Bayesian Formulation of Data Association and Markov Chain Monte Carlo Data Association

Songhwai Oh

Electrical Engineering and Computer Science
University of California at Merced
Merced, California 95343
Email: sho@ucmerced.edu

***Abstract*— This paper presents the Bayesian formulation of data association and reviews an approximation algorithm called Markov chain Monte Carlo data association (MCMCDA) for solving data association problems arising in multi-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). While the exact computation of association probabilities in JPDA is NP-hard, single-scan MCMCDA algorithm provides a fully polynomial randomized approximation scheme for JPDA. For general multi-target tracking problems, in which unknown numbers of targets appear and disappear at random times, a multi-scan MCMCDA algorithm approximates the optimal Bayesian filter and outperforms multiple hypothesis tracking (MHT).**

I. Introduction

Multi-target tracking is a general mathematical formulation of dynamical data association problems [1]. This paper provides the Bayesian formulation of multi-target tracking and reviews recently developed Markov chain Monte Carlo data association (MCMCDA) for solving data association problems arising in multi-target tracking in a cluttered environment [2]–[4].

The essence of the multi-target tracking problem is to find tracks from the noisy measurements. If the sequence of measurements associated with each target is known, multi-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 [5]. In many practical cases, uncertainty as to the correct association is unavoidable.

There are two well-known algorithms for solving the multi-target tracking problems and they are joint probabilistic data association (JPDA) [1] and multiple hypothesis tracking (MHT) [6]. The data association problem is known to be NP-hard [7] and we do not expect to find efficient, exact algorithms. 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 enumerating over all possible associations. Single-scan MCMCDA is a fully polynomial randomized approximation scheme for JPDA while multi-scan MCMCDA outperforms MHT [4]. MCMCDA has been successfully applied to wireless sensor networks [8], distributed identity management [9], and computer vision [10], to name a few.

The remainder of this paper is structured as follows. The Bayesian formulation of data association for multi-target tracking is 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. The multi-scan MCMCDA algorithm is described in Section V. In Section V, we give simulation results and a demonstration of multi-target tracking using wireless sensor networks.

II. Bayesian Formulation of Data Association

A. Problem Formulation

Let $T \in \mathbb{Z}^+$ be the duration of surveillance. Let K be the (unknown) number of objects that appear in the surveillance region $\mathcal{R}$ during the surveillance period. Each object k moves in $\mathcal{R}$ for some unknown duration $[t_{\mathrm{i}}^k, t_{\mathrm{f}}^k] \subseteq [1, T]$. 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 disappears 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 noisy observation (or measurement) of the state of the object is measured with a detection probability p_{d}. 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 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 an 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:

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

where $v_t^j \in \mathbb{R}^{n_y}$ are white noise processes and $u_t \sim \text{Unif}(\mathcal{R})$ is a random process for false alarms.

The multi-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. Probability Model

In order to define the prior model for data association independently from measurements, we first define $\{\omega\}$ for all possible measurement sizes. Let μ be a nonnegative T-dimensional vector, i.e., $\mu = [\mu_1, \ldots, \mu_T]^{\mathrm{T}}$, representing the possible numbers of measurements from $t = 1$ to $t = T$, where $\mu_t \in \mathbb{Z}^+ \cup \{0\}$. For each value of μ, define a set of measurement indices $\Upsilon_t^\mu = \{(t,1), (t,2), \ldots, (t, \mu_t)\}$ for $\mu_t > 0$, where (t,i) is an index to the i^{th} measurement at time t, and $\Upsilon_t^\mu = \emptyset$ for $\mu_t = 0$. Now let $\Upsilon^\mu = \cup_{t=1}^T \Upsilon_t^\mu$ be an index set to a set of measurements whose size matches μ and let the set $\{\Upsilon^\mu : \mu \in \mathbb{Z}^T\}$ contain all possible index sets.

For each μ, let Ω^μ be a collection of partitions of Υ^μ such that, for $\omega \in \Omega^\mu$, $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$, where τ_0 is a set of indices to false alarms and τ_k is a set of indices to measurements from target k, for $k = 1, \ldots, K$. More formally, $\omega \in \Omega^\mu$ is defined as follows:

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

Here, $K = K(\omega)$ is the number of tracks for the given partition $\omega \in \Omega^\mu$ and $|S|$ denotes the cardinality of the set S. We call τ_k a track when there is no confusion, although the actual track is a sequence of state estimates computed from the observations indexed by τ_k. (We assume there is a deterministic function that returns a sequence 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 relax this requirement to allow multiple observations per track. For special cases in which $p_{\mathrm{d}} = 1$ or $\lambda_{\mathrm{f}} = 0$, the definition of Ω^μ can be adjusted accordingly.

Now let $\widetilde{\Omega} = \{\omega \in \Omega^\mu : \mu \in \mathbb{Z}^T\}$. Notice that $\mu = \mu(\omega)$ is a deterministic function of $\omega \in \widetilde{\Omega}$. In addition, we can compute the following numbers from $\omega \in \widetilde{\Omega}$:

- e_t, the number of targets present at time t with $e_0 = 0$;
- z_t, the number of targets terminated at time t with $z_1 = 0$;
- a_t, the number of new targets at time t;
- d_t, the number of detected targets at time t; and
- f_t, the number of false alarms at time t, $f_t = \mu_t - d_t$.

