Markov Chain Monte Carlo Data Association for Multiple-Target Tracking

Songhwai Oh, Stuart Russell, and Shankar Sastry

Abstract

This paper presents Markov chain Monte Carlo data association (MCMCDA) for solving data association problems arising in multiple-target tracking in a cluttered environment. When the number of targets is fixed, the single-scan version of MCMCDA approximates joint probabilistic data association (JPDA). Although the exact computation of association probabilities in JPDA is NP-hard, we prove that the single-scan MCMCDA algorithm provides a fully polynomial randomized approximation scheme for JPDA. For general multiple-target tracking problems, in which unknown numbers of targets appear and disappear at random times, we present a multi-scan MCMCDA algorithm that approximates the optimal Bayesian filter. It exhibits remarkable performance compared to multiple hypothesis tracking (MHT) under extreme conditions, such as a large number of targets in a dense environment, low detection probabilities, and high false alarm rates.

I. Introduction

Multiple-target tracking plays an important role in many areas of engineering such as surveillance, computer vision, and signal processing [1]–[3]. Under the general setup, some indistinguishable targets move continuously in a given region, typically independently according to a known, Markovian process. Targets arise at random in space and time, persist for a random length of time, and then cease to exist; the sequence of states that a target follows during its lifetime is called a *track*. The positions of moving targets are measured, either at random intervals or, more typically, in periodic *scans* that measure the positions of all targets simultaneously. The

The authors are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720. {sho,russell,sastry}@eecs.berkeley.edu.

This material is based upon work supported by the National Science Foundation under Grant No. EIA-0122599.

position measurements are noisy and occur with detection probability less than one, and there is a noise background of spurious position reports, *i.e.*, false alarms.

The essence of the multiple-target tracking problem is to find tracks from the noisy measurements. Now, if the sequence of measurements associated with each target is known, multiple-target tracking (at least under the assumption of independent motion) reduces to a set of state estimation problems, which, for the purposes of this paper, we assume to be straightforward. Unfortunately, the association between measurements and targets is unknown. The *data association* problem is to work out which measurements were generated by which targets; more precisely, we require a partition of measurements such that each element of a partition is a collection of measurements generated by a single target or clutter [4]. In the general case, uncertainty as to the correct association is unavoidable.

Multiple-target tracking algorithms are often categorized according to the objective function that they purport to optimize:

- *Heuristic* approaches typically involve no explicit objective function. For example, the greedy nearest-neighbor filter (NNF) [1] processes the new measurements in some order and associates each with the target whose predicted position is closest, thereby selecting a single association after each scan. Although effective under benign conditions, the NNF gives order-dependent results and breaks down under more difficult circumstances.
- *Maximum a posteriori* (MAP) approaches find the most probable association, given the measurements made so far, and estimate tracks given this association.
- The *Bayesian* approach generates optimal filtering predictions by summing over all possible associations, weighted by their probabilities. Under certain distributional assumptions (*e.g.*, linear–Gaussian models), the optimal Bayesian filter can be shown to minimize the mean squared error in the track estimates. For this reason, approaches that sum over multiple associations are sometimes called *minimum mean square error* (MMSE) approaches.

Tracking algorithms can also be categorized by the way in which they process measurements:

- *Single-scan* algorithms estimate the current states of targets based on their previously computed tracks and the current scan of measurements.
- *Multi-scan* algorithms may revisit past scans when processing each new scan, and can thereby revise previous association decisions in the light of new evidence.

MAP approaches include the well-known *multiple hypothesis tracking* (MHT) algorithm [5]. MHT is a multi-scan tracking algorithm that maintains multiple hypotheses associating past measurements with targets. When a new set of measurements arrives, a new set of hypotheses is formed from each previous hypothesis. The algorithm returns a hypothesis with the highest posterior as a solution. MHT is categorized as a “deferred logic” method [6] in which the decision about forming a new track or removing an existing track is delayed until enough measurements are collected. MHT is capable of initiating and terminating a varying number of tracks and is suitable for autonomous surveillance applications. The main disadvantage of MHT in its pure form is its computational complexity since the number of hypotheses grows exponentially over time. Various heuristic methods have been developed to control this growth [5], [7], [8]; but these methods are applied at the expense of sacrificing the MAP property. Other MAP approaches have been tried besides MHT, including 0-1 integer programming [9] and multidimensional assignment [6]. As the latter reference shows, the underlying MAP data association problem is NP-hard, so we do not expect to find efficient, exact algorithms.

Exact Bayesian data association is even less tractable than the MAP computation. Several “pseudo-Bayesian” methods have been proposed, of which the best-known is the *joint probabilistic data association* (JPDA) filter [1]. JPDA is a suboptimal single-scan approximation to the optimal Bayesian filter; it can also be viewed as an assumed-density filter in which the joint state estimate is always a single set of tracks for a “known” set of targets. At each time step, instead of finding a single best association between measurements and tracks, JPDA enumerates all possible associations and computes association probabilities $\{\beta_{jk}\}$, where β_{jk} is the probability that j-th measurement extends the k-th track. Given an association, the state of a target is estimated by a filtering algorithm and this conditional state estimate is weighted by the association probability. Then the state of a target is estimated by summing over the weighted conditional estimates. JPDA has proved very effective in cluttered environments compared with NNF [1]. The exact calculation of association probabilities $\{\beta_{jk}\}$ in JPDA, which requires the summation over all association event probabilities, is NP-hard [10] since the related problem of finding the permanent of a matrix is #P-complete [11]. Some heuristic approaches to approximate JPDA include a “cheap” JPDA algorithm [12], “suboptimal” JPDA [13] and “near-optimal” JPDA [14]. In [15], a single-scan data association problem is considered and a leave-one-out heuristic is developed to avoid the enumeration of all possible associations.

The main contribution of this paper is the development of a real-time multiple-target tracking method called Markov chain Monte Carlo data association (MCMCDA). Unlike MHT and JPDA, MCMCDA is a true approximation scheme for the optimal Bayesian filter; *i.e.*, when run with unlimited resources, it converges to the Bayesian solution. As the name suggests, MCMCDA uses Markov chain Monte Carlo (MCMC) sampling instead of summing over all possible associations. MCMC was first used to solve data association problems by Pasula *et al*. [16], [17], who showed it to be effective for multi-camera traffic surveillance problems involving hundreds of vehicles. More recently, in [18], MCMC was used to approximate the association probabilities in JPDA and was shown to outperform Fitzgerald's cheap JPDA.[1] MCMCDA goes beyond these contributions by incorporating missing measurements, false alarms and an ability to initiate and terminate tracks, so that the algorithm can be applied to the full range of data association problems.

The paper has two main technical results. The first is a theorem showing that, when the number of targets is fixed, single-scan MCMCDA is a fully polynomial randomized approximation scheme for JPDA. More specifically, for any $\epsilon > 0$ and any $0 < \eta < .5$, the algorithm finds "good estimates" with probability at least $1-\eta$ in time complexity $O(\epsilon^{-2} \log \eta^{-1} N(N \log N + \log(\epsilon^{-1})))$, where N is the number of measurements per scan. (The precise meaning of good estimates is defined in Section IV-C.) The theorem is based on the seminal work of Jerrum and Sinclair [20], who designed an MCMC algorithm for approximating the permanent of a matrix and developed new techniques for analyzing its rate of convergence. As mentioned above, the relationship between JPDA and computing the permanent was identified by Collins and Uhlmann [10]; the connection to the polynomial-time approximation theorems of Jerrum and Sinclair was first suggested by Pasula *et al.* [16]. Although our proof has the same structure as that of Jerrum and Sinclair, substantial technical work was required to complete the mapping from computing the permanent to solving JPDA, including the usage of gating conditions that ensure appropriate lower bounds on individual association probabilities.

Our second technical result is the complete specification of the transition structure for a multi-scan version of MCMCDA that includes detection failure, false alarms, and track initiation and termination. We prove that the resulting algorithm converges to the full Bayesian solution.

[1] MCMC has also been used for problems that are roughly isomorphic to the data association problem, including state estimation in the switching Kalman filter [19] and stereo correspondence in computer vision [3].

We also provide the first extensive experimental investigation of MCMCDA's performance on classical data association problems. We demonstrate remarkably effective real-time performance compared to MHT under extreme conditions, such as a large number of targets in a dense environment, low detection probabilities, and high false alarm rates. We also show the application of MCMCDA to track people in video sequences.

The remainder of this paper is structured as follows. The multiple-target tracking problem and its probabilistic model are described in Section II. In Section III, the Markov chain Monte Carlo (MCMC) method is summarized. The single-scan MCMCDA algorithm is presented in Section IV along with the proof that it approximates JPDA in polynomial time. The multi-scan MCMCDA algorithm is described in Section V along with our experimental results.

II. Multiple-Target Tracking

A. Problem Formulation

Let $T \in \mathbb{Z}^+$ be the duration of surveillance. Let K be the number of objects that appear in the surveillance region $\mathcal{R}$ during the surveillance period. Each object k moves in $\mathcal{R}$ for some duration $[t_\mathrm{i}^k, t_\mathrm{f}^k] \subset [1,T]$. Notice that the exact values of K and $\{t_\mathrm{i}^k, t_\mathrm{f}^k\}$ are unknown. Each object arises at a random position in $\mathcal{R}$ at t_i^k, moves independently around $\mathcal{R}$ until t_f^k and disappears. At each time, an existing target persists with probability $1-p_\mathrm{z}$ and disppears with probability p_z. The number of objects arising at each time over $\mathcal{R}$ has a Poisson distribution with a parameter $\lambda_\mathrm{b} V$ where λ_b is the birth rate of new objects per unit time, per unit volume, and V is the volume of $\mathcal{R}$. The initial position of a new object is uniformly distributed over $\mathcal{R}$.

Let $F^k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ be the discrete-time dynamics of the object k, where n_x is the dimension of the state variable, and let $x_t^k \in \mathbb{R}^{n_x}$ be the state of the object k at time t. The object k moves according to

$$x_{t+1}^k = F^k(x_t^k) + w_t^k, \qquad \text{for } t = t_\mathrm{i}^k, \ldots, t_\mathrm{f}^k - 1, \tag{1}$$

where $w_t^k \in \mathbb{R}^{n_x}$ are white noise processes. The white noise process is included to model non-rectilinear motions of targets. The noisy observation (or measurement[2]) of the state of the object is measured with a detection probability p_d. Notice that, with probability $1-p_\mathrm{d}$, the object is not detected and we call this a missing observation. There are also false alarms and the number of

[2] Note that the terms *observation* and *measurement* are used interchangeably in this paper.

false alarms has a Poisson distribution with a parameter $\lambda_{\mathrm{f}}V$ where λ_{f} is the false alarm rate per unit time, per unit volume. Let n_t be the number of observations at time t, including both noisy observations and false alarms. Let $y_t^j \in \mathbb{R}^{n_y}$ be the j-th observation at time t for $j = 1, \ldots, n_t$, where n_y is the dimension of each observation vector. Each object generates a unique observation at each sampling time if it is detected. Let $H^j : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ be the observation model. Then the observations are generated as follows:

$$y_t^j = \begin{cases} H^j(x_t^k) + v_t^j & \text{if } j\text{-th observation is from } x_t^k \\ u_t & \text{otherwise,} \end{cases} \tag{2}$$

where $v_t^j \in \mathbb{R}^{n_y}$ are white noise processes and $u_t \sim \mathrm{Unif}(\mathcal{R})$ is a random process for false alarms. We assume that targets are indistinguishable in this paper, but if observations include target type or attribute information, the state variable can be extended to include target type information. The multiple-target tracking problem is to estimate K, $\{t_{\mathrm{i}}^k, t_{\mathrm{f}}^k\}$ and $\{x_t^k : t_{\mathrm{i}}^k \leq t \leq t_{\mathrm{f}}^k\}$, for $k = 1, \ldots, K$, from observations.

B. Solutions to the Multiple-Target Tracking Problem

Let $Y_t = \{y_t^j : j = 1, \ldots, n_t\}$ be all measurements at time t and $Y = \{Y_t : 1 \leq t \leq T\}$ be all measurements from $t = 1$ to $t = T$. Let Ω be a collection of partitions of Y such that, for $\omega \in \Omega$,

1) $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$;
2) $\bigcup_{k=0}^{K} \tau_k = Y$ and $\tau_i \cap \tau_j = \emptyset$ for $i \neq j$;
3) τ_0 is a set of false alarms;
4) $|\tau_k \cap Y_t| \leq 1$ for $k = 1, \ldots, K$ and $t = 1, \ldots, T$; and
5) $|\tau_k| \geq 2$ for $k = 1, \ldots, K$.

