MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets

Zia Khan, Tucker Balch, *Member*, *IEEE*, and Frank Dellaert, *Member*, *IEEE*

Abstract—We describe a particle filter that effectively deals with *interacting* targets—targets that are influenced by the proximity and/or behavior of other targets. The particle filter includes a Markov random field (MRF) motion prior that helps maintain the identity of targets throughout an interaction, significantly reducing tracker failures. We show that this MRF prior can be easily implemented by including an additional interaction factor in the importance weights of the particle filter. However, the computational requirements of the resulting multitarget filter render it unusable for large numbers of targets. Consequently, we replace the traditional importance sampling step in the particle filter with a novel Markov chain Monte Carlo (MCMC) sampling step to obtain a more efficient MCMC-based multitarget filter. We also show how to extend this MCMC-based filter to address a variable number of interacting targets. Finally, we present both qualitative and quantitative experimental results, demonstrating that the resulting particle filters deal efficiently and effectively with complicated target interactions.

Index Terms—Particle filters, multitarget tracking, Markov random fields, Markov chain Monte Carlo.

1 Introduction

This work is concerned with the problem of tracking a variable number of interacting targets. Our objective is to correctly detect entering and leaving targets and obtain a record of the trajectories of targets over time, maintaining a correct, unique identification for each target throughout. Tracking multiple targets which are identical in appearance becomes significantly more challenging when the targets *interact*. Methods that appropriately deal with this difficult issue are useful for applications where many interacting targets need to be tracked over time. In particular, they have important implications for vision-based tracking of animals, which has countless applications in biology and medicine. While more generally applicable, the results in this paper specifically center on tracking multiple interacting insects in video, which we pursue as part of a larger research project to analyze multiagent system behavior [1]. This domain offers many challenges that are quite different from the domains in which most multitarget tracking algorithms are evaluated.

The multitarget tracking literature contains a number of classical approaches to the problem of data-association, most notably the multiple hypothesis tracker (MHT) [2] and the joint probabilistic data association filter (JPDAF) [3], [4]. These multitarget tracking algorithms have been used extensively in the context of computer vision, e.g., nearest neighbor tracking in [5], the MHT in [6], and the JPDAF in [7] and [8]. However, these data association methods typically do not include a model of interaction.

In this paper, we address the problem of interacting targets, an issue not handled by traditional methods. Our approach relies on the use of a motion model that is able to adequately describe target behavior throughout an interaction event. As an example, consider the setup in Fig. 1, which shows 20 insects (ants) being tracked in a small enclosed area. In this case, the targets do *not* behave independently: Whenever one ant encounters another, some amount of interaction takes place and the behavior of a given ant before and after an interaction can be quite different. We propose to model some of this additional complexity of target behavior using a more capable motion model. While we do not aim to create a fully accurate behavioral model, a difficult proposition at best, we have found that even a small amount of domain knowledge can significantly improve tracking performance.

Our approach is based on the well-known particle filter [9], [10], [11], [12]. Particle filters offer a degree of robustness to unpredictable motion and can correctly handle complicated, nonlinear measurement models. In computer vision applications, it is often the target *appearance* which is hard to model and, hence, inference in closed form is difficult. Additionally, in animal tracking, the unpredictable motion of targets is another reason to use particle filters. As we briefly review in Section 2.1, a particle filter uses a Monte Carlo approximation to the optimal Bayes filtering distribution, which captures the knowledge about the location of all targets given all observations. The standard particle filter weights particles based on a likelihood score and then propagates these weighted particles according to a motion model.

When dealing with interacting targets, simply running one individual particle filter for each target is not a viable option. As we argue below, this does not address the complex interactions between targets and leads to frequent tracker failures. Whenever targets pass close to one another, the target with the best likelihood score typically "hijacks" the filters of nearby targets. This is illustrated in Fig. 2.

In a first contribution of this paper, we show how a Markov random field (MRF) motion prior, built dynamically at each time step, can cope with the "hijacking" problem by adequately modeling target interactions and maintaining identity as a result. This model is discussed in detail below in Section 3, where we show that incorporating an MRF to model interactions is equivalent to adding an additional interaction factor to the importance weights in a multitarget or *joint*

• *The authors are with the GVU Center, College of Computing, Georgia Institute of Technology, 85 5th Street NW, Atlanta, GA 30332-0760. E-mail: {zkhan, tucker, dellaert}@cc.gatech.edu.*

Manuscript received 12 July 2004; revised 4 Mar. 2005; accepted 7 Mar. 2005; published online 14 Sept. 2005.
Recommended for acceptance by R. Chellappa.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0347-0704.