Since these numbers are deterministic functions of $\omega \in \widetilde{\Omega}$, we have $P(\omega) = P(\omega, \mathcal{N}) = P(\omega|\mathcal{N})P(\mathcal{N})$, where $\mathcal{N} = \{\mu_t, e_t, z_t, a_t, d_t : 1 \leq t \leq T\}$. Based on the target termination, target detection, new target arrival, and false alarm models described in Section II-A, we can show that

$$\begin{aligned} P(\mathcal{N}) &= \prod_{t=1}^T \Bigg[\binom{e_{t-1}}{z_t} p_{\mathrm{z}}^{z_t} (1-p_{\mathrm{z}})^{e_{t-1}-z_t} \\ &\times \binom{e_{t-1}-z_t+a_t}{d_t} p_{\mathrm{d}}^{d_t} (1-p_{\mathrm{d}})^{e_{t-1}-z_t+a_t-d_t} \\ &\times \frac{(\lambda_{\mathrm{b}}V)^{a_t}}{a_t!} \exp(-\lambda_{\mathrm{b}}V) \frac{(\lambda_{\mathrm{f}}V)^{f_t}}{f_t!} \exp(-\lambda_{\mathrm{f}}V) \Bigg]. \end{aligned} \tag{3}$$

Since $\omega \in \widetilde{\Omega}$ with the same $\mathcal{N}$ are indistinguishable, i.e., invariant under permutation of target indices, they are exchangeable and we assign a uniform prior on them. Hence,

$$P(\omega|\mathcal{N}) \propto \textstyle\prod_{t=1}^T \left[\binom{e_{t-1}}{z_t} \binom{e_{t-1}-z_t+a_t}{d_t} \binom{\mu_t}{d_t} \binom{d_t}{a_t} (d_t - a_t)! \right]^{-1}. \tag{4}$$

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 $t = T$. Y_t can be considered as a vector with random ordering as indicated by the exchangeability of indices in (4). Combining (3) and (4), it can be shown that the posterior of $\omega \in \widetilde{\Omega}$ becomes:

$$\begin{aligned} &P(\omega|Y) \propto P(Y|\omega)P(\omega) \\ &\quad \propto P(Y|\omega) \\ &\quad \times \textstyle\prod_{t=1}^{\mathrm{T}} \frac{1}{\mu_t!} p_{\mathrm{z}}^{z_t} (1-p_{\mathrm{z}})^{c_t} p_{\mathrm{d}}^{d_t} (1-p_{\mathrm{d}})^{g_t} (\lambda_{\mathrm{b}}V)^{a_t} (\lambda_{\mathrm{f}}V)^{f_t}, \end{aligned} \tag{5}$$

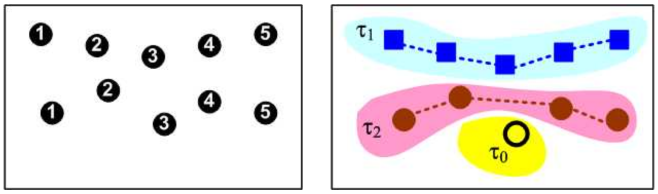
where $P(Y|\omega)$ is the likelihood of observations Y given $\omega \in \widetilde{\Omega}$.

It is important to notice that $P(Y|\omega) = 0$ if $\mu(\omega) \neq n(Y)$, where $n(Y) = [n_1(Y), \ldots, n_T(Y)]^{\mathrm{T}}$ denotes the number of measurements at each time in Y. Hence, we can restrict our attention to those $\omega \in \widetilde{\Omega}$ with $\mu(\omega) = n(Y)$. This crucial observation makes the numerous computations based on (5) practical. The set of all possible associations is now defined as $\Omega := \Omega^{n(Y)} = \{\omega \in \widetilde{\Omega} : \mu(\omega) = n(Y)\}$ and Ω is used instead of $\widetilde{\Omega}$ throughout this paper. Thus, it is convenient to view Ω as a collection of partitions of Y. An example of one such partition is shown in Figure 1.

The posterior (5) can be further simplified as

$$\begin{aligned} &P(\omega|Y) \propto P(Y|\omega) \\ &\quad \times \textstyle\prod_{t=1}^{\mathrm{T}} p_{\mathrm{z}}^{z_t} (1-p_{\mathrm{z}})^{c_t} p_{\mathrm{d}}^{d_t} (1-p_{\mathrm{d}})^{g_t} (\lambda_{\mathrm{b}}V)^{a_t} (\lambda_{\mathrm{f}}V)^{f_t}, \end{aligned} \tag{6}$$

where the term $\prod_{t=1}^T V^{a_t + f_t}$ will be canceled out by the matching initial state and false alarm densities in $P(Y|\omega)$. The likelihood $P(Y|\omega)$ can be computed based on the chosen dynamic and measurement models (for a solid example, see [2]). The posterior $P(\omega|Y)$ can be applied to both MAP and



(a) (b)

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.

Bayes estimator approaches to solve the multi-target tracking problem.

III. Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) plays a significant role in many fields such as physics, statistics, economics, finance, and engineering [11]–[13]. The MCMC method includes algorithms such as Gibbs sampling [14] and the Metropolis-Hastings algorithm [15], [16]. Beichl and Sullivan described the Metropolis-Hastings algorithm as "the most successful and influential of all the members of ... the *Monte Carlo Method*" [12].

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 $\mathcal{M}$ is at state $\omega \in \Omega$, $\omega' \in \Omega$ is proposed following the proposal distribution $q(\omega, \omega')$. The move is accepted with an acceptance probability $A(\omega, \omega')$ where

$$A(\omega, \omega') = \min\left(1, \frac{\pi(\omega')q(\omega', \omega)}{\pi(\omega)q(\omega, \omega')}\right), \tag{7}$$

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

$$Q(\omega, \omega') := \pi(\omega)P(\omega, \omega') = \pi(\omega')P(\omega', \omega), \tag{8}$$

where $P(\omega, \omega') = q(\omega, \omega')A(\omega, \omega')$ is the transition probability from ω to ω'. Hence, $\mathcal{M}$ is a reversible Markov chain.

If $\mathcal{M}$ is also irreducible and aperiodic, then $\mathcal{M}$ converges to its stationary distribution by the ergodic theorem [17]. Hence, for any bounded function f, the sample mean $\hat{f} = \frac{1}{N}\sum_{n=1}^{N} f(\omega^{(n)})$ converges to $\mathbb{E}_\pi f(\omega)$ as $N \to \infty$, where $\omega^{(n)}$ is the state of $\mathcal{M}$ at the n^{th} MCMC step and $\mathbb{E}_\pi f(\omega)$ is the expected value of $f(\omega)$ with respect to measure π. Notice that (7) requires only the ability to compute the ratio $\pi(\omega')/\pi(\omega)$, avoiding the need to normalize π, and this is why MCMC, especially the Metropolis-Hastings algorithm, can be applied to a wide range of applications.

IV. Single-Scan MCMCDA

The main result for single-scan MCMCDA is its theoretical analysis. The data association problem is known to be NP-hard [7], [18] and we do not expect to find efficient, exact algorithms. MCMCDA is the first data association algorithm with provable guaranteed error bounds, i.e., MCMCDA is a fully polynomial randomized approximation scheme for JPDA [4]. More specifically, for any $\epsilon > 0$ and any $0 < \eta < 0.5$, the algorithm finds ϵ-*good estimates* with probability at least $1-\eta$ in time complexity $O(\epsilon^{-2} \log \eta^{-1} N(N \log N + \log(\epsilon^{-1})))$, i.e., polynomial in number of measurements, where N is the number of measurements. The precise meaning of ϵ-*good estimates* and related theorems are given in [4].

The single-scan MCMCDA filter follows the same filtering steps as the assumed-density single-scan Bayesian filter [4] or JPDA [1], except the computation of association probabilities. Since the filtering steps are well known, we only describe the approximation method using MCMC.

Suppose that there are K targets and n_t measurements at time t (the number of targets are assumed to be fixed in single-scan MCMCDA). We also assume that an estimated likelihood $\hat{P}^k(y_t^j|y_{1:t-1})$ is available for each target k from the previous filtering step. $\hat{P}^k(y_t^j|y_{1:t-1})$ is a probability density of having observation y_t^j given $y_{1:t-1}$, when y_t^j is a measurement originated from target k. The goal is to estimate $\hat{P}(X_t^k|y_{1:t})$, where X_t^k is the state of k-th target at time t, from $\hat{P}^k(y_t^j|y_{1:t-1})$ and Y_t for all k.

Let ω be a feasible association between n_t measurements and K targets and we let $\Omega = \{\omega\}$ be a set of all feasible (joint) association events at time t. For each $\omega \in \Omega$, $\omega = \{(j,k)\}$, where (j,k) denotes the event that observation j is associated with target k. An association event ω is *feasible* when (i) for each $(j,k) \in \omega$, y_t^j is validated for target k (i.e., $\hat{P}^k(y_t^j|y_{1:t-1}) \geq \delta^k$ for $\delta^k > 0$); (ii) an observation is associated with at most one target; and (iii) a target is associated with at most one observation.

Let $N \leq n_t$ be the number of validated observations. We encode the feasible association events in a bipartite graph $G = (U, V, E)$, 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. A feasible 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 association events Ω can be represented as $\Omega = M_0(G) \cup \cdots \cup M_K(G)$, where $M_k(G)$ is the set of k-matchings in G.

We can compute the approximate distribution as [4]:

$$\hat{P}(X_t^k|y_{1:t}) = \sum_{j=0}^{N} \beta_{jk} \hat{P}(X_t^k|\omega_{jk}, y_{1:t}), \tag{9}$$

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

$$\beta_{jk} = \hat{P}(\omega_{jk}|y_{1:t}) = \sum_{\omega:(j,k)\in\omega} \hat{P}(\omega|y_{1:t}). \tag{10}$$

$\hat{P}(X_t^k|\omega_{jk}, y_{1:t})$ in (9) can be easily computed by considering it as a single-target estimation problem with a single ob-