An example of a partition is shown in Figure 1 and ω is also known as a *joint association event* in literature. Here, K is the number of tracks for the given partition $\omega \in \Omega$ and $|\tau_k|$ denotes the cardinality of the set τ_k. We call τ_k a track when there is no confusion although the actual track is the set of estimated states from the observations τ_k. However, we assume there is a deterministic function that returns a set of estimated states given a set of observations, so no distinction is required. The fourth requirement says that a track can have at most one observation at each time, but, in the case of multiple sensors with overlapping sensing regions, we can easily

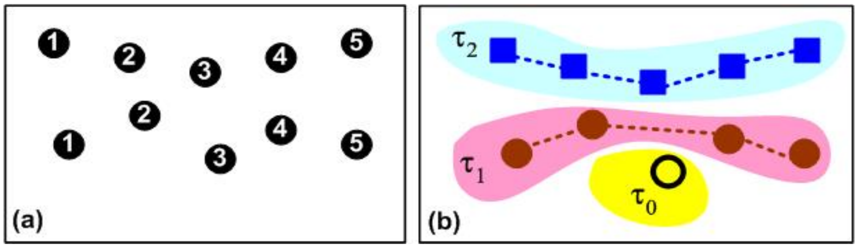


Fig. 1. (a) An example of observations Y (each circle represents an observation and numbers represent observation times). (b) An example of a partition ω of Y

relax this requirement to allow multiple observations per track. A track is assumed to contain at least two observations since we cannot distinguish a track with a single observation from a false alarm, assuming $\lambda_{\mathrm{f}} > 0$. For special cases, in which $p_{\mathrm{d}} = 1$ or $\lambda_{\mathrm{f}} = 0$, the definition of Ω can be adjusted accordingly.

Let e_{t-1} be the number of targets from time $t-1$, z_t be the number of targets terminated at time t and $c_t = e_{t-1} - z_t$ be the number of targets from time $t-1$ that have not terminated at time t. Let a_t be the number of new targets at time t, d_t be the number of actual target detections at time t and $g_t = c_t + a_t - d_t$ be the number of undetected targets. Finally, let $f_t = n_t - d_t$ be the number of false alarms. It can be shown that the posterior of ω is:

$$P(\omega|Y) \propto P(Y|\omega) \prod_{t=1}^{T} p_{\mathrm{z}}^{z_t}(1-p_{\mathrm{z}})^{c_t} p_{\mathrm{d}}^{d_t}(1-p_{\mathrm{d}})^{g_t} \lambda_{\mathrm{b}}^{a_t} \lambda_{\mathrm{f}}^{f_t}, \tag{3}$$

where $P(Y|\omega)$ is the likelihood of observations Y given ω, which can be computed based on the chosen dynamic and measurement models.

As described in the introduction, two approaches to solve the multiple-target tracking problem are *maximum a posteriori* (MAP) and Bayesian (or *minimum mean square error* (MMSE)) approaches. The MAP approach finds a partition of observations such that $P(\omega|Y)$ is maximized and estimates states of targets based on the partition which maximizes $P(\omega|Y)$. The MMSE approach seeks the conditional expectations such as $\mathbb{E}(x_t^k|Y)$ to minimize the expected (square) error. However, when the number of targets is not fixed, a unique labeling of each target is required to find $\mathbb{E}(x_t^k|Y)$ under the MMSE approach.

III. Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) plays a significant role in many fields such as physics, statistics, economics, and engineering [21]. In some cases, MCMC is the only known general algorithm that finds a good approximate solution to a complex problem in polynomial time [20]. MCMC techniques have been applied to complex probability distribution integration problems, counting problems, and combinatorial optimization problems [20], [21].

MCMC is a general method to generate samples from a distribution π on a space Ω by constructing a Markov chain $\mathcal{M}$ with states $\omega \in \Omega$ and stationary distribution $\pi(\omega)$. We now describe an MCMC algorithm known as the Metropolis-Hastings algorithm. If we are at state $\omega \in \Omega$, we propose $\omega' \in \Omega$ following the proposal distribution $q(\omega, \omega')$. The move is accepted with an acceptance probability $A(\omega, \omega')$ where

$$A(\omega, \omega') = \min\left(1, \frac{\pi(\omega')q(\omega', \omega)}{\pi(\omega)q(\omega, \omega')}\right), \tag{4}$$

otherwise the sampler stays at ω. With this construction, the detailed balance condition is satisfied, *i.e.*, for all $\omega, \omega' \in \Omega$ with $\omega' \neq \omega$,

$$Q(\omega, \omega') = \pi(\omega)P(\omega, \omega') = \pi(\omega')P(\omega, \omega'), \tag{5}$$

where $P(\omega, \omega') = q(\omega, \omega')A(\omega, \omega')$ is the transition probability from ω to ω'.

If $\mathcal{M}$ is irreducible and aperiodic, then $\mathcal{M}$ converges to its stationary distribution by the ergodic theorem [22]. Hence, for a given bounded function f, the sample mean $\hat{f} = \frac{1}{T}\sum_{t=1}^{T} f(\omega_t)$, where ω_t is the state of $\mathcal{M}$ at time t, converges to $\mathbb{E}_\pi f(\omega)$ as $T \to \infty$. Notice that (4) requires only the ability to compute the ratio $\pi(\omega')/\pi(\omega)$, avoiding the need to normalize π.

An ergodic chain $\mathcal{M}$ on state space Ω converges to its stationary distribution asymptotically. But a practical question is how fast $\mathcal{M}$ approaches stationarity. One way to measure the rate of convergence of $\mathcal{M}$ to stationarity is the "mixing time" of the Markov chain. Let P be the transition probabilities of $\mathcal{M}$ and let $P_\omega^t(\cdot)$ be the distribution of the state at time t given that $\mathcal{M}$ is started from the initial state $\omega \in \Omega$. If π is the stationary distribution of $\mathcal{M}$, then the *total variation distance* at time t with initial state ω is defined as

$$\Delta_\omega(t) = \|P_\omega^t - \pi\| = \max_{S \subset \Omega} |P_\omega^t(S) - \pi(S)| = \frac{1}{2}\sum_{y \in \Omega} |P_\omega^t(y) - \pi(y)|. \tag{6}$$

The rate of convergence of $\mathcal{M}$ to stationarity can be measured by the *mixing time*:

$$\tau_\omega(\epsilon) = \min\{t : \Delta_\omega(s) \leq \epsilon \text{ for all } s \geq t\}. \tag{7}$$

One approach to bound $\tau_\omega(\epsilon)$ of a Markov chain with a complex structure is the canonical path method [20]. In this paper, we consider a highly complex Markov chain, hence we use the canonical path method to bound $\tau_\omega(\epsilon)$ of the Markov chain simulated by the MCMCDA algorithm given in Section IV. For the remainder of this section, we describe the canonical path method.

For a finite, reversible and ergodic Markov chain $\mathcal{M}$ with state space Ω, consider an undirected graph $G = (V, E)$ where $V = \Omega$ and $E = \{(x, y) : Q(x, y) > 0\}$ (recall the definition of $Q(\cdot, \cdot)$ from (5)). So an edge $(x, y) \in E$ indicates that the Markov chain $\mathcal{M}$ can make a transition from x to y or from y to x in a single step. For each ordered pair $(x, y) \in \Omega^2$, the canonical path γ_{xy} is a simple path from x to y in G. In terms of $\mathcal{M}$, the canonical path γ_{xy} is a sequence of legal transitions from x to y in $\mathcal{M}$. Let $\Gamma = \{\gamma_{xy} : x, y \in \Omega\}$ be the set of all canonical paths. Now the mixing time of the chain is related to the *maximum edge loading*:

$$\bar{\rho} = \bar{\rho}(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)|\gamma_{xy}|. \tag{8}$$

If $\bar{\rho}$ is not so big, *i.e.*, no single edge is overloaded, then the Markov chain can move around fast and achieve the rapidly mixing property. The main result for the canonical path method is as follows [20], [23]:

Theorem 1: Let $\mathcal{M}$ be a finite, reversible, ergodic Markov chain with loop probabilities $P(x, x) \geq \frac{1}{2}$ for all states x. Let Γ be a set of canonical paths with maximum edge loading $\bar{\rho}$. Then the mixing time of $\mathcal{M}$ satisfies $\tau_x(\epsilon) \leq \bar{\rho}(\log \pi(x)^{-1} + \log \epsilon^{-1})$, for any choice of initial state x.

IV. Single-Scan MCMCDA

In this section, we consider a special case of the multiple-target tracking problem described in Section II, in which the number of targets is fixed and known, and propose the single-scan MCMCDA. Then, we prove that the single-scan MCMCDA algorithm finds an approximate solution to JPDA in polynomial time.

A. Single-Scan MCMCDA Filter

The single-scan MCMCDA filter is based on the same filtering method used in the JPDA filter [1]. JPDA has been traditionally used with the Kalman filter, assuming linear-Gaussian models, *i.e.*, linear dynamic and measurement models and white Gaussian noise processes [1]. However, JPDA has also been applied with a nonlinear filtering algorithm such as a particle filter [24]. We present the single-scan MCMCDA filter under the general dynamics and measurement models defined in Section II. The description of the single-scan MCMCDA filter for linear-Gaussian models is given in [25].

Suppose that we have the distribution $\hat{P}(X_{t-1}^k|y_{1:t-1})$ from the previous filtering time $t-1$, for each target k, where $\hat{P}(X|y)$ is an approximation of the distribution $P(X|y)$ and $y_{1:t} = \{y_1, \ldots, y_t\}$. Notice that we can only approximate $P(X_{t-1}^k|y_{1:t-1})$ since we process each measurement scan sequentially; this is why the JPDA filter is a suboptimal approximation to the optimal Bayesian filter. We follow the notations defined in Section II, except that random variables are denoted by capital letters. Notice that, for linear-Gaussian models such as those used in [1], [25], $\hat{P}(X_{t-1}^k|y_{1:t-1})$ is a Gaussian distribution and completely described by its mean and variance.

Step 1 (Prediction): For each k, compute the distribution

$$\begin{aligned} \hat{P}(X_t^k|y_{1:t-1}) &:= \int P(X_t^k|x_{t-1}^k, y_{1:t-1})\hat{P}(x_{t-1}^k|y_{1:t-1})dx_{t-1}^k \\ &= \int P(X_t^k|x_{t-1}^k)\hat{P}(x_{t-1}^k|y_{1:t-1})dx_{t-1}^k, \end{aligned} \tag{9}$$

where the Markovian assumption is used in the second equality and $P(X_t^k|x_{t-1}^k)$ is determined by the noise process w_t^k in (1).

Step 2 (Measurement Validation): For each k and j, compute the distribution

$$\begin{aligned} \hat{P}^k(Y_t^j|y_{1:t-1}) &:= \int P(Y_t^j|x_t^k, y_{1:t-1})\hat{P}(x_t^k|y_{1:t-1})dx_t^k \\ &= \int P(Y_t^j|x_t^k)\hat{P}(x_t^k|y_{1:t-1})dx_t^k, \end{aligned} \tag{10}$$

where the second equality uses the fact the current observation is independent of previous observations given the current state and $P(Y_t^j|x_t^k)$ is determined by the noise process v_t^j in (2). Notice that $\hat{P}^k(y_t^j|y_{1:t-1})$ is the probability density of having observation y_t^j given $y_{1:t-1}$, when y_t^j is an observation of target k. Again, for linear-Gaussian models, $\hat{P}^k(Y_t^j|y_{1:t-1})$ is a Gaussian distribution and completely determined by its mean and variance. As in JPDA, we validate measurements and use only validated measurements when estimating states of targets. The measurement y_t^j is validated for target k, if and only if

$$\hat{P}^k(y_t^j|y_{1:t-1}) \geq \delta^k, \tag{11}$$

where δ^k are appropriate thresholds. We assume that all measurements are validated with at least one target. If not, we can always consider the reduced problem, which consists of only validated measurements and targets with at least one validated measurement, and separately estimate states of targets with no validated measurement.

Step 3 (State Estimation): Let Ω be a set of all feasible joint association events at time t. For notational convenience, the subscript t is dropped when there is no confusion. For each $\omega \in \Omega$, $\omega = \{(j,k)\}$, where (j,k) denotes an event that observation j is associated with target k. A joint association event ω is feasible when (i) for each $(j,k) \in \omega$, y_t^j is validated for target k; (ii) an observation is associated with at most one target; and (iii) a target is associated with at most one observation.

Let N be the number of validated observations. We encode the feasible joint association events in a bipartite graph. Let $G = (U, V, E)$ be a bipartite graph, where $U = \{y_t^j : 1 \leq j \leq N\}$ is a vertex set of validated observations, $V = \{k : 1 \leq k \leq K\}$ is a vertex set of target indices, and $E = \{(u,v) : u \in U, v \in V, \hat{P}^v(u|y_{1:t-1}) \geq \delta^v\}$. An edge $(u,v) \in E$ indicates that observation u is validated for target v according to (11). Then a feasible joint association event is a *matching* in G, *i.e.*, a subset $M \subset E$ such that no two edges in M share a vertex. The set of all feasible joint association events Ω can be represented as $\Omega \subset M_0(G) \cup \cdots \cup M_K(G)$, where $M_k(G)$ is a set of k-matchings in G.