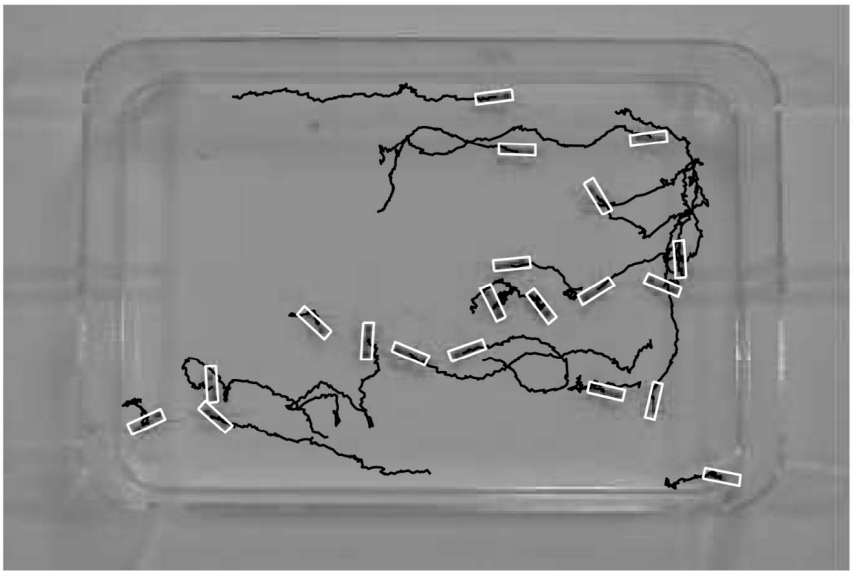


Fig. 1. Twenty ants (*Aphaenogaster cockerelli*) in a closed arena are being tracked by an MCMC-based particle filter. Targets do *not* behave independently: Whenever one ant encounters another, some amount of interaction takes place and the behavior of a given ant before and after an interaction can be quite different. This observation is generally applicable to any situation where many interacting targets need to be tracked over time.

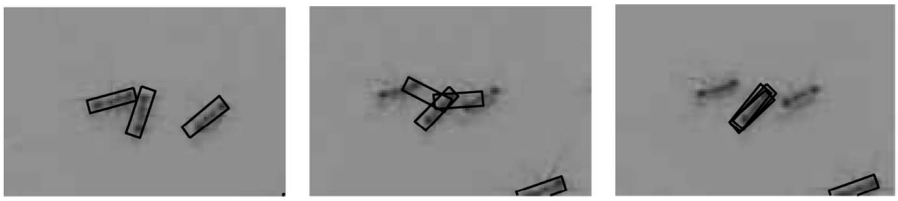


Fig. 2. (a) Frame 9043. Three interacting ants tracked using independent particle filters. (b) Frame 9080. The target with the best likelihood score typically "hijacks" the filters of nearby targets. (c) Frame 9083. Resulting tracker failure. We address this problem using a Markov random field motion prior, built dynamically at each time step that can adequately model interactions to overcome these failure modes.

particle filter. However, joint particle filters suffer from exponential complexity in the number of tracked targets, the associated computational requirements render the joint filter unusable for more than three or four targets [13].

As a second contribution, we show how we can address the exponential complexity induced by the MRF formulation using Markov chain Monte Carlo (MCMC) sampling. In our solution, we replace the traditional importance sampling step in the joint particle filter with a novel and efficient MCMC sampling step. This approach has the appealing property that *the filter behaves as a set of individual particle filters when the targets are not interacting, but efficiently deals with complicated interactions as targets approach each other*. The details of this MCMC-based particle filter are given in Section 4.

A third and final contribution is to show how the proposed particle filter can be extended to problem domains in which the number of targets changes over time. A variable number of targets necessitates inference in a union space of state spaces with a differing number of dimensions. In Section 5, we show how reversible-jump MCMC (RJMCMC) sampling can be used to successfully address this issue [14], [15]. In particular, we propose replacing the importance sampling step in the particle filter with an RJMCMC sampling step.

2 Background and Related Work

In what follows, we adopt a Bayesian approach to multitarget tracking. The Bayesian approach offers a systematic way to combine prior knowledge of target positions, modeling assumptions, and observation information to the problem of tracking multiple targets [16], [17]. In this section, we review related and supporting work, and we introduce the mathematical notation necessary for describing our work.