Algorithm 1 Single-scan-MCMCDA

```
INPUT: $G = (U, V, E), n_{\mathrm{mc}}, n_{\mathrm{bi}}, \theta$
OUTPUT: $\{\hat{\beta}_{jk}\}$
$\hat{\beta}_{jk} = 0$ for all $j$ and $k$
choose $\omega^{(0)}$ randomly from $\Omega$
for $n = 1$ to $n_{\mathrm{mc}}$ do
    $\omega^{(n)}$ = Single-scan-MCMCDA.single-step($G, \omega^{(n-1)}, \theta$) (see Algorithm 2)
    if $n > n_{\mathrm{bi}}$ then
        for each $(y^j, k) \in \omega^{(n)}$ do
            $\hat{\beta}_{jk} = \hat{\beta}_{jk} + 1/(n_{\mathrm{mc}} - n_{\mathrm{bi}})$
        end for
    end if
end for
```

Algorithm 2 Single-scan-MCMCDA.single-step

```
INPUT: $G = (U, V, E), \omega, \theta$
OUTPUT: $\omega$
sample $Z$ from Unif$[0, 1]$
if $Z < \frac{1}{2}$ 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 $A(\omega, \omega')$
```

servation. On the other hand, the computation of β_{jk} requires a summation over exponentially many association events.

Based on the parametric false alarm model described in Section II-A, for each $\omega \in \Omega$, the prior $P(\omega)$ can be written as

$$P(\omega) \propto (\lambda_{\mathrm{f}} V)^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1 - p_{\mathrm{d}})^{K-|\omega|}. \tag{11}$$

Then, the estimated posterior of $\omega \in \Omega$ can be written as

$$\hat{P}(\omega|y_{1:t}) = \frac{1}{Z} \lambda_{\mathrm{f}}^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1 - p_{\mathrm{d}})^{K-|\omega|} \prod_{(u,v)\in\omega} \hat{P}^v(u|y_{1:t-1}) \tag{12}$$

where Z is a normalizing constant.

The MCMC data association (MCMCDA) algorithm is an MCMC algorithm whose state space is the set of all feasible association events Ω and whose stationary distribution is the posterior $\hat{P}(\omega|y_{1:t})$ (12). The single-scan MCMCDA algorithm is shown in Algorithm 1, where $\theta = \{\{\hat{P}^v(u|y_{1:t-1})\}, \lambda_{\mathrm{f}}, p_{\mathrm{d}}, K, N\}$, along with its MCMC step described in Algorithm 2. The inputs to Algorithm 1 are the graph G, the number of samples n_{mc}, the number of burn-in samples n_{bi}, and θ. The input θ contains likelihoods $\{\hat{P}^v(u|y_{1:t-1})\}$ and model parameters $\lambda_{\mathrm{f}}, p_{\mathrm{d}}, K$, and N. Algorithm 1 computes the approximate association probabilities $\{\hat{\beta}_{jk}\}$, which can be used in (9) to compute the approximate posterior distribution $\hat{P}(X_t^k|y_{1:t})$. 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 (12).

V. Multi-Scan MCMCDA

The single-scan MCMCDA algorithm described in Section IV assumes a fixed, known number of targets. This assumption leads to a simple filtering scheme, but in most situations of interest the number of targets is unknown and changes over time. Furthermore, a single-scan algorithm that makes approximations (such as measurement validation and independence) to avoid complexity may end up being unable to maintain tracks over long periods because it cannot revisit previous, possibly incorrect, association decisions in the light of new evidence. For these reasons, methods for solving the general multi-target tracking problem described in Section II often adopt a multi-scan design, maintaining state in the form of both the posterior approximation and the observation history. This section describes a multi-scan MCMCDA algorithm that can handle unknown numbers of targets. The solution space Ω for this algorithm contains association *histories* over multiple time steps, as well as considering all possible numbers of targets at each step, and is therefore much larger than the solution space considered by a single-scan algorithm. The multi-scan MCMCDA algorithm features efficient mechanisms to search over this large solution space in addition to birth and death moves to add or remove tracks. The multi-scan MCMCDA algorithm and its extension to an online version are presented in this section along with simulation and experimental results.

A. Multi-Scan MCMCDA Algorithm

The multi-scan MCMCDA algorithm is described in Algorithm 3. It is an MCMC algorithm whose state space is Ω described in Section II and whose stationary distribution is the posterior (6). 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 multi-scan MCMCDA moves are graphically illustrated in Figure 2. 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

Algorithm 3 Multi-scan-MCMCDA

```
INPUT: $Y, n_{\mathrm{mc}}, \omega_{\mathrm{init}}$
OUTPUT: $\hat{\omega}$
$\omega = \omega_{\mathrm{init}}$; $\hat{\omega} = \omega_{\mathrm{init}}$
for $n = 1$ to $n_{\mathrm{mc}}$ do
    propose $\omega'$ based on $\omega$ (see text)
    sample $U$ from Unif$[0,1]$
    $\omega = \omega'$ if $U < A(\omega, \omega')$
    $\hat{\omega} = \omega$ if $p(\omega|Y)/p(\hat{\omega}|Y) > 1$
end for
```

$\xi_0(m=1) = 1$ 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 measurements Y, the number of samples n_{mc}, and the initial state ω_{init}. At each step of the algorithm, ω is the current state of the Markov chain. The acceptance probability $A(\omega, \omega')$ is defined in (7) where $\pi(\omega) = P(\omega|Y)$ from (6). The output $\hat{\omega}$ approximates the MAP estimate $\arg\max P(\omega|Y)$. Given $\hat{\omega}$, The states of the targets can be easily computed by any filtering algorithm since the associations between the targets and the measurements are known. Algorithm 3 can be also used to find the Bayesian estimates of the target states (see [4] for more detail).