Finally, using the total probability theorem, compute the distribution

$$\hat{P}(X_t^k|y_{1:t}) := \sum_{\omega \in \Omega} \hat{P}(X_t^k|\omega, y_{1:t})\hat{P}(\omega|y_{1:t}) = \sum_{j=0}^{n_t} \beta_{jk}\hat{P}(X_t^k|\omega_{jk}, y_{1:t}), \tag{12}$$

where ω_{jk} denotes the event $\{\omega \ni (j,k)\}$, ω_{0k} denotes the event that no observation is associated with target k, and β_{jk} is an association probability, such that,

$$\beta_{jk} = \hat{P}(\omega_{jk}|y_{1:t}) = \sum_{\omega:(j,k)\in\omega} \hat{P}(\omega|y_{1:t}). \tag{13}$$

$\hat{P}(X_t^k|\omega_{jk}, y_{1:t})$ in (12) can be easily computed by considering it as a single target estimation problem with a single observation. Hence, the computation of $\hat{P}(X_t^k|y_{1:t})$ reduces to the computation of β_{jk}. The computation of β_{jk} requires a summation over the posteriors, hence the enumeration of all joint association events. In JPDA, $\mathbb{E}(X_t^k|y_{1:t})$ is estimated in the same manner as (12) and JPDA is a method for estimating expectations such as $\mathbb{E}(X_t^k|y_{1:t})$ using the association probabilities $\{\beta_{jk}\}$ in the presence of the identity uncertainty. As mentioned earlier, the exact calculation of $\{\beta_{jk}\}$ in JPDA is NP-hard [10] and it is the major drawback of JPDA.

In the next sections, we describe an algorithm which approximates the association probabilities $\{\beta_{jk}\}$ and prove that the running time of the algorithm is polynomial in the size of the problem.

B. Single-Scan MCMCDA Algorithm

The single-scan MCMCDA algorithm is used to approximate $\{\beta_{jk}\}$ in the single-scan MCMCDA filter described in the previous section. Based on the parametric false alarm model described in Section II-A, the posterior of $\omega \in \Omega$ can be written as

$$\begin{aligned} P(\omega|y_{1:t}) &= \frac{1}{Z_0} P(\omega|y_{1:t-1}) P(y_t|\omega, y_{1:t-1}) = \frac{1}{Z} P(\omega) P(y_t|\omega, y_{1:t-1}) \\ &\approx \frac{1}{Z} P(\omega) \hat{P}(y_t|\omega, y_{1:t-1}) \\ &= \frac{1}{Z} \lambda_{\mathrm{f}}^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1-p_{\mathrm{d}})^{K-|\omega|} \prod_{(u,v)\in\omega} \hat{P}^{v}(u|y_{1:t-1}) =: \hat{P}(\omega|y_{1:t}), \end{aligned} \tag{14}$$

where Z_0 and Z are normalizing constants, the Bayes rule is used in the first equality, and the second equality uses that fact that ω is independent of $y_{1:t-1}$.

The MCMC data association (MCMCDA) algorithm is an MCMC algorithm whose state space is the set of all feasible joint association events Ω and whose stationary distribution is the posterior $\hat{P}(\omega|y_{1:t})$ (14). Each step of the MCMCDA algorithm is described in Algorithm 1 along with three MCMC moves (addition, deletion and switch moves), where we use the sampling method from [20]. In Algorithm 1, since we have a uniform proposal distribution, $A(\omega, \omega') = \min\left(1, \frac{\pi(\omega')}{\pi(\omega)}\right)$, where $\pi(\omega) = \hat{P}(\omega|y_{1:t})$ from (14).

C. Analysis

Let $\mathcal{M}$ be the Markov chain simulated by Algorithm 1. Since the self-loop probability is nonzero, $\mathcal{M}$ is aperiodic. It can be easily seen that $\mathcal{M}$ is irreducible, *i.e.*, all states communicate, for example via the empty matching. In addition, the transitions described in Algorithm 1 satisfy the detailed balance condition (5) so $\mathcal{M}$ is reversible. Hence, by the ergodic theorem, the chain converges to its stationary distribution [22].

Let us first take a look at the complexity of the problem. As noted earlier, the state space of the Markov chain $\mathcal{M}$ is $\Omega \subset M_0(G) \cup \cdots \cup M_K(G)$. For each k, $|M_k(G)| \leq \binom{K}{k} \frac{N!}{(N-k)!}$ with equality if the subgraph of G with the k chosen vertices in V is a complete bipartite graph, *i.e.*,

Algorithm 1 MCMCDA (single step)

```
sample $U$ from Unif$[0,1]$
if $U < \frac{1}{2}$ then
    $\omega' = \omega$
else
    choose $e = (u,v) \in E$ uniformly at random
    if $e \in \omega$ then
        $\omega' = \omega - e$        (deletion move)
    else if both $u$ and $v$ are unmatched in $\omega$ then
        $\omega' = \omega + e$        (addition move)
    else if exactly one of $u$ and $v$ is matched in $\omega$ and $e'$ is the matching edge then
        $\omega' = \omega + e - e'$   (switch move)
    else
        $\omega' = \omega$
    end if
end if
$\omega = \omega'$ with probability $A(\omega, \omega')$
```

all observations are validated for all k chosen targets. Hence, we can bound the size of Ω as

$$|\Omega| \le |M_0(G)| + \cdots + |M_K(G)| \le \sum_{k=0}^{K} \binom{K}{k} \frac{N!}{(N-k)!}. \tag{15}$$

Certainly, the size of the state space grows exponentially as the number of targets or the number of observations increases, hence the exact calculation of JPDA by enumeration is not feasible when the number of targets or the number of observations is large.

In (14), the normalizing constant becomes

$$Z = \sum_{\omega \in \Omega} \left(\lambda_{\mathrm{f}}^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1-p_{\mathrm{d}})^{K-|\omega|} \prod_{(u,v)\in\omega} \hat{P}^v(u|y_{1:t-1}) \right). \tag{16}$$

We assume that each likelihood term can be bounded as $\underline{L} \le \hat{P}^v(u|y_{1:t-1}) \le \bar{L}$, for all $(u,v) \in E$, where $\underline{L} = \min \delta^k$ and $\bar{L}$ can be precomputed based on $\hat{P}^v(u|y_{1:t-1})$. Here, we are making a reasonable assumption that $\hat{P}^v(u|y_{1:t-1}) \leq \bar{L} < \infty$ for all $(u,v) \in E$, *e.g.* linear-Gaussian models [25]. Notice that the lower bound $\underline{L}$ is due to the measurement validation. In JPDA, the measurement validation is used to reduce the number of feasible joint association events. However, we later find that it is required to approximate the association probabilities in polynomial time.

For Theorem 2 below, let $C = \frac{p_{\mathrm{d}}\bar{L}}{\lambda_{\mathrm{f}}(1-p_{\mathrm{d}})}$, $D = \frac{\lambda_{\mathrm{f}}(1-p_{\mathrm{d}})}{\underline{L}p_{\mathrm{d}}}$ and $R = \max\{1, C, D\}$. Also define $m_1 = \max\{1, \bar{L}\}$, $m_2 = \min\{1, \underline{L}\}$,

$$\begin{aligned} m_3(K,N) &= \max_{0\leq k\leq K}\{\lambda_{\mathrm{f}}^{N-k}p_{\mathrm{d}}^k(1-p_{\mathrm{d}})^{K-k}\}, \\ m_4(K,N) &= \min_{0\leq k\leq K}\{\lambda_{\mathrm{f}}^{N-k}p_{\mathrm{d}}^k(1-p_{\mathrm{d}})^{K-k}\}, \quad \text{and} \\ m_5(K,N) &= K\log\frac{m_1}{m_2} + \log\frac{m_3(K,N)}{m_4(K,N)} + \sum_{k=1}^{K+1}\log k + \sum_{n=1}^{N}\log n. \end{aligned}$$

Remark 1: If $.5 < p_{\mathrm{d}} < 1$ and $\lambda_{\mathrm{f}} < 1 - p_{\mathrm{d}}$, then $m_3(K,N) = \lambda_{\mathrm{f}}^{N-K}p_{\mathrm{d}}^K$ and $m_4(K,N) = \lambda_{\mathrm{f}}^N(1-p_{\mathrm{d}})^K$. So $m_3(K,N)/m_4(K,N) = \left(\frac{p_{\mathrm{d}}}{\lambda_{\mathrm{f}}(1-p_{\mathrm{d}})}\right)^K$ and K is the only remaining exponent.

Notice that the omitted proofs appear in Appendix.

Theorem 2: Suppose that $\lambda_{\mathrm{f}} > 0$ and $0 < p_{\mathrm{d}} < 1$. Then the mixing time of the Markov chain $\mathcal{M}$ is bounded by $\tau_x(\epsilon) \leq 4R^4K^2N(m_5(K,N) + \log\epsilon^{-1})$ for all $x \in \Omega$.

Remark 2: Let $\bar{\tau}(\epsilon)$ be the upper bound found in Theorem 2. $\bar{\tau}(\epsilon)$ is polynomial in K and N. If $m_3(K,N)/m_4(K,N)$ does not grow fast, *e.g.*, Remark 1, $\bar{\tau}(\epsilon) = O(K^2N(K\log K + N\log N + \log\epsilon^{-1}))$. If K is fixed, $\bar{\tau}(\epsilon) = O(N(N\log N + \log\epsilon^{-1}))$.

Let $p(\omega)$ be the distribution of the states of $\mathcal{M}$ after simulating Algorithm 1 for at least $\bar{\tau}(\epsilon)$ steps. Then the total variation distance satisfies $\|p - \pi\| \leq \epsilon$. So we can sample from p to estimate $\{\beta_{jk}\}$. However, there is a small bias in our estimates since we are not sampling from π. The following theorem gives an upper bound on the number of samples needed for finding good estimates.

Theorem 3: Let $0 < \epsilon_1, \epsilon_2 \leq 1$ and $0 < \eta < .5$. Suppose that $\|p - \pi\| \leq \epsilon$ for $\epsilon \leq \epsilon_1\epsilon_2/8$. Then, with a total of $504\epsilon_1^{-2}\epsilon_2^{-1}\lceil\log\eta^{-1}\rceil$ samples from p, we can find estimates $\hat{\beta}_{jk}$ for β_{jk} with probability at least $1-\eta$, such that, for $\beta_{jk} \geq \epsilon_2$, $\hat{\beta}_{jk}$ estimates β_{jk} within ratio $1+\epsilon_1$, *i.e.*, $(1-\epsilon_1)\beta_{jk} \leq \hat{\beta}_{jk} \leq (1+\epsilon_1)\beta_{jk}$, and, for $\beta_{jk} < \epsilon_2$, $|\hat{\beta}_{jk} - \beta_{jk}| \leq (1+\epsilon_1)\epsilon_2$.

Remark 3: Following Remark 2, for fixed K, $\bar{\tau}(\epsilon) = O(N(N\log N + \log\epsilon^{-1}))$. Combining this fact with Theorem 3, the time complexity of the overall procedure is

$$T = O(\epsilon_1^{-2}\epsilon_2^{-1}\log\eta^{-1}N(N\log N + \log(\epsilon_1^{-1}\epsilon_2^{-1}))).$$

Hence, with a total of T samples, Algorithm 1 finds estimates $\hat{\beta}_{jk}$ for β_{jk} with probability at least $1-\eta$, such that, for $\beta_{jk} \geq \epsilon_2$, $\hat{\beta}_{jk}$ estimates β_{jk} within ratio $1+\epsilon_1$, and, for $\beta_{jk} < \epsilon_2$, $|\hat{\beta}_{jk} - \beta_{jk}| \leq (1+\epsilon_1)\epsilon_2$. We can simplify further by letting $\epsilon_0 = \epsilon_1\epsilon_2$. Then the time complexity is $O(\epsilon_0^{-2}\log\eta^{-1}N(N\log N + \log(\epsilon_0^{-1})))$.

