} P(i_{xy}) \prod_{j=1}^{k} P(m_{xy}^j|i_{xy})$$

The ratio can then be expressed simply as

$$\frac{P(z|w, z=w)P(y|x, y=x)S(wz)S(xy)}{P(y|w, y=w)P(z|x, z=x)S(xz)S(wy)} \qquad (7)$$

Notice that the only necessary computations will be evaluations of the local dynamic models and summations or integrations over the intrinsics along the relevant trajectories. All of our intrinsic variable models are Gaussians, so the integral over their product can be evaluated as an integral over a single Gaussian. This will take time proportional to the longest network trajectory, which is constant for any given network. Before the ratio was simplified, the calculation would have taken time proportional to the number of objects in the system.

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 approximate E step yields samples of ω: they are generated quite naturally as the MCMC is run. For each sample, we then obtain values for the intrinsics by evaluating $E(I|\omega, O)$, which (using our current models) is an integral over a product of Gaussians and can be computed in time proportional to trajectory length.[5] The M step then uses the ω and $E(I)$ values of each sample as if they were observed variables, to perform conventional, exact parameter learning.

Running EM to its convergence requires a Markov chain run for each EM iteration. In practice, the chain converges very quickly, as the models change slowly. 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 models in the limit, and our preliminary experimental results are encouraging.

4 Experimental results

We have performed an experiment comparing our approach to that of Huang and Russell. The data sets used were created using a freeway simulator which allowed us to control road configurations, camera characteristics, and vehicle behaviour. Each set included the observations generated by approximately a hundred cars. Both of our implementations ran quickly. In the case of the MCMC this was due to its fast convergence rate: even though every new vehicle observation would create a new chain by expanding the state space, we were able to make use of the old assignment state when setting up a starting point, so that the process would always reach convergence in under fifty attempted transitions.

4.1 Overcoming faulty sensors

In this experiment, the algorithms were applied to a network such as that in Figure 1, set up in a manner analogous to the licence-plate example of Section 2.1. The outer two cameras were completely reliable to the point where they could distinguish all cars, while the central camera was subjected to various degrees of distortion (Its ability to see color was reduced.) Thus, the data set contained enough data for a complete matching between cameras, but a pairwise appearance-based model could be expected to find extracting it difficult.

The results of the experiment, averaged over three runs, can be seen in Figure 4. 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 the MCMC is clear: the rapid drop in the accuracy of the

[5] Other types of statistical models, where this calculation may be unfeasible, may require a different approach, such as sampling over I as well as ω.

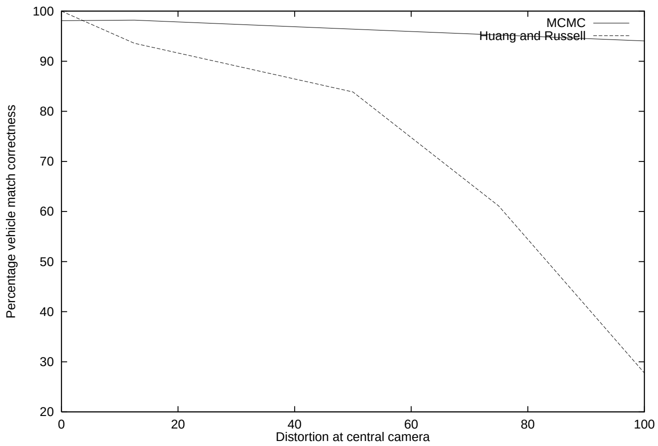

Figure 4: The vehicle matching accuracy of two algorithms as a function of the accuracy of a sensor.

pairwise algorith demonstrates the worth of estimating intrinsics.

5 Summary

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

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.

[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.