2.1 Bayesian Multitarget Tracking

Our primary goal in a multitarget tracking application is to determine the posterior distribution $P(X_t|Z^t)$ over the current *joint configuration* of the targets X_t at the current time step t, given all observations $Z^t \triangleq \{Z_1, \ldots, Z_t\}$ up to that time step. Under the commonly made assumption that target motion is Markovian, the Bayes filter offers a concise way to express the multiple target tracking problem. We consider two cases: 1) when the number of targets is fixed and 2) when the number of targets may vary.

2.1.1 Fixed Number of Targets

When the number of targets is fixed, the state X_t is simply the collection of individual target states, i.e., $X_t \triangleq \{X_{it}\}_{i=1}^{n}$, with n as the number of targets. The Bayes filter updates the posterior distribution $P(X_t|Z^t)$ over the joint state X_t of all targets given all observations up to and including time t, according to:

$$P(X_t|Z^t) = cP(Z_t|X_t) \int P(X_t|X_{t-1})P\big(X_{t-1}|Z^{t-1}\big)dX_{t-1}. \quad (1)$$

Here, c is a normalization constant, the likelihood $P(Z_t|X_t)$ expresses the *measurement model*, i.e., the probability we would have observed the measurement Z_t given the state X_t at time t, and the *motion model* $P(X_t|X_{t-1})$ predicts the state X_t given the previous state X_{t-1}.

2.1.2 Variable Number of Targets

In this case, targets may enter or leave the area under observation and both the number and identity of the targets need to be estimated. To model this, we can introduce a new variable, namely, the set of identifiers k_t of targets currently in view [18]. It is clear that there are many such hypotheses (potentially infinite) and each of these different hypotheses indexes a corresponding joint state space X_{k_t}, its dimensionality determined by the cardinality of the set k_t. For example, if the dimension of a single target is 3, $k_t = \{1, 2\}$ corresponds to a joint state $X_{\{1,2\}}$ of dimension 6, and this state space is separate from the one indexed by $k_t = \{1, 2, 3\}$. The total state space is the *union space* $\mathcal{X}$ of all such joint state spaces of different dimensionality, each indexed by a unique set of target identifiers.

The posterior $P(k_t, X_{k_t}|Z^t)$ is then a distribution over this variable-dimension space $\mathcal{X}$ and can again be expressed recursively using the Bayes filter equation:

$$\begin{aligned} P(k_t, X_{k_t}|Z^t) = cP(Z_t|k_t, X_{k_t}) \\ \times \sum_{k_{t-1}} \int P(k_t, X_{k_t}|k_{t-1}, X_{k_{t-1}}) \\ \times P\big(k_{t-1}, X_{k_{t-1}}|Z^{t-1}\big) dX_{k_{t-1}}. \end{aligned} \tag{2}$$

As before, the likelihood $P(Z_t|k_t, X_{k_t})$ expresses the *variable target measurement model*, and $P(k_{t-1}, X_{k_{t-1}}|Z^{t-1})$ is the posterior at time $t-1$. However, the *variable target motion model* $P(k_t, X_{k_t}|k_{t-1}, X_{k_{t-1}})$ is now considerably more complex: not only does it serve to predict how targets will move, but it additionally includes the probability of targets entering and leaving the area. It can be factored further by partitioning the state (k_t, X_{k_t}) into targets (k_E, X_{k_E}) that *enter* at the current time step and targets (k_S, X_{k_S}) that *stay*, yielding:

$$\begin{aligned} P(k_t, X_{k_t}|k_{t-1}, X_{k_{t-1}}) &= P(X_{k_t}|k_t, k_{t-1}, X_{k_{t-1}}) \\ &\quad \times P(k_t|k_{t-1}, X_{k_{t-1}}) \\ &= P(X_{k_E}) P\big(X_{k_S}|X_{k_S(t-1)}\big) \\ &\quad \times P(k_t|k_{t-1}, X_{k_{t-1}}), \end{aligned} \tag{3}$$

where we have assumed that targets that enter or stay behave independently of one another. In (3), $P(X_{k_E})$ predicts *where* targets will enter, $P(X_{k_S}|X_{k_S(t-1)})$ models the motion of targets that stay, and $P(k_t|k_{t-1}, X_{t-1})$ models which targets are likely to enter, stay or, leave.