D. Simulation Results

In this section, we show a simulation confirming our findings from last section. Since our goal is to estimate the association probabilities, we define the variation distance between two sets of association probabilities β_{jk} and $\hat{\beta}_{jk}$ by $\Delta_\beta = \max_{j,k}|\beta_{jk} - \hat{\beta}_{jk}|$. Each β_{jk} is computed exactly by JPDA and each $\hat{\beta}_{jk}$ is estimated by MCMCDA. The upper bound on the number of required samples found in Section IV-C is based on the worst-case analysis and, in practice, MCMC finds solutions much faster in most cases. We use the most common method to estimate $\hat{\beta}_{jk}$ following [26]:

$$\hat{\beta}_{jk} = \frac{1}{n_{\text{mc}} - n_{\text{bi}}} \sum_{n=n_{\text{bi}}}^{n_{\text{mc}}} \mathbb{I}((j,k) \in \omega_n),$$

where n_{mc} and n_{bi} are the total number of samples and the number of initial burn-in samples, respectively, and ω_n is the n-th sample. A simple case is chosen to demonstrate MCMCDA, in which two predicted observations are located at $[0,1]^T$ and $[0,-1]^T$. $\hat{P}^k(y_t^j|y_{1:t-1})$ has a Gaussian distribution with zero mean and covariance $B^k = \text{diag}(1,1)$ for $k \in \{1,2\}$. There are 15 observations as shown in Figure 2 (left). Other parameters are: $\delta^k = p((y_t^j - \hat{y}^k)^T(B^k)^{-1}(y_t^j - \hat{y}^k) = 4)$ for $k \in \{1,2\}$, $V = 16$, $\lambda_{\text{f}} = .8125$, and $p_{\text{d}} = .98$. In Figure 2 (right), the average variation distance between two sets of association probabilities β_{jk} and $\hat{\beta}_{jk}$ from 10 independent runs is shown as a function of number of samples.

V. Multi-Scan MCMCDA

In this section, we present an algorithm for solving the multiple-target tracking problem described in Section II. The algorithm is presented in Section V-A and its performance is compared against MHT in Section V-B.

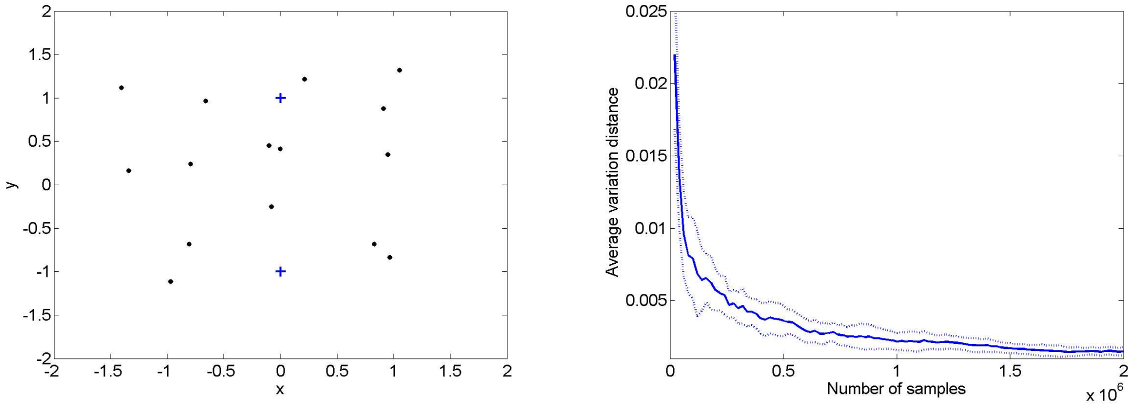


Fig. 2. (left) Expected observations (crosses) and observations (dots). (right) Average variation Δ_β as a function of the total number of samples (solid line). The dotted lines show the sample standard deviation from the average.

A. *Multi-Scan MCMCDA Algorithm*

The multi-scan MCMCDA algorithm is described in Algorithm 2. It is an MCMC algorithm whose state space is Ω as defined in Section II-B and whose stationary distribution is the posterior (3). The proposal distribution for MCMCDA consists of five types of moves (a total of eight moves). They are (1) birth/death move pair; (2) split/merge move pair; (3) extension/reduction move pair; (4) track update move; and (5) track switch move. The MCMCDA moves are graphically illustrated in Figure 3. We index each move by an integer such that $m = 1$ for a birth move, $m = 2$ for a death move and so on. The move m is chosen randomly from the distribution $\xi_K(m)$ where K is the number of tracks of the current partition ω. When there is no track, we can only propose a birth move, so we set $\xi_0(m = 1) = 1$ and 0 for all other moves. When there is only a single target, we cannot propose a merge or track switch move, so $\xi_1(m = 4) = \xi_1(m = 8) = 0$. For other values of K and m, we assume $\xi_K(m) > 0$. The inputs for MCMCDA are the set of all observations Y, the number of samples n_{mc}, the initial state ω_{init}, and a bounded function $X : \Omega \to \mathbb{R}^n$. At each step of the algorithm, ω is the current state of the Markov chain. The acceptance probability $A(\omega, \omega')$ is defined in (4) where $\pi(\omega) = P(\omega|Y)$ from (3). The output $\hat{X}$ approximates the MMSE estimate $\mathbb{E}_\pi X$ and $\hat{\omega}$ approximates the MAP estimate $\arg\max P(\omega|Y)$. The computation of $\hat{\omega}$ can be considered as simulated annealing at a constant temperature. Notice that MCMCDA can provide both MAP and MMSE solutions to the multiple-target tracking problem.

An MCMC algorithm can be specialized and made more efficient by incorporating the domain

Algorithm 2 Multi-Scan MCMCDA

```
Input: $Y, n_{\text{mc}}, \omega_{\text{init}}, X : \Omega \to \mathbb{R}^n$
Output: $\hat{\omega}, \hat{X}$
  $\omega = \omega_{\text{init}}$; $\hat{\omega} = \omega_{\text{init}}$; $\hat{X} = 0$
  for $n = 1$ to $n_{\text{mc}}$ do
    propose $\omega'$ based on $\omega$ (see sections from V-A.1 to V-A.5)
    sample $U$ from Unif$[0, 1]$
    $\omega = \omega'$ if $U < A(\omega, \omega')$
    $\hat{\omega} = \omega$ if $p(\omega|Y)/p(\hat{\omega}|Y) > 1$
    $\hat{X} = \frac{n}{n+1}\hat{X} + \frac{1}{n+1}X(\omega)$
  end for
```

specific knowledge. In multiple-target tracking, we can make two assumptions: (1) the maximum directional speed of any target in $\mathcal{R}$ is less than $\bar{v}$; and (2) the number of consecutive missing observations of any track is less than $\bar{d}$. The first assumption is reasonable in a surveillance scenario since, in many cases, the maximum speed of a vehicle is generally known based on the vehicle type and terrain conditions. The second assumption is a user-defined parameter. Let $p_{\text{dt}}(s) = 1 - (1 - p_{\text{d}})^s$ be the probability that an object is observed at least once out of s measurement times. Then, for given $\bar{p}_{\text{dt}}$, we set $\bar{d} \geq \log(1 - \bar{p}_{\text{dt}})/\log(1 - p_{\text{d}})$ to detect a track with probability at least $\bar{p}_{\text{dt}}$. For example, given $p_{\text{d}} = .7$ and $\bar{p}_{\text{dt}} = .99$, a track is detected with probability larger than $.99$ for $\bar{d} \geq 4$. We will now assume that these two new conditions are added to the definition of Ω so each element $\omega \in \Omega$ satisfies these two additional assumptions.

We use a data structure, a neighborhood tree of observations, which groups temporally separated observations based on distances, to propose a new partition ω' in Algorithm 2. A neighborhood tree of observations is defined as

$$L_d(y_t^j) = \{y_{t+d}^k \in y_{t+d} : \|y_t^j - y_{t+d}^k\| \leq d \cdot \bar{v}\}$$

for $d = 1, \ldots, \bar{d}$, $j = 1, \ldots, n_t$ and $t = 1, \ldots, T - 1$. Here $\|\cdot\|$ is the usual Euclidean distance. The parameter d allows missing observations. The use of this neighborhood tree makes the algorithm more scalable since distant observations will be considered separately and makes the computations of the proposal distribution easier. It is similar to the clustering technique used in

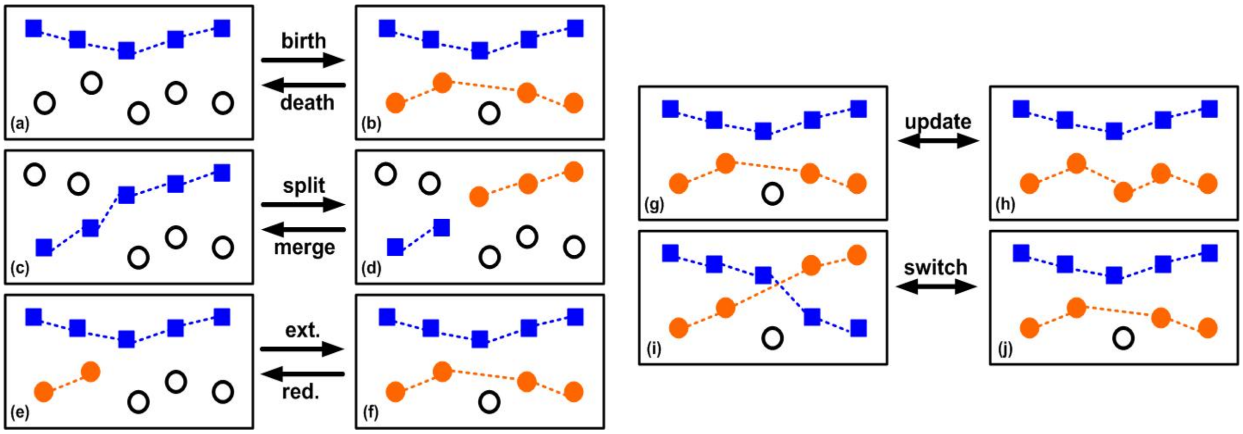


Fig. 3. Graphical illustration of MCMCDA moves (associations are indicated by dotted lines and rings are false alarms)

MHT but L_d is fixed for a given set of observations.

We now describe each move of the sampler in detail. First, let $\zeta(d)$ be a distribution of a random variable d taking values from $\{1, 2, \ldots, \bar{d}\}$. We assume the current state of the chain is $\omega = \omega^0 \cup \omega^1 \in \Omega$, where $\omega^0 = \{\tau_0\}$ and $\omega^1 = \{\tau_1, \ldots, \tau_K\}$. The proposed partition is denoted by $\omega' = \omega'^0 \cup \omega'^1 \in \Omega$. Note the abuse of notation below with indexing of time, *i.e.*, when we say $\tau(t_i)$, t_i means the time at which a target corresponding to the track τ is observed i times.

1) Birth and Death Moves (Fig. 3, $a \leftrightarrow b$): For a birth move, we increase the number of tracks from K to $K' = K+1$ and select t_1 uniformly at random (u.a.r.) from $\{1, \ldots, T-1\}$ as an appearance time of a new track. Let $\tau_{K'}$ be the track of this new object. Then we choose d_1 from the distribution ζ. Let $L^1_{d_1} = \{y^j_{t_1} : L_{d_1}(y^j_{t_1}) \neq \emptyset, y^j_{t_1} \notin \tau_k(t_1), j = 1, \ldots, n_{t_1}, k = 1, \ldots, K\}$. $L^1_{d_1}$ is a set of observations at t_1 such that, for any $y \in L^1_{d_1}$, y does not belong to other tracks and y has at least one descendant in $L_{d_1}(y)$. We choose $\tau_{K'}(t_1)$ u.a.r. from $L^1_{d_1}$. If $L^1_{d_1}$ is empty, the move is rejected since the move is not reversible. Once the initial observation is chosen, we then choose the subsequent observations for the track $\tau_{K'}$. For $i = 2, 3, \ldots$, we choose d_i from ζ and choose $\tau_{K'}(t_i)$ u.a.r. from $L_{d_i}(\tau_{K'}(t_{i-1})) \setminus \{\tau_k(t_{i-1}+d_i) : k = 1, \ldots, K\}$ unless this set is empty. But, for $i = 3, 4, \ldots$, the process of adding observations to $\tau_{K'}$ terminates with probability γ, where $0 < \gamma < 1$. If $|\tau_{K'}| \leq 1$, the move is rejected. We then propose this modified partition where $\omega'^1 = \omega^1 \cup \{\tau_{K'}\}$ and $\omega'^0 = \{\tau_0 \setminus \tau_{K'}\}$. For a death move, we simply choose k u.a.r. from $\{1, \ldots, K\}$ and delete the k-th track and propose a new partition where $\omega'^1 = \omega^1 \setminus \{\tau_k\}$ and $\omega'^0 = \{\tau_0 \cup \tau_k\}$.