An MCMC algorithm can be specialized and made more efficient by incorporating the domain specific knowledge. In multi-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 measurements 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 $\bar{p}_{\mathrm{dt}}(s) = 1 - (1 - p_{\mathrm{d}})^s$ be the probability that a target is observed at least once out of s measurement times. Then, for given $\bar{p}_{\mathrm{dt}}$, we set $\bar{d} = \lceil \log(1 - \bar{p}_{\mathrm{dt}}) / \log(1 - p_{\mathrm{d}}) \rceil$ to detect a track with probability at least $\bar{p}_{\mathrm{dt}}$. For example, given $p_{\mathrm{d}} = 0.7$ and $\bar{p}_{\mathrm{dt}} = 0.99$, a track is detected with probability larger than 0.99 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, called a neighborhood tree of measurements, which groups temporally separated measurements based on distances, to propose a new partition ω' in Algorithm 3. A neighborhood tree of measurements 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 Euclidean distance. The parameter d allows missing measurements. The use of this neighborhood tree makes the algorithm more scalable since distant measurements will be considered separately and makes the computations of the proposal distribution easier. It is similar to the clustering technique used in MHT but L_d is fixed for a given set of measurements.

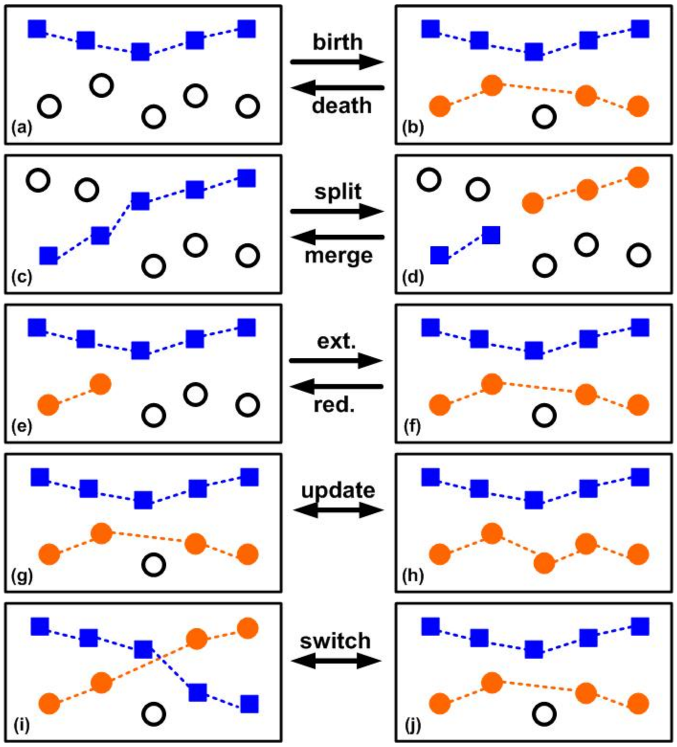


Fig. 2. Graphical illustration of MCMCDA moves (associations are indicated by dotted lines and hollow circles are false alarms). Each move proposes a new joint association event ω' which is a modification of the current joint association event ω. The birth move proposes ω' by forming a new track from the set of false alarms ($(a) \rightarrow (b)$). The death move proposes ω' by combining one of the existing tracks into the set of false alarms ($(b) \rightarrow (a)$). The split move splits a track from ω into two tracks ($(c) \rightarrow (d)$) while the merge move combines two tracks in ω into a single track ($(d) \rightarrow (c)$). The extension move extends an existing track in ω ($(e) \rightarrow (f)$) and the reduction move reduces an existing track in ω ($(f) \rightarrow (e)$). The track update move chooses a track in ω and assigns different measurements from the set of false alarms ($(g) \leftrightarrow (h)$). The track switch move chooses two track from ω and switches some measurement-to-track associations ($(i) \leftrightarrow (j)$).

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.

Birth and Death Moves (Figure 2, $(a) \leftrightarrow (b)$): 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 target. Then we choose d_1 from the distribution ζ. Let $L_{d_1}^1 = \{y_{t_1}^j : L_{d_1}(y_{t_1}^j) \neq \emptyset, y_{t_1}^j \notin \tau_k(t_1), j = 1, \ldots, n_{t_1}, k = 1, \ldots, K\}$. $L_{d_1}^1$ is a set of measurements at t_1 such that, for any $y \in L_{d_1}^1$, 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 $L_{d_1}^1$. If $L_{d_1}^1$ is empty, the move is rejected since the move is not reversible. Once the initial measurement is chosen, we then choose the subsequent measurements 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 measurements to $\tau_{K'}$ terminates with probability p_{z}. 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 $\omega'^0 = \{\tau_0 \cup \tau_k\}$.

Split and Merge Moves (Figure 2, $(c) \leftrightarrow (d)$): 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 following set of possible merge move pairs:

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

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 $\omega'^0 = \omega^0$.