2.2 SIR Particle Filters

The integrals in the Bayes filter (1) and (2) are often analytically intractable. In this paper, we make use of sequential importance resampling (SIR) particle filters to obtain sample approximations of these integrals [9], [10], [11], [12]. To derive the particle filter, one inductively assumes that the posterior $P(X_{t-1}|Z^{t-1})$ over the state X_{t-1} at the previous time step is approximated by a set of N weighted particles, i.e., $P(X_{t-1}|Z^{t-1}) \approx \{X_{t-1}^{(r)}, \pi_{t-1}^{(r)}\}_{r=1}^{N}$, where $\pi_{t-1}^{(r)}$ is the weight of the rth particle. The integral in the Bayes filter (1) can then be obtained by the following Monte Carlo approximation:

$$P(X_t|Z^t) \approx cP(Z_t|X_t) \sum_r \pi_{t-1}^{(r)} P\Big(X_t|X_{t-1}^{(r)}\Big). \tag{4}$$

A particle filter can be seen as an importance sampler for this distribution. Specifically, in the most often used variant of the particle filter one draws, at time t, N samples $X_t^{(s)}$ from the following *proposal distribution* $q(X_t)$,

$$X_t^{(s)} \sim q(X_t) \triangleq \sum_r \pi_{t-1}^{(r)} P\Big(X_t|X_{t-1}^{(r)}\Big),$$

i.e., q is a mixture density with the motion models $P(X_t|X_{t-1}^{(r)})$ as mixture components. Each sample $X_t^{(s)}$ is then weighted by its likelihood $\pi_t^{(s)} = P(Z_t|X_t^{(s)})$, resulting in a weighted particle approximation $\{X_t^{(s)}, \pi_t^{(s)}\}_{s=1}^{N}$ of the posterior $P(X_t|Z^t)$. While this view of particle filters is slightly nonstandard, it explains both resampling and prediction as sampling from the mixture and this is convenient below. Note that there exist other choices for the proposal distribution that yield more efficient variants of the particle filter [19].

2.2.1 Independent Particle Filters

When targets with identical appearance do not interact, we can run multiple *independent* particle filters. In this case, in each of n particle filters, the state X_t is simply the state of a single target. Because each particle filter samples in a small state space, it obtains a good approximation of the posterior for the single target. However, this approach is susceptible to tracking failures when interactions do occur. In a typical failure mode, illustrated in Fig. 2, several trackers will start tracking the single target with the highest likelihood score.

2.2.2 Joint Particle Filters

Instead, the interaction model we propose in Section 3 necessitates running a particle filter in the *joint* configuration space, i.e., the state X_t now includes the state of all n targets. To emphasize this point, we refer to the SIR particle filter in that case as the "joint particle filter." Unfortunately, a straightforward implementation of the joint particle filter suffers from exponential complexity in the number of tracked targets, rendering it unusable for more than three or four targets [13]. In spite of this limitation, joint particle filter-based approaches have been applied to tracking small numbers of identical targets, e.g., by binding particles to specific targets [20] or by using a mixture of particle filters along with a particle clustering step to maintain the identities of targets [21], [22]. The mixture approach has been shown to require fewer samples, but necessitates a reclustering of particles at each time step. Instead of clustering particles, the probabilistic exclusion principle approach [23] adds a term to the measurement model that assures that every feature measured belongs to only one target. However, probabilistic exclusion is less suited for appearance-based tracking and fails to capture information about the joint behavior of tracked targets. In this work, we overcome the exponential complexity of the joint particle filter by using MCMC sampling (see Section 4).

2.2.3 Variable Target Joint Particle Filter

The joint particle filter approach also extends to a variable numbers of targets. In this case, we again approximate the posterior $P(k_{t-1}, X_{k_{t-1}}|Z^{t-1})$ as a set of weighted samples $\{k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)}, \pi_{t-1}^{(r)}\}_{r=1}^{N}$, where each sample now includes a set of identifiers $k_{t-1}^{(r)}$. The Monte Carlo approximation of (2) then becomes:

$$P(k_t, X_{k_t}|Z^t) \approx cP(Z_t|k_t, X_{k_t}) \times \sum_r \pi_{t-1}^{(r)} P\Big(k_t, X_{k_t}|k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)}\Big).$$

At each time step, we now draw N samples $(k_t, X_{k_t})^{(s)}$ from the variable-dimension state space $\mathcal{X}$, according to the following mixture proposal distribution $q(k_t, X_{k_t})$:

$$\Big(k_t^{(s)}, X_{k_t}^{(s)}\Big) \sim q(k_t, X_{k_t}) \triangleq \sum_r \pi_{t-1}^{(r)} P\Big(k_t, X_{k_t}|k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)}\Big). \quad (5)$$

To sample from this distribution, we either keep or delete targets based on their previous position, move the targets that stay according to their motion model, and sample the number and initial state of new targets, all in accordance to (3). The samples are then weighted by their likelihood $\pi^{(s)} = P(Z_t|k_t^{(s)}, X_{k_t}^{(s)})$, which results in a weighted particle approximation for the posterior $P(k_t, X_{k_t}|Z^t) \approx \{k_t^{(s)}, X_{k_t}^{(s)}, \pi_t^{(s)}\}_{s=1}^{N}$ at time t.