2) Split and Merge Moves (Fig. 3, $c \leftrightarrow d$): For a split move, we select $\tau_s(t_r)$ u.a.r. from $\{\tau_k(t_i) : |\tau_k| \geq 4, i = 2, \ldots, |\tau_k| - 2, k = 1, \ldots, K\}$. Then we split the track τ_s into τ_{s_1} and τ_{s_2} such that $\tau_{s_1} = \{\tau_s(t_i) : i = 1, \ldots, r\}$ and $\tau_{s_2} = \{\tau_s(t_i) : i = r + 1, \ldots, |\tau_s|\}$. The modified track partition becomes $\omega'^1 = (\omega^1 \setminus \{\tau_s\}) \cup \{\tau_{s_1}\} \cup \{\tau_{s_2}\}$ and $\omega'^0 = \omega^0$. For a merge move, we consider the set

$$M = \{(\tau_{k_1}(t_f), \tau_{k_2}(t_1)) : \tau_{k_2}(t_1) \in L_{t_1 - t_f}(\tau_{k_1}(t_f)), f = |\tau_{k_1}| \text{ for } k_1 \neq k_2, 1 \leq k_1, k_2 \leq K\}.$$

We select a pair $(\tau_{s_1}(t_f), \tau_{s_2}(t_1))$ u.a.r. from M. The tracks are combined into a single track $\tau_s = \tau_{s_1} \cup \tau_{s_2}$. Then we propose a new partition where $\omega'^1 = (\omega^1 \setminus (\{\tau_{s_1}\} \cup \{\tau_{s_2}\})) \cup \{\tau_s\}$ and $\omega'^0 = \omega^0$.

3) Extension and Reduction Moves (Fig. 3, $e \leftrightarrow f$): In a track extension move, we select a track τ u.a.r. from K available tracks in ω. We reassign observations for τ after the disappearance time $t_{|\tau|}$ as done in the track birth move. For a track reduction move, we select a track τ u.a.r. from K available tracks in ω and r u.a.r. from $\{2, \ldots, |\tau| - 1\}$. We shorten the track τ to $\{\tau(t_1), \ldots, \tau(t_r)\}$ by removing the observations assigned to τ after the time t_{r+1}.

4) Track Update Move (Fig. 3, $g \leftrightarrow h$): In a track update move, we select a track τ u.a.r. from K available tracks in ω. Then we pick r u.a.r. from $\{1, 2, \ldots, |\tau|\}$ and reassign observations for τ after the time t_r as done in the track birth move.

5) Track Switch Move (Fig. 3, $i \leftrightarrow j$): For a track switch move, we select a pair of observations $(\tau_{k_1}(t_p), \tau_{k_2}(t_q))$ from two different tracks such that, $\tau_{k_1}(t_{p+1}) \in L_d(\tau_{k_2}(t_q))$ and $\tau_{k_2}(t_{q+1}) \in L_{d'}(\tau_{k_1}(t_p))$, where $d = t_{p+1} - t_q$, $d' = t_{q+1} - t_p$ and $0 < d, d' \leq \bar{d}$. Then we let

$$\begin{aligned}
\tau_{k_1} &= \{\tau_{k_1}(t_1), \ldots, \tau_{k_1}(t_p), \tau_{k_2}(t_{q+1}), \ldots, \tau_{k_2}(t_{|\tau_{k_2}|})\} \\
\tau_{k_2} &= \{\tau_{k_2}(t_1), \ldots, \tau_{k_2}(t_q), \tau_{k_1}(t_{p+1}), \ldots, \tau_{k_1}(t_{|\tau_{k_1}|})\}.
\end{aligned}$$

We now show that MCMCDA is an optimal Bayesian filter in the limit. Let $\mathcal{M}$ be the Markov chain specified by Algorithm 2.

Lemma 1: Suppose that $0 < p_{\mathrm{z}}, p_{\mathrm{d}} < 1$ and $\lambda_{\mathrm{b}}, \lambda_{\mathrm{f}} > 0$. If $\zeta(d) > 0$, for all $d \in \{1, \ldots, \bar{d}\}$, then the Markov chain $\mathcal{M}$ is irreducible. (See [27] for the proof.)

Theorem 4: Under the assumptions in Lemma 1, the Markov chain $\mathcal{M}$ is ergodic and $\hat{X} \to \mathbb{E}_\pi X$ as $n_{\mathrm{mc}} \to \infty$.

Proof: From Lemma 1, $\mathcal{M}$ is irreducible. $\mathcal{M}$ is aperiodic since there is always a positive probability of staying at the current state in the track update move. Now the transitions described in Algorithm 2 satisfy the detailed balance condition since it uses the Metropolis-Hastings kernel (4). Hence, by the ergodic theorem [22], the chain converges to its stationary distribution $\pi(\omega)$ almost surely and $\hat{X} \to \mathbb{E}_\pi X$ as $n_{\text{mc}} \to \infty$. ■

B. Simulation Results

For the simulations we consider surveillance over a rectangular region on a plane, $\mathcal{R} = [0, L] \times [0, L] \subset \mathbb{R}^2$. The state vector is $x = [x, y, \dot{x}, \dot{y}]^T$ where (x, y) is a position on $\mathcal{R}$ along the usual x and y axes and $(\dot{x}, \dot{y})$ is a velocity vector. The linear dynamics and measurement model are used:

$$x_{t+1}^k = Ax_t^k + Gw_t^k \qquad\qquad y_t^j = Cx_t^k + v_t^j$$

where

$$A = \begin{bmatrix} 1 & 0 & T_{\text{s}} & 0 \\ 0 & 1 & 0 & T_{\text{s}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad G = \begin{bmatrix} T_{\text{s}}^2/2 & 0 \\ 0 & T_{\text{s}}^2/2 \\ T_{\text{s}} & 0 \\ 0 & T_{\text{s}} \end{bmatrix} \qquad C = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}^T ,$$

and T_{s} is the sampling period, w_t^k is a zero-mean Gaussian process with covariance $Q = \text{diag}(100, 100)$, and v_t^j is a zero-mean Gaussian process with covariance $R = \text{diag}(25, 25)$.

The complexity of multiple-target tracking problems can be measured by several metrics: (1) the intensity of the false alarm rate λ_{f}; (2) the detection probability p_{d}; and (3) the density of tracks. The problem gets more challenging with increasing λ_{f}, decreasing p_{d}, increasing K, and increasing density of tracks. The number of tracks itself may not make the problem more difficult if they are scattered apart. The difficulty arises when there are many tracks that are moving closely and crossing each other; this is when the ambiguity of data association is greater. Hence, we only consider situations in which tracks move very closely so we can control the density of tracks by the number of tracks. We study the performance of the MCMCDA algorithm against the greedy algorithm and MHT by varying the parameters listed above. To make the comparison easier, we take the MAP approach, in which the states of targets are estimated from $\hat{\omega}$ computed from Algorithm 2. The greedy algorithm is a batch-mode nearest neighbor multiple-target tracking algorithm. The algorithm first marks all observations as false alarms, and then picks two unmarked observations at different times to estimate an initial state. Then it forms a canditate track by picking unmarked observations which are the nearest to the predicted states for subsequent time steps. The candidate track is validated as a track and observations associated to the candidate track are marked if the marginal of the candidate track exceeds a threshold. The process is repeated until no more tracks can be found.

Since the number of targets is not fixed, it is difficult to compare algorithms using a standard criterion such as the mean square error. Hence, we introduce two new metrics to measure the effectiveness of each data association algorithm: the normalized correct associations (NCA) and incorrect-to-correct association ratio (ICAR)

$$\begin{aligned}\text{NCA} &= \frac{\text{number of correct associations}}{\text{number of associations in test case}}\\ \text{ICAR} &= \frac{\text{number of incorrect associations}}{\text{number of correct associations}}.\end{aligned}$$

We measure the performance of each algorithm by NCA, ICAR, the estimation error in the number of tracks, $K_{\text{err}} = ||\omega^*| - |\omega||$, and the running time of the algorithm.

Both MCMCDA and greedy algorithms are written in C++ with Matlab interfaces. We have used the C++ implementation of MHT [28], which implements pruning, gating, clustering, N-scan-back logic and k-best hypotheses. The parameters for MHT are fine-tuned so that it gives similar performance as MCMCDA when there are 10 targets: the maximum number of hypotheses in a group is 1000, the maximum track tree depth is 5, and the maximum Mahalanobis distance is 5.9. All simulations are run on a PC with a 2.6-GHz Intel processor.

1) Experiment I (Number of Tracks): In this experiment, we vary K from 5 to 100. The other parameters are held fixed: $\mathcal{R} = [0, 1000] \times [0, 1000]$, $T = 10$, $\lambda_{\text{f}} V = 1$, $\bar{d} = 1$, $\bar{v} = 130$ unit lengths per unit time. Since all tracks are observed, the number of observations increases as the number of tracks increases. The results for MCMCDA are the average values over 10 repeated runs and the initial state is initialized with the greedy algorithm and 10,000 samples are used. The average NCAs, ICARs, the estimation error in the numbers of tracks and the running times for three different algorithms are shown in Figure 4 and Figure 5 (the running time of MCMCDA includes the initialization step). Although the maximum number of hypotheses of 1000 per group is a large number, with increasing numbers of tracks, the performance of MHT deteriorates due to pruning. But both greedy and MCMCDA maintain good performance, although the greedy algorithm detects fewer tracks for large K. In addition, the running times of both greedy and MCMCDA are significantly less than that of MHT.

2) Experiment II (False Alarms): Now the settings are the same as Experiment I but we vary the false alarm rates while the number of tracks is fixed at $K = 10$. The false alarm rates are

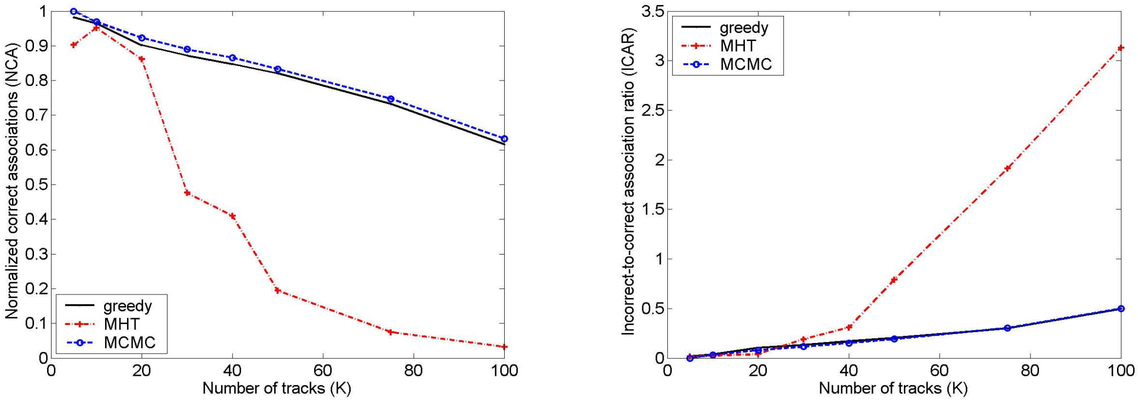


Fig. 4. NCA (left) and ICAR (right) as functions of K

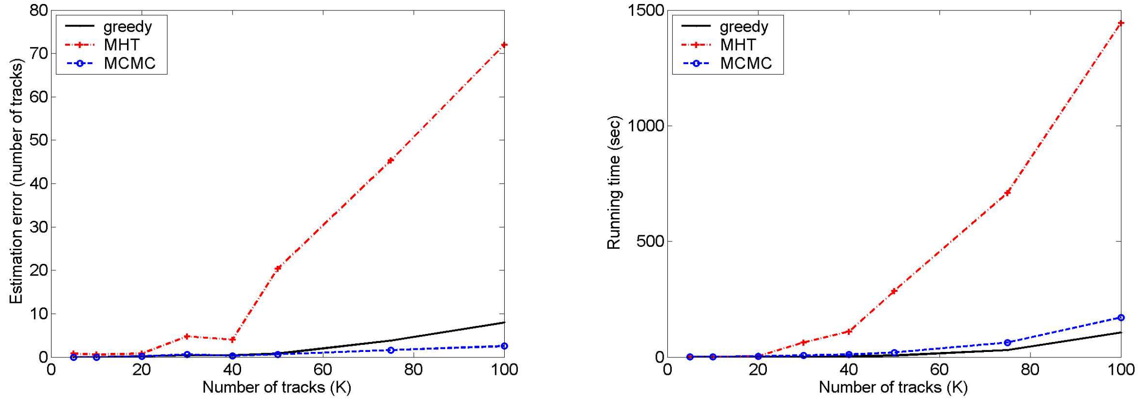


Fig. 5. The estimation error in K, the number of tracks (left), and average running time (right) as functions of K.

varied from $\lambda_{\mathrm{f}}V = 1$ to $\lambda_{\mathrm{f}}V = 100$ with an increment of 10. Again, 10,000 samples are used for MCMCDA. The average NCAs, ICARs and the estimation error in the numbers of tracks for three different algorithms at different false alarm rates are shown in Figure 6 and Figure 8 (left). It shows the remarkable performance of MCMCDA at high false alarm rates while the other two algorithms perform poorly. The greedy algorithm scores slightly higher in NCA than MCMCDA but poorly in ICAR. In addition, it reports spurious tracks at high false alarm rates. Notice that MHT does not make any correct associations at high false alarm rates, $\lambda_{\mathrm{f}}V \geq 80$, so ICARs for MHT at $\lambda_{\mathrm{f}}V \geq 80$ are not reported.