Extension and Reduction Moves (Figure 2, $(e) \leftrightarrow (f)$): In a track extension move, we select a track τ u.a.r. from K available tracks in ω. We reassign measurements 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 measurements assigned to τ after the time t_{r+1}.

Track Update Move (Figure 2 $(g) \leftrightarrow (h)$): 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 measurements for τ after the time t_r as done in the track birth move.

Track Switch Move (Figure 2, $(i) \leftrightarrow (j)$): For a track switch move, we select a pair of measurements $(\tau_{k_1}(t_p), \tau_{k_2}(t_q))$ 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

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

B. Online MCMCDA

While the computational complexity of the multi-scan MCMCDA algorithm is lighter than MHT [4], it grows as more measurements are collected. Since recent measurements are more relevant to the current states, good estimates of the current states can still be found from recent measurements. Based on this idea, we propose an online multi-scan MCMCDA algorithm whose estimates are based on measurements from a window of time $[t_{\text{curr}} - t_{\text{win}} + 1, \ldots, t_{\text{curr}}]$, where t_{curr} is the current time and t_{win} is the size of a window. Hence, at all times, only a finite number of measurements are kept by the algorithm. This online implementation of multi-scan MCMCDA is suboptimal because it considers only a subset of past measurements. At each time step, we use the previous estimate to initialize multi-scan MCMCDA and run multi-scan MCMCDA on the measurements belonging to the current window. The measurements belonging to the current window are $Y_{\text{w}} = \{y^j(t) : 1 \leq j \leq n(t), t_{\text{curr}} - t_{\text{win}} + 1 \leq t \leq t_{\text{curr}}\}$. At time $t_{\text{curr}} + 1$, the measurements at time $t_{\text{curr}} - t_{\text{win}} + 1$ are removed from Y_{w} and a new set of measurements is appended to Y_{w}. Any delayed measurements are inserted into the appropriate slots. Then, we initialized the Markov chain with the previously estimated tracks and executes Algorithm 3 on Y_{w}.

C. Simulation Results

An example of tracking multiple targets in a densely cluttered environment is used to demonstrate online-version of multi-scane MCMCDA. For more extensive comparison of MCMCDA against MHT and multi-scan NNF, see [4].

For this example, the surveillance duration is $T = 100$ and the scenario is generated according to the model for multi-target tracking described in Section II-A. The surveillance region is $\mathcal{R} = [0, 100] \times [0, 100]$ and the model parameters are: $\lambda_{\text{b}} V = 5$, $p_{\text{z}} = 1/20$, $p_{\text{d}} = 0.7$, and $\lambda_{\text{f}} V = 30$. There are a total of 380 targets. The following linear model is used:

$$x_{t+1}^k = A x_t^k + G w_t^k \qquad\qquad y_t^j = C x_t^k + v_t^j \tag{13}$$

where

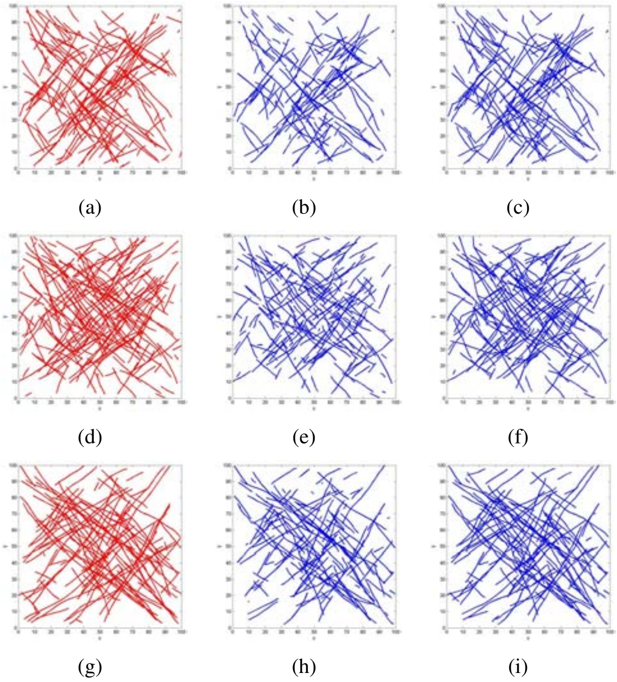
$$A = \begin{bmatrix} 1 & 0 & T_{\text{s}} & 0 \\ 0 & 1 & 0 & T_{\text{s}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

and T_{s} is the sampling period, w_t^k is a zero-mean Gaussian process with covariance $Q = \text{diag}(0.031, 0.031)$, and v_t^j is a zero-mean Gaussian process with covariance $R = \text{diag}(0.031, 0.031)$. The size of the sliding window is $t_{\text{win}} = 14$ for the online MCMCDA algorithm while $\bar{d} = 5$ and $\bar{v} = 3$ unit lengths per unit time.

The C++ implementation of MHT [19] is used for comparison, 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 to that of MCMCDA when there are 10 targets: the maximum number of hypotheses in a group is 1,000, the maximum track tree depth is 5, and the maximum Mahalanobis distance is 11.8. All simulations are run on a PC with a 2.6-GHz Intel processor.