Several authors have addressed the problem of tracking a variable number of targets in the joint particle filter framework. The BraMBLe tracker [18] uses a proposal step that adds and removes targets. Similarly, sequential importance sampling schemes have been introduced that use trans-dimensional importance weights [24], in a way that closely resemble reversible-jump MCMC [14], [15]. Finally, finite set statistics as introduced in [25], [26] enable a joint particle filter to address situations in which both the number of targets and the number of measurements changes. However, the tracking accuracy of all these approaches decreases when many targets enter the field of view and they have not been extended to tracking large numbers of targets.

2.3 MCMC Methods

Due to the limitations of importance sampling in high-dimensional state spaces, researchers have applied Markov chain Monte Carlo (MCMC) methods to tracking problems [27], [28]. All MCMC methods work by defining a Markov Chain over the space of configurations X, such that the stationary distribution $\pi(X)$ of the chain is equal to the sought posterior $P(X|Z)$ over the configurations X given the measurements Z. Typically, MCMC methods have been applied as a search technique to obtain maximum a posteriori (MAP) estimate, or as a way to diversify samples inside the framework of traditional particle filters. Both uses are discussed below.

2.3.1 MAP Estimates

A MAP estimate can be obtained by taking the most likely sample generated from the Markov Chain. In [28], MAP estimation has been applied to people tracking, more specifically to fit high-dimensional articulated models to video sequences. In [27], it has been applied to multitarget tracking in combination with temporal information from a Kalman filter. This suffers from the typical problems associated with a Kalman filter, most notably the inability to cope with non-Gaussian motion noise or multimodal distributions over the target states. More generally, however, MCMC is a method intended to produce a *sample* of a distribution and no guarantees exist about it yielding good point estimates.

2.3.2 MCMC Moves

The use of MCMC in particle filters has primarily focused on increasing the diversity of particles, as introduced by the so-called RESAMPLE-MOVE particle filtering schemes [29], [30], [31]. These schemes use periodic MCMC steps to diversify particles in an importance sampling-based particle filter. RESAMPLE-MOVE particle filters have also been designed to use RJMCMC steps to switch between variable dimensional state spaces [32], [33]. The introduction of MCMC steps to improve importance sampling suggests that MCMC alone could be used to obtain a particle filter that can effectively handle high-dimensional state spaces.

2.4 Summary

The principal distinction of our approach from previous work is three-fold: 1) we model interactions through a Markov random field (MRF) motion prior, 2) we leverage MCMC sampling to efficiently address tracking the resulting joint state of many interacting targets, and 3) we extend this approach via RJMCMC to handle a variable number of targets.

In Section 3, we introduce the MRF motion model for dealing with interacting targets. The MRF necessitates a joint particle filter which is inefficient in high-dimensional state spaces. To address this inefficiency, we introduce an MCMC-based particle filter in Section 4. In the MCMC filter, we replace the importance sampling step of the joint particle filter with an efficient MCMC sampling step. As described in Section 5, the approach extends to variable numbers of targets if RJMCMC is used. Finally, in Section 6, we present both qualitative and quantitative results on real, challenging image sequences from the social insect domain.

3 An MRF Motion Model for Interacting Targets

As a first contribution, we show how to reduce the number of tracker failures by explicitly modeling interactions using a Markov random field (MRF) motion model. Specifically, at each time step we dynamically construct an MRF that addresses interactions between nearby targets. In Section 6, we show empirically that this MRF-based approach significantly improves tracking.

An MRF is a factored probability model specified by an undirected graph (V, E), where the nodes V represent random variables and the edges E specify a neighborhood system. The joint probability over the random variables is then factored as a product of local potential functions ϕ at each node, and interaction potentials ψ defined on neighborhood cliques. See [34] for a thorough exposition on MRFs. A commonly used form is a pairwise MRF, where the cliques are restricted to the pairs of nodes that are directly connected in the graph.