3) Experiment III (Detection Probability): The detection probability p_{d} is varied from 0.3 to 0.9 with an increment of 0.1 while keeping the other parameters as the previous experiments except $K = 10$, $\lambda_{\mathrm{f}}V = 1$, $T = 15$ and $\bar{d} = 5$. Now the tracks are not observed all the time.

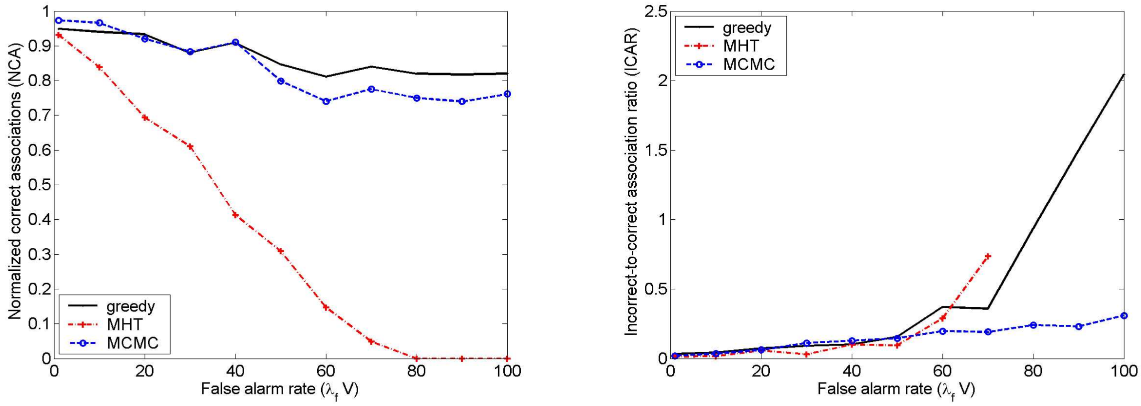


Fig. 6. NCA (left) and ICAR (right) as functions of false alarm rate

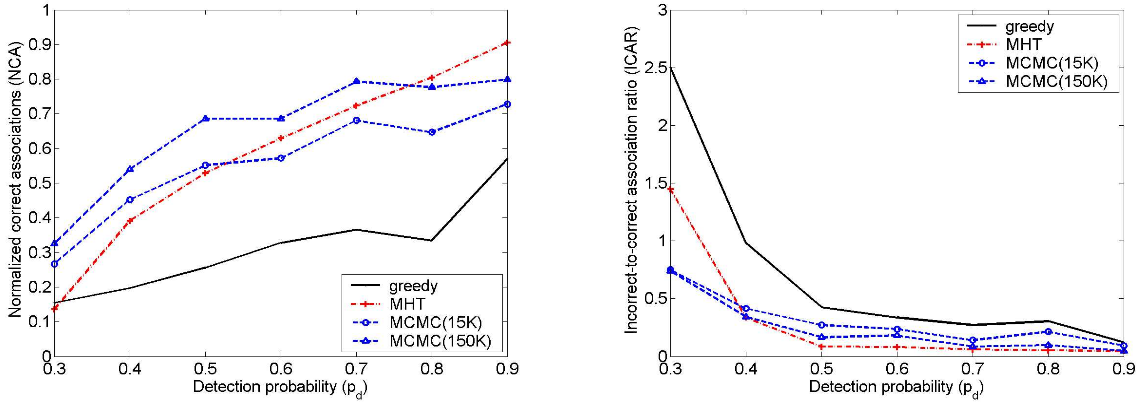


Fig. 7. NCA (left) and ICAR (right) as functions of detection probability

The average NCAs, ICARs and the estimation error in the numbers of tracks are shown in Figure 7 and Figure 8 (right). For MCMCDA, we present two cases: MCMC(15K) with 15,000 samples and MCMC(150K) with 150,000 samples. It shows that MCMCDA outperforms the other algorithms at low detection probabilities. At high detection probabilities, MHT scores better than MCMCDA but it reports a higher number of tracks, meaning that it fragments tracks.

Although, in theory, MHT gives an optimal solution in the sense of MAP, it performs poorly in practice when the detection probability is low or the false alarm rate is high due to the heuristics such as pruning and N-scan-back techniques used to reduce the complexity. The heuristics are required parts of MHT. Without the pruning and N-scan-back logic, the problem complexity grows exponentially fast even for a small problem. In practice, MHT with heuristics works well when a few hypotheses carry most of the weight. When the detection probability is low or

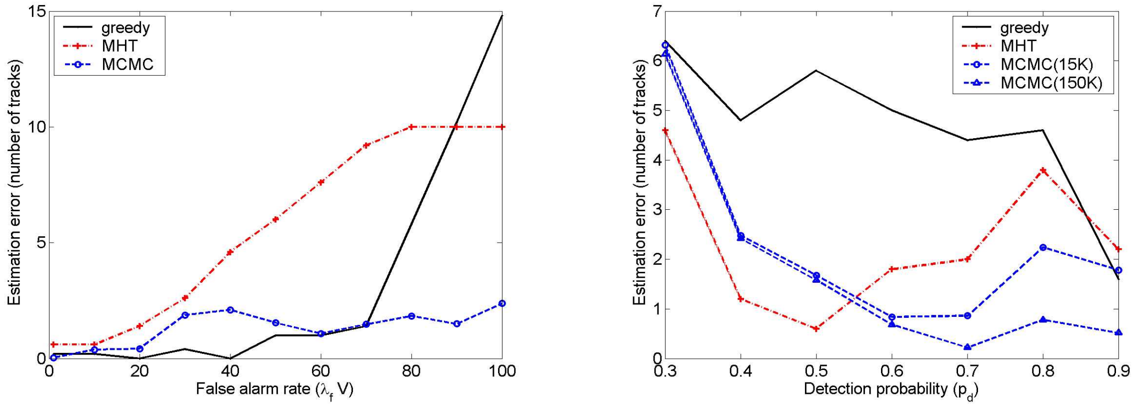


Fig. 8. (left) K_{err} as a function of false alarm rate. (right) K_{err} as a function of detection probability

the false alarm rate is high, there are many hypotheses with appreciable weights and there is no small set of dominating hypotheses, so MHT cannot perform well. In addition, when the detection probability is high, MHT again suffers from a large number of observations. Another noticeable benefit of the MCMCDA algorithm is that its running time can be regulated by the number of samples and the number of observations but the running time of MHT depends on the complexity of the problem instance, which is not predictable in advance.

4) Online MCMCDA Multiple-Target Tracker: The extension of MCMCDA to an online, real-time tracking is a trivial task. We implement a sliding window of size w_{s} using Algorithm 2. At each time step, we use the previous estimate to initialize MCMCDA and run MCMCDA on the observations belonging to the current window. A total of three test cases are generated: (case 1) 100 tracks, (case 2) 200 tracks and (case 3) 300 tracks. The surveillance duration is increased to $T = 1000$ and the surveillance region is now $\mathcal{R} = [0, 10000] \times [0, 10000]$. The other parameters are: $\lambda_{\text{f}} V = 10$, $p_{\text{d}} = .9$, $\bar{d} = 3$, $\bar{v} = 230$ and $w_{\text{s}} = 10$. The objects appear and disappear at random in time and space so the number of tracks changes in time. These test cases represent instances of the general (discrete-time) multiple-target tracking problem. The average NCAs and ICARs over the sliding window and the average execution time per simulation time are shown in Table I. Notice that MCMCDA achieves excellent performance in all cases with less than a second of execution time.

TABLE I

PERFORMANCE OF ONLINE MCMCDA TRACKER: RUNNING TIME (RT) IN SECONDS

	Number of samples					
	1,000			5,000		
K	NCA	ICAR	RT	NCA	ICAR	RT
100	.95	.19	.06	.98	.13	.28
200	.94	.06	.09	.97	.05	.41
300	.92	.07	.11	.97	.05	.55

C. Experiment

We have applied our algorithm to track people from video sequences taken by a stationary camera. The sampling rate was 2 frames per second. Since we do not know the initial positions of appearing objects, a tracking algorithm must be able to initiate and terminate tracks, hence this application is a test of the algorithm presented in the previous section. We applied a simple background subtraction algorithm to detect moving objects and Algorithm 2 is used to track these objects. Some selected sequences are shown in Figure 9 along with the estimated tracks. Figure 10 illustrates some failures of background subtraction as an object detection method. The top row of Figure 10 shows a person walking under a tree who is not detected for six frames. The bottom row of Figure 10 shows the case when the detection algorithm reports a single detection when one person is occluded by another. In both cases, a person is not detected for many frames, but MCMCDA was able to resume the track when the person appears again, showing the robustness of the algorithm against missing observations.

VI. CONCLUSIONS

In this paper, we have presented Markov chain Monte Carlo data association (MCMCDA) for solving data association problems arising in multiple-target tracking in a cluttered environment. For the case of a fixed number of targets, we have shown that a single-scan MCMCDA algorithm provides a fully polynomial randomized approximation scheme for the calculation involved in the JPDA filter, which is known to be NP-hard. For the general multiple-target tracking problem, in which an unknown number of targets appears and disappears at random times, we have presented a multi-scan MCMCDA algorithm that is capable of initiating and terminating an unknown number of tracks. The MCMCDA algorithm is flexible and can easily incorporate any

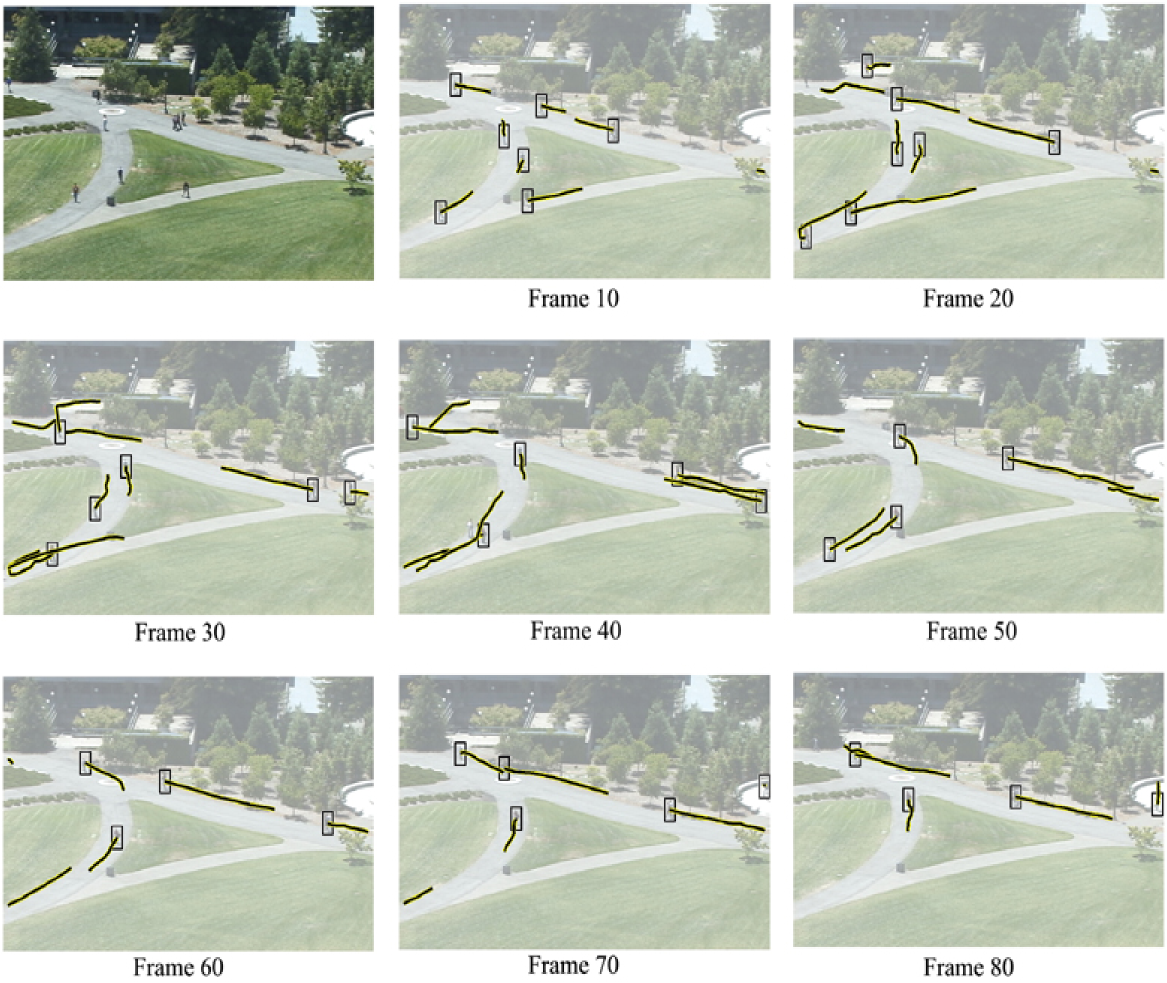
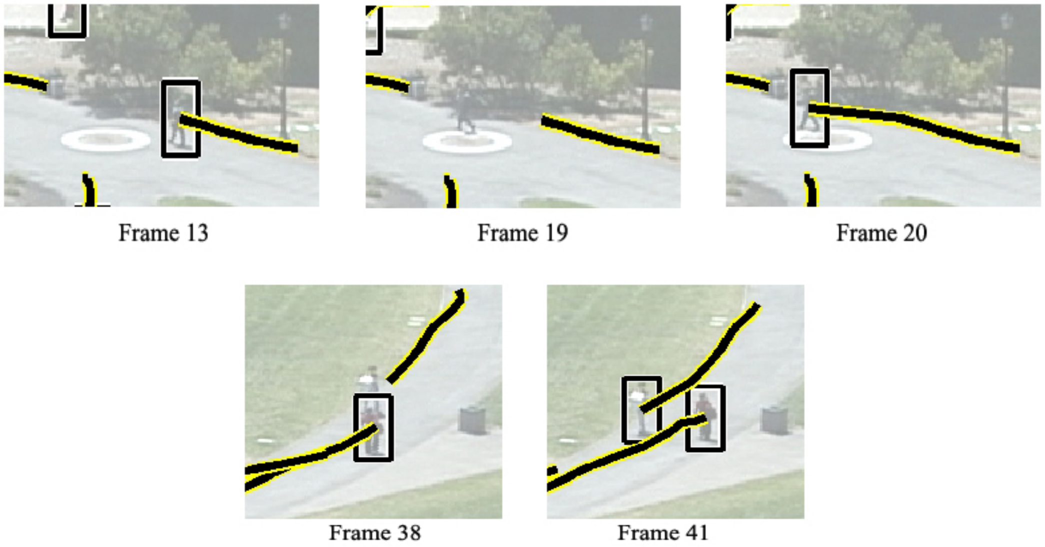