For this example, MHT took 6,995 seconds while online MCMCDA took only 343 seconds, i.e., a 20-fold reduction in computation time. On the F_1 measure, MHT scored 0.85 and MCMCDA scored 0.91. The F_1 measure is a harmonic mean



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

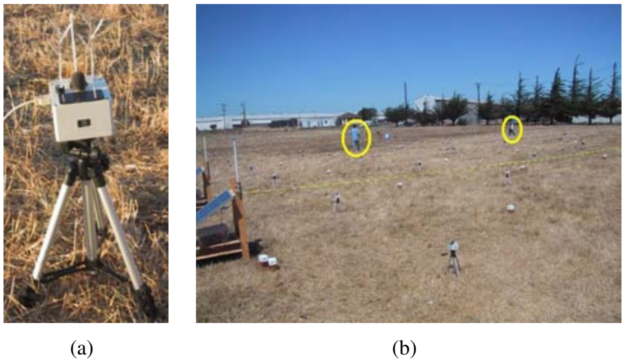
Fig. 3. Actual trajectories of targets at different time intervals and tracks estimated by MHT and MCMCDA. The running time of MCMCDA for this example was 343 seconds while it took 6,995 seconds for MHT to complete. MCMCDA scored 0.91 in the F_1 measure and MHT scored 0.85. (a) Actual trajectories of targets from $t = 10$ to $t = 40$. (b) Tracks estimated by MHT from $t = 10$ to $t = 40$. (c) Tracks estimated by MCMCDA from $t = 10$ to $t = 40$. (d) Actual trajectories of targets from $t = 40$ to $t = 70$. (e) Tracks estimated by MHT from $t = 40$ to $t = 70$. (f) Tracks estimated by MCMCDA from $t = 40$ to $t = 70$. (g) Actual trajectories of targets from $t = 70$ to $t = 100$. (h) Tracks estimated by MHT from $t = 70$ to $t = 100$. (i) Tracks estimated by MHT from $t = 70$ to $t = 100$.

between recall and precision with an equal weight. The higher the value of the F_1 measure, the more effective the algorithm is. In addition, MHT found 494 targets but MCMCDA detected 335 targets which is close to the actual number of 380 targets. The tracks estimated by MHT and MCMCDA are shown in Figure 3. For easy comparison, Figure 3 also shows the actual trajectories of targets. In summary, this example shows that MCMCDA is very effective in a dense environment and achieves superior performance with a fraction of the computation time required by MHT.

D. Experiments

Multi-target tracking and a pursuit evasion game using MCMCDA were demonstrated at the Defense Advanced Research Projects Agency (DARPA) Network Embedded Systems Technology (NEST) final experiment on August 30, 2005. In this section, we describe experimental results reported in [8].

The experiment was performed on a large-scale, long-term, outdoor sensor network testbed deployed on a short grass field at U.C. Berkeley's Richmond Field Station (see Figure 4). A total of 557 sensor nodes were deployed and 144 of these nodes were allotted for the tracking and PEG experiments. Each sensor node was elevated using a camera tripod to prevent the passive infrared (PIR) sensors from being obstructed by grass and uneven terrain (see Figure 4(a)). The locations of the nodes were measured during deployment using differential GPS and stored in a table at the base station for reference. However, in the experiments the system assumed the nodes were placed exactly on a 5 meter spacing grid to highlight the robustness of the system against localization error. For more detail about the experiment setup and description about the system used in the experiments, see [8].



(a) (b)

Fig. 4. Hardware for the sensor nodes. (a) Trio sensor node on a tripod. On top is the microphone, buzzer, solar panel, and user and reset buttons. On the sides are the windows for the passive infrared sensors. (b) A live picture from the 2 target PEG experiment. The targets are circled.

The online multi-scan MCMCDA algorithm was demonstrated on one, two, and three human targets, with targets entering the field at different times. In all three experiments, the tracking algorithm correctly estimated the number of targets and produced correct tracks. Furthermore, the algorithm correctly disambiguated crossing targets in the two and three target experiments without classification labels on the targets, using the dynamic models and target trajectories before crossing to compute the tracks.

Figure 5 shows the multi-target tracking results with three targets walking through the field. The three targets entered and exited the field around time 10 and 80, respectively. During the experiment, the algorithm correctly rejected false alarms and compensated for missing detections. There were many false alarms during the span of the experiments. Though not shown in the figures, the algorithm dynamically corrected previous track hypotheses as it received more sensor readings.

In another demonstration, two simulated pursuers were dispatched to chase two crossing human targets. The pursuer-to-target assignment and the robust minimum time-to-capture control law were computed in real-time, in tandem with the real-time tracking of the targets. The simulated pursuers captured the human targets, as shown in Figure 6. In particular, note that the MTT module is able to correctly disambiguate the presence of two targets (right panel of Figure 6(a)) using past measurements, despite the fact that the MSF module reports the detection of a single target (upper left panel of Figure 6(a)). A live picture of this experiment is shown on the right of Figure 4.