Fig. 9. Tracking people from video sequences. Some selected sequences are shown for illustration. Detections are shown in boxes and tracks are shown in solid lines. (Video scenes courtesy of Parvez Ahammad)

domain specific knowledge to make it more efficient. Instead of enumerating the entire space of associations, MCMCDA randomly samples the region where the posterior is concentrated. Our simulation results show the remarkable performance of the MCMCDA algorithm under extreme conditions such as a large number of targets in a dense environment, low detection probabilities, and high false alarm rates. We have also shown that the algorithm can be formulated as an online, real-time algorithm with excellent performance.

VII. Appendix

The proofs shown here parallel the proofs by Jerrum and Sinclair [20] in both structure and details; the main differences are its application to the well-known data association problem in



Frame 13 Frame 19 Frame 20

Frame 38 Frame 41

Fig. 10. Examples of missing detections. (top) A person walking under a tree is not detected from frame 14 to frame 19. (bottom) A person is occluded by another person and not detected from frame 38 to frame 40

multiple-target tracking and the use of the non-uniform likelihood function into the posterior, allowing us to understand the relationship between the parameters in the multiple-target tracking problem and the mixing time of the Markov chain.

A. Proof of Theorem 2

To prove Theorem 2, we need the following lemmas.

Lemma 2: Let $C = \frac{p_{\mathrm{d}}\bar{L}}{\lambda_{\mathrm{f}}(1-p_{\mathrm{d}})}$ and $D = \frac{\lambda_{\mathrm{f}}(1-p_{\mathrm{d}})}{\underline{L}p_{\mathrm{d}}}$. For any $\omega_0, \omega_1, \omega_2 \in \Omega$, if $\omega_1 = \omega_0 - e_0$, for some edge $e_0 \in \omega_0$, and $\omega_2 = \omega_1 - e_1$, for some edge $e_1 \in \omega_1$, then:

$$\begin{aligned} \pi(\omega_0)/\pi(\omega_1) &\leq C \\ \pi(\omega_0)/\pi(\omega_2) &\leq C^2 \end{aligned} \quad \text{and} \quad \begin{aligned} \pi(\omega_1)/\pi(\omega_0) &\leq D \\ \pi(\omega_2)/\pi(\omega_0) &\leq D^2. \end{aligned}$$

Proof: ω_0 and ω_1 are identical except that ω_1 is missing the edge e_0. So $|\omega_0| = |\omega_1| + 1$. If $e_0 = (u, v)$ and $k = |\omega_0|$,

$$\begin{aligned} \pi(\omega_0)/\pi(\omega_1) &= \frac{\lambda_{\mathrm{f}}^{N-k} p_{\mathrm{d}}^{k} (1-p_{\mathrm{d}})^{K-k}}{\lambda_{\mathrm{f}}^{N-(k-1)} p_{\mathrm{d}}^{k-1} (1-p_{\mathrm{d}})^{K-(k-1)}} \hat{P}^v(u|y_{1:t-1}) \\ &= \frac{p_{\mathrm{d}}}{\lambda_{\mathrm{f}}(1-p_{\mathrm{d}})} \hat{P}^v(u|y_{1:t-1}) \qquad \leq C. \end{aligned}$$

On the other hand,

$$\begin{aligned} \pi(\omega_1)/\pi(\omega_0) &= \frac{\lambda_{\mathrm{f}}^{N-(k-1)} p_{\mathrm{d}}^{k-1} (1-p_{\mathrm{d}})^{K-(k-1)}}{\lambda_{\mathrm{f}}^{N-k} p_{\mathrm{d}}^{k} (1-p_{\mathrm{d}})^{K-k}} \frac{1}{\hat{P}^v(u|y_{1:t-1})} \\ &= \frac{\lambda_{\mathrm{f}}(1-p_{\mathrm{d}})}{p_{\mathrm{d}}} \frac{1}{\hat{P}^v(u|y_{1:t-1})} \qquad \leq D. \end{aligned}$$

Since $\pi(\omega_0)/\pi(\omega_2) = \pi(\omega_0)/\pi(\omega_1) \times \pi(\omega_1)/\pi(\omega_2)$, by repeating the above argument twice, we get $\pi(\omega_0)/\pi(\omega_2) \leq C^2$. Similarly, we have $\pi(\omega_2)/\pi(\omega_0) \leq D^2$. ■

Lemma 3: Let $R = \max\{1, C, D\}$, where C and D are defined in Lemma 2. Then the maximum edge loading of the Markov chain $\mathcal{M}$ is bounded as $\bar{\rho} \leq 4R^4K^2N$.

Proof: For each pair of matchings X, Y in G, we define the canonical path γ_{XY} as in [20]. Consider the symmetric difference $X \oplus Y$, where $X \oplus Y = (X - Y) \cup (Y - X)$. $X \oplus Y$ is a disjoint collection of paths in G including closed cycles, each of which has edges that belong to X and Y alternately. Suppose that we have fixed some arbitrary ordering on all simple paths in G, and designate a "start vertex" to each of the paths, which is arbitrary if the path is a closed cycle but must be an endpoint otherwise. This gives a unique ordering $P_1, P_2, \ldots, P_m$ on the paths appearing in $X \oplus Y$. The canonical path from X to Y involves "unwinding" each of the P_i in turn as follows. We need to consider two cases:

(i) P_i is not a cycle. Let P_i consist of the sequence $(v_0, v_1, \ldots, v_l)$ of vertices with the start vertex v_0. If $(v_0, v_1) \in Y$, perform a sequence of switching moves replacing (v_{2j+1}, v_{2j+2}) by (v_{2j}, v_{2j+1}) for $j = 0, 1, \ldots$, and finish with an addition move if l is odd. If $(v_0, v_1) \in X$, remove (v_0, v_1) and proceed as before for the reduced path $(v_1, \ldots, v_l)$.

(ii) P_i is a cycle. Let P_i consist of the sequence $(v_0, v_1, \ldots, v_{2l+1})$ of vertices, for $l \geq 1$, where v_0 is the start vertex, and $(v_{2j}, v_{2j+1}) \in X$ for $j = 0, \ldots, l$, with remaining edges belonging to Y. We first remove the edge (v_0, v_1). Now we are left with an open path O with endpoints v_0, v_1, with the start vertex v_k of O, for $k \in \{0, 1\}$. Then we unwind O as in (i) above but treating v_{1-k} as the start vertex to identify that it was a cycle.

Let t be an arbitrary edge in the Markov chain $\mathcal{M}$, *i.e.*, a transition from ω to $\omega' \neq \omega$. Let $cp(t) = \{(X, Y) : \gamma_{XY} \ni t\}$ be the set of canonical paths that use t. We define a function $\eta_t : cp(t) \rightarrow \Omega$ as in [20],

$$\eta_t(X, Y) = \begin{cases} X \oplus Y \oplus (\omega \cup \omega') - e_{XY_t}, \\ \qquad \text{if } t \text{ is a switch move and the current path is a cycle;} \\ X \oplus Y \oplus (\omega \cup \omega'), \qquad \text{otherwise,} \end{cases}$$

where e_{XY_t} is the edge in X adjacent to the start vertex that was removed first in (ii) above. $\eta_t(X, Y)$ is always a matching in G and η_t is injective as shown in [20]. Notice that the bipartite graph G considered here is a subset of the graphs considered in [20] so the arguments about η_t

can be directly applied here.

Notice that

$$Q(t) \;=\; Q(\omega,\omega') = \pi(\omega)P(\omega,\omega') = \frac{1}{2|E|}\min\{\pi(\omega),\pi(\omega')\}. \qquad (17)$$

Next, we bound $\pi(X)\pi(Y)$ and we need to consider four cases:

(i) t is a deletion move. We have $\omega' = \omega - e$ and $\eta_t(X,Y) = X \oplus Y \oplus (\omega \cup \omega')$. Since $\omega \cup \eta_t(X,Y)$ and $X \cup Y$ are identical when viewed as multisets,

$$\begin{aligned}\pi(X)\pi(Y) &= \pi(\omega)\pi(\eta_t(X,Y)) = \frac{2|E|Q(t)}{\min\{\pi(\omega),\pi(\omega')\}}\pi(\omega)\pi(\eta_t(X,Y)) \\ &= 2|E|Q(t)\max\left\{1,\frac{\pi(\omega)}{\pi(\omega')}\right\}\pi(\eta_t(X,Y)) \le 2R|E|Q(t)\pi(\eta_t(X,Y)),\end{aligned}$$

where we used the identity (17) in the second equality and Lemma 2 for the last inequality.

(ii) t is an addition move. We have $\omega' = \omega + e$ and $\eta_t(X,Y) = X \oplus Y \oplus (\omega \cup \omega')$. Since $\omega \cup \eta_t(X,Y)$ and $X \cup Y$ are identical when viewed as multisets, using the arguments from (i),

$$\pi(X)\pi(Y) \;\le\; 2R|E|Q(t)\pi(\eta_t(X,Y)).$$

(iii) t is a switch move and the current path is a cycle. Suppose $\omega' = \omega + e - e'$. Let $\omega_1 = \omega + e$. Then $\omega' = \omega_1 - e'$. Since $\frac{\pi(\omega)}{\pi(\omega')} = \frac{\pi(\omega_1)}{\pi(\omega')}\frac{\pi(\omega)}{\pi(\omega_1)}$, by Lemma 2, $\frac{\pi(\omega)}{\pi(\omega')} \le CD \le R^2$. Since $\eta_t(X,Y) = X \oplus Y \oplus (\omega \cup \omega') - e_{XY_t}$, the multisets $\omega \cup \eta_t(X,Y)$ differs from $X \cup Y$ only in that e and e_{XY_t} are missing from it. Hence, by Lemma 2,

$$\begin{aligned}\pi(X)\pi(Y) &\le C^2\pi(\omega)\pi(\eta_t(X,Y)) = 2C^2|E|Q(t)\max\left\{1,\frac{\pi(\omega)}{\pi(\omega')}\right\}\pi(\eta_t(X,Y)) \\ &\le 2R^4|E|Q(t)\pi(\eta_t(X,Y)).\end{aligned}$$

(iv) t is a switch move and the current path is not a cycle. This case is similar to (iii) but the multisets $\omega \cup \eta_t(X,Y)$ differs from $X \cup Y$ only in that e is missing from it. Hence, by Lemma 2,

$$\begin{aligned}\pi(X)\pi(Y) &\le C\pi(\omega)\pi(\eta_t(X,Y)) = 2C|E|Q(t)\max\left\{1,\frac{\pi(\omega)}{\pi(\omega')}\right\}\pi(\eta_t(X,Y)) \\ &\le 2R^3|E|Q(t)\pi(\eta_t(X,Y)).\end{aligned}$$

In summary, we have, in all cases, $\pi(X)\pi(Y) \le 2R^4|E|Q(t)\pi(\eta_t(X,Y))$. Thus, for any transition t,

$$\begin{aligned}\frac{1}{Q(t)}\sum_{\gamma_{XY}\ni t}\pi(X)\pi(Y)|\gamma_{XY}| &\le 2R^4|E|\sum_{\gamma_{XY}\ni t}\pi(\eta_t(X,Y))|\gamma_{XY}| \\ &\le 4R^4K|E|\sum_{\gamma_{XY}\ni t}\pi(\eta_t(X,Y)) \\ &\le 4R^4K|E| \le 4R^4K^2N\end{aligned}$$

where the second inequality follows from the fact that the length of any canonical path is bounded by $2K$, the third equality is due to the fact that η_t is injective and π is a probability distribution, and the last inequality follows from $|E| \le KN$. Hence, $\bar{\rho} \le 4R^4K^2N$. ■

We now prove Theorem 2. $\mathcal{M}$ is a finite, reversible, ergodic Markov chain with loop probabilities $P(x,x) \geq \frac{1}{2}$ for all states x (see Section IV-B). Hence, by Theorem 1, we have

$$\tau_x(\epsilon) \leq \bar{\rho}(\log \pi(x)^{-1} + \log \epsilon^{-1}). \tag{18}$$

The upper bound for $\bar{\rho}$ is computed from Lemma 3. Now we just need to find the upper bound for $\pi(x)^{-1}$. From (16),

$$\begin{aligned} Z &\leq \sum_{\omega \in \Omega} m_1^K m_3(K,N) = m_1^K m_3(K,N)|\Omega| \\ &\leq m_1^K m_3(K,N) \sum_{k=0}^{K} \binom{K}{k} \frac{N!}{(N-k)!} \leq m_1^K m_3(K,N)(K+1)!N!, \end{aligned}$$

where the second inequality is by (15). Although this bound on Z is not tight, it will serve our purpose. For any $\omega \in \Omega$, $\pi(\omega) \geq \frac{1}{Z} m_2^K m_4(K,N)$ so

$$\frac{1}{\pi(\omega)} \leq \frac{Z}{m_2^K m_4(K,N)} \leq \left(\frac{m_1}{m_2}\right)^K \frac{m_3(K,N)}{m_4(K,N)}(K+1)!N!.$$

Hence,

$$\log \frac{1}{\pi(\omega)} \leq \log \left(\left(\frac{m_1}{m_2}\right)^K \frac{m_3(K,N)}{m_4(K,N)}(K+1)!N! \right) = m_5(K,N).$$

Putting all together, we have, for all initial state $x \in \Omega$, $\tau_x(\epsilon) \leq 4R^4K^2N(m_5(K,N) + \log \epsilon^{-1})$.

B. *Proof of Theorem 3*

Let $\beta_{\epsilon_2} = \{(j,k) : \beta_{jk} \geq \epsilon_2\}$. For now, assume $(j,k) \in \beta_{\epsilon_2}$, *i.e.*, $\beta_{jk} \geq \epsilon_2$. Let $X_{jk}(\omega) = \mathbb{I}((\hat{y}^k, y^j) \in \omega)$ where $\mathbb{I}$ is an indicator function. Notice that $\mathbb{E}_\pi(X_{jk}) = \pi(\omega_{jk}) = \beta_{jk}$, where $\omega_{jk} = \{\omega \in \Omega : (y^j, k) \in \omega\}$. Since $\|p - \pi\| \leq \epsilon$,

$$\begin{aligned} |p(\omega_{jk}) - \pi(\omega_{jk})| &\leq \epsilon \leq \frac{\epsilon_1 \pi(\omega_{jk})}{8} \\ |\mathbb{V}\text{ar}_p(X_{jk}) - \mathbb{V}\text{ar}_\pi(X_{jk})| &\leq 3\epsilon \leq \frac{3\epsilon_1 \pi(\omega_{jk})}{8}. \end{aligned} \tag{19}$$

Let $\bar{\beta}_{jk} = \frac{1}{s}\sum_{i=1}^{s} X_{jk}(\omega_i)$ be the sample mean of s samples from p. Then $\mathbb{E}(\bar{\beta}_{jk}) = p(\omega_{jk})$ and $\mathbb{V}\text{ar}(\bar{\beta}_{jk}) = \frac{1}{s}\mathbb{V}\text{ar}_p(X_{jk})$. By Chebyshev's inequality,

$$P\left(|\bar{\beta}_{jk} - p(\omega_{jk})| > \frac{\epsilon_1}{3} p(\omega_{jk})\right) \leq \frac{9}{\epsilon_1^2 s} \frac{\mathbb{V}\text{ar}_p(X_{jk})}{p(\omega_{jk})^2}. \tag{20}$$

Now if $|\bar{\beta}_{jk} - p(\omega_{jk})| \leq \frac{\epsilon_1}{3} p(\omega_{jk})$, from (19),

$$\begin{aligned} |\bar{\beta}_{jk} - \pi(\omega_{jk})| &\leq |\bar{\beta}_{jk} - p(\omega_{jk})| + |p(\omega_{jk} - \pi(\omega_{jk}| \\ &\leq \frac{\epsilon_1}{3} p(\omega_{jk}) + \frac{\epsilon_1}{8}\pi(\omega_{jk}) \leq \frac{\epsilon_1}{2}\pi(\omega_{jk}) \end{aligned}$$

and $\bar{\beta}_{jk}$ estimates $\pi(\omega_{jk})$ within ratio $1+\epsilon_1$. Since $\epsilon_1 < 1$ and $\mathbb{V}\mathrm{ar}_\pi(X_{jk}) \le \pi(\omega_{jk})$,

$$\frac{\mathbb{V}\mathrm{ar}_p(X_{jk})}{p(\omega_{jk})^2} \le \frac{\mathbb{V}\mathrm{ar}_\pi(X_{jk}) + \frac{3}{8}\pi(\omega_{jk})}{\left(\frac{7}{8}\pi(\omega_{jk})\right)^2} \le \frac{2}{\pi(\omega_{jk})}. \tag{21}$$

Hence, by choosing $s = 72\epsilon_1^{-2}\epsilon_2^{-1}$ and using (20) and (21), $P\left(|\bar{\beta}_{jk} - p(\omega_{jk})| > \frac{\epsilon_1}{3}p(\omega_{jk})\right) \le \frac{1}{4}$, that is, $\bar{\beta}_{jk}$ estimates $\pi(\omega_{jk})$ within ratio $1+\epsilon_1$ with probability at least $3/4$.

Now consider repeating the above experiment by an odd number t times, independently. Let $\hat{\beta}_{jk}$ be the median of the resulting t values of $\bar{\beta}_{jk}$. From above, the probability that $\hat{\beta}_{jk}$ fails to approximate β_{jk} within ratio $1+\epsilon_1$ is at most

$$\begin{aligned} \sum_{i=(t+1)/2}^{t} \binom{t}{i} \left(\frac{1}{4}\right)^{i} \left(\frac{3}{4}\right)^{t-i} &\le \left(\frac{1}{4}\right)^{t/2} \left(\frac{3}{4}\right)^{t/2} \sum_{i=(t+1)/2}^{t} \binom{t}{i} \\ &\le \left(\frac{3}{16}\right)^{t/2} 2^t = \left(\frac{3}{4}\right)^{t/2}. \end{aligned}$$

Now let $t = 6\lceil \log \eta^{-1} \rceil + 1$, this probability is bounded above by η. Hence, with a total of st samples, $\hat{\beta}_{jk}$ estimates $\pi(\omega_{jk})$ within ratio $1+\epsilon_1$ with probability at least $1-\eta$ for $\beta_{jk} \ge \epsilon_2$. Notice that st is upper bounded by $504\epsilon_1^{-2}\epsilon_2^{-1}\lceil \log \eta^{-1} \rceil$.

Now consider β_{jk} that are smaller than ϵ_2. With probability at least $1-\eta$, for $(j,k) \in \beta_{\epsilon_2}$, $(1-\epsilon_1)\beta_{jk} \le \hat{\beta}_{jk} \le (1+\epsilon_1)\beta_{jk}$. So if $\hat{\beta}_{jk} \ge (1+\epsilon_1)\epsilon_2$, we must have $(j,k) \in \beta_{\epsilon_2}$. Hence, $\hat{\beta}_{jk} \le (1+\epsilon_1)\epsilon_2$ or $|\hat{\beta}_{jk} - \beta_{jk}| \le (1+\epsilon_1)\epsilon_2$ for $\beta_{jk} < \epsilon_2$.

REFERENCES

[1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. San Diego, CA: Academic Press, 1988.

[2] I. Cox, "A review of statistical data association techniques for motion correspondence," *International Journal of Computer Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[3] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "EM, MCMC, and chain flipping for structure from motion with unknown correspondence," *Machine Learning*, vol. 50, pp. 45–71, 2003.

[4] R. Sittler, "An optimal data association problem on surveillance theory," *IEEE Trans. on Military Electronics*, vol. MIL-8, pp. 125–139, April 1964.

[5] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transaction on Automatic Control*, vol. 24, no. 6, pp. 843–854, December 1979.

[6] A. Poore, "Multidimensional assignment and multitarget tracking," *Partitioning Data Sets. DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 19, pp. 169–196, 1995.

[7] T. Kurien, "Issues in the design of practical multitarget tracking algorithms," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Artech House, Norwood, MA, 1990.

[8] I. Cox and S. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.

[9] C. L. Morefield, “Application of 0-1 integer programming to multitarget tracking problems,” *IEEE Trans. on Automatic Control*, vol. 22(3), pp. 302–312, June 1971.

[10] J. Collins and J. Uhlmann, “Efficient gating in data association with multivariate distributed states,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 909–916, July 1992.

[11] L. Valiant, “The complexity of computing the permanent,” *Theoretical Computer Science*, vol. 8, pp. 189–201, 1979.

[12] R. Fitzgerald, “Development of practical PDA logic for multipltarget tracking by microprocessor,” in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Artech House: Norwood, MA, 1990.

[13] J. Roecker and G. Phillis, “Suboptimal joint probabilistic data association,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-29, 2, pp. 510–517, April 1993.

[14] J. Roecker, “A class of near optimal JPDA algorithms,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-30, 2, pp. 504–510, April 1994.

[15] T. Huang and S. J. Russell, “Object identification in a Bayesian context,” in *Proc. of the International Joint Conference on Artificial Intelligence*, Nagoya, Japan, Aug. 1997.

[16] H. Pasula, S. J. Russell, M. Ostland, and Y. Ritov, “Tracking many objects with many sensors,” in *Proc. of the International Joint Conference on Artificial Intelligence*, Stockholm, 1999.

[17] H. Pasula, *Identity Uncertainty*. Ph.D. Thesis, Computer Science Division: Univ. of California, Berkeley, CA, 2003.

[18] S. Cong, L. Hong, and D. Wicker, “Markov-chain Monte-Carlo approach for association probability evaluation,” *IEE Proceedings of Control, Theory and Applications*, vol. 151, no. 2, pp. 185–193, March 2004.

[19] N. Bergman and A. Doucet, “Markov chain Monte Carlo data association for target tracking,” in *Proc. of IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, June 2000.

[20] M. Jerrum and A. Sinclair, “The Markov chain Monte Carlo method: An approach to approximate counting and integration,” in *Approximations for NP-hard Problems*, D. Hochbaum, Ed. PWS Publishing, Boston, MA, 1996.

[21] I. Beichl and F. Sullivan, “The Metropolis algorithm,” *Computing in Science and Engineering*, vol. 2, no. 1, pp. 65–69, 2000.

[22] G. Roberts, “Markov chain concepts related to sampling algorithms,” in *Markov Chain Monte Carlo in Practice*, ser. Interdisciplinary Statistics Series, W. Gilks, S. Richardson, and D. Spiegelhalter, Eds. Chapman and Hall, 1996.

[23] P. Diaconis and D. Stroock, “Geometric bounds for eigenvalues of Markov chains,” *Annals of Applied Probability*, vol. 1, pp. 36–61, 1991.

[24] D. Schulz, W. Burgard, D. Fox, and A. Cremers, “Tracking multiple moving targets with a mobile robot using particle filters and statistical data association,” in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2001.

[25] S. Oh and S. Sastry, “A polynomial-time approximation algorithm for joint probabilistic data association,” in *Proc. of the American Control Conference*, Portland, OR, June 2005.

[26] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*, ser. Interdisciplinary Statistics Series. Chapman and Hall, 1996.

[27] S. Oh, S. Russell, and S. Sastry, “Markov chain Monte Carlo data association for general multiple-target tracking problems,” in *Proc. of the 43rd IEEE Conference on Decision and Control*, Paradise Island, Bahamas, Dec. 2004.

[28] I. Cox, “Multiple hypothesis tracking code,” http://www.ee.ucl.ac.uk/ ˜icox/.