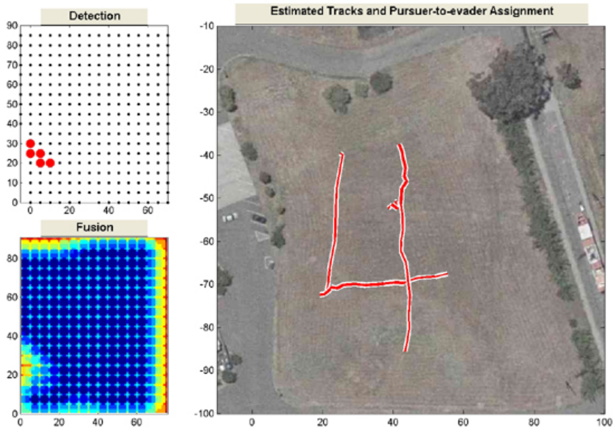
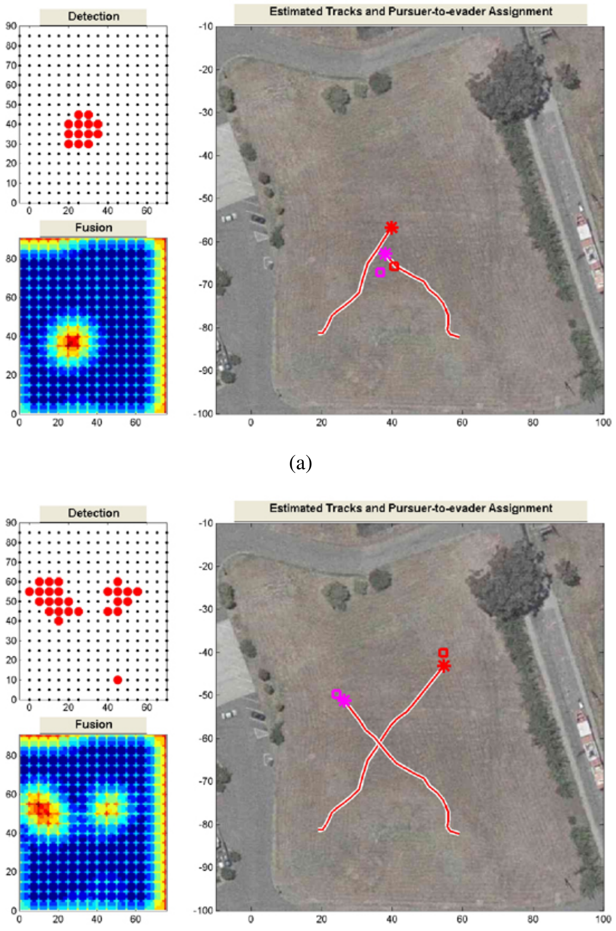


Fig. 5. Estimated tracks of targets at time 70 from the experiment with three people walking in the field. (upper left) Detection panel. Sensors are marked by small dots and detections are shown in large disks. (lower left) Fusion panel shows the fused likelihood. (right) Estimated Tracks and Pursuer-to-evader Assignment panel shows the tracks estimated by the MTT module, estimated evader positions (stars) and pursuer positions (squares).



(a)

(b)

Fig. 6. Estimated tracks of evaders and pursuer positions from the pursuit evasion game experiment. (a) Before crossing. (b) After crossing.

VI. Conclusions

In this paper, we have described the Bayesian formulation of data association and Markov chain Monte Carlo data association (MCMCDA) for solving data association problems arising in multi-target tracking in a cluttered environment. In both simulations and experiments, MCMCDA is shown to be computationally efficient approach to solve a large-scale data association problem. We plan to apply MCMCDA to more general data association problems by extending the state space of targets to include information such as features, texture, and shape.

References

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

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

[3] 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.

[4] S. Oh, S. Russell, and S. Sastry, “Markov chain Monte Carlo data association for multi-target tracking,” *IEEE Trans. Automatic Control (accepted)*, 2008.

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

[6] D. Reid, “An algorithm for tracking multiple targets,” *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843–854, December 1979.

[7] A. Poore, “Multidimensional assignment and multitarget tracking,” in *Partitioning Data Sets*, I. J. Cox, P. Hansen, and B. Julesz, Eds. American Mathematical Society, 1995, pp. 169–196.

[8] S. Oh, L. Schenato, P. Chen, and S. Sastry, “Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments,” *Proceedings of the IEEE*, vol. 95, no. 1, pp. 234–254, January 2007.

[9] S. Oh, I. Hwang, and S. Sastry, “Distributed multi-target tracking and identity management,” *Journal of Guidance, Control, and Dynamics (to appear)*, 2007.

[10] M. Meingast, S. Oh, and S. Sastry, “Automatic camera network localization using object image tracks,” in *Proc. of the IEEE International Conference on Computer Vision Workshop on Visual Representations and Modeling of Large-scale environments*, Rio de Janeiro, Brazil, Oct. 2007.

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

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

[13] B. Eraker, “MCMC analysis of diffusion models with application to finance,” *J. Bus. Econom. Statist.*, vol. 19, no. 2, pp. 177–191, 2001.

[14] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[15] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, “Equation of state calculation by fast computing machines,” *Journal of Chemical Physics*, vol. 21, pp. 1087–1092, 1953.

[16] W. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” *Biometrika*, vol. 57, pp. 97–109, 1970.

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

[18] 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.

[19] I. Cox, “Multiple hypothesis tracking code.” [Online]. Available: http://www.ee.ucl.ac.uk/∼icox/