Here, we use a pairwise MRF to obtain a more realistic motion model $P(X_t|X_{t-1})$, namely, one that models the interactions between nearby targets. Specifically, we adopt

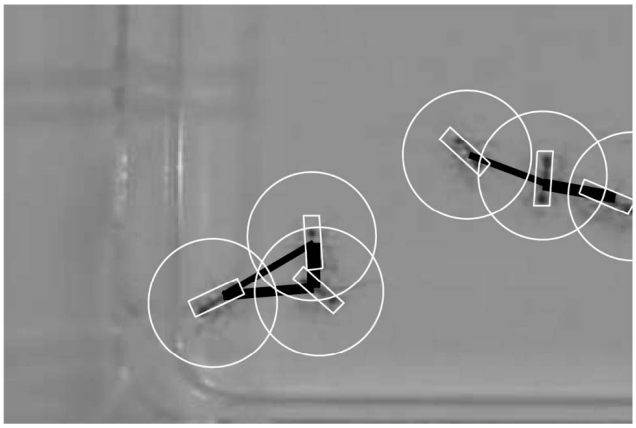


Fig. 3. To model interactions, we dynamically construct a Markov random field (MRF) at each time step, with edges for targets that are close to one another. An example is shown here for six ants. Targets that are far from one another are not linked by an edge, reflecting that there is no interaction. The circles indicate that predicted target positions in the overlapping regions are accounted for by the pairwise interaction potentials of MRF.

the following model, where the $\psi(X_{it}, X_{jt})$ are pairwise interaction potentials:

$$P(X_t|X_{t-1}) \propto \prod_i P\big(X_{it}|X_{i(t-1)}\big) \prod_{ij\in E} \psi(X_{it}, X_{jt}). \quad (6)$$

Using this form, we still retain the predictive motion models $P(X_{it}|X_{i(t-1)})$ of each individual target. Additionally, the the MRF interaction potentials $\psi(X_{it}, X_{jt})$ afford us the possibility of easily specifying domain knowledge governing the joint behavior of interacting targets. We favor the pairwise potentials over more complex formulations because they are easy to represent and implement. Also, sets of pairwise representations are sufficient for modeling all but the most complex group interactions.

As an example, in the ant tracking application we present in Section 6, we know that two ants rarely occupy the same space. Taking advantage of this can greatly help in tracking two targets that pass close to one another. An example MRF for a particular configuration is shown in Fig. 3. The circles represent a "region of influence" in which interactions are significant and the presence of another target there adds an edge in the MRF. The absence of edges between two ants captures the intuition that ants far away will not influence each other's motion.

We express the pairwise potentials $\psi(X_{it}, X_{jt})$ by means of a Gibbs distribution

$$\psi(X_{it}, X_{jt}) \propto \exp\big(-g(X_{it}, X_{jt})\big), \quad (7)$$

where $g(X_{it}, X_{jt})$ is a penalty function. For example, in the ant tracking application we use a penalty function $g(X_{it}, X_{jt})$ that only depends on the number of pixels overlap between the appearance templates associated with the two targets. It is maximal when two targets coincide and gradually falls off as targets move apart.

We use the Gibbs form above because we found that pairwise penalty functions are easier to specify than potentials. Note that this penalty function is a function on pairwise states and, consequently, can be made quite sophisticated. For example, as pointed out by a referee, it could be advantageous to include trajectory dynamics in the specification of the motion model. This is possible if velocity and/or acceleration is included in the single target states, at the cost of additional modeling complexity. In our applications, we have found that an interaction potential based on static poses only was sufficient to eliminate most tracking failures.

3.1 The Joint MRF Particle Filter

The MRF terms that model interactions can be incorporated into the Bayes filter in a straightforward manner. We can easily plug the MRF motion model (6) into the joint particle filter (4). Note that the interaction potential (7) does not depend on the previous target state X_{t-1} and, hence, the target distribution (4) for the joint MRF filter factors as

$$\begin{aligned} P(X_t|Z^t) \approx kP(Z_t|X_t) \prod_{ij\in E} \psi(X_{it}, X_{jt}) \\ \times \sum_r \pi_{t-1}^{(r)} \prod_i P\Big(X_{it}|X_{i(t-1)}^{(r)}\Big). \end{aligned} \quad (8)$$

In other words, the interaction term moves out of the mixture distribution and we can simply treat the interaction term as an additional factor in the importance weight. In other words, we sample from the joint proposal distribution

$$X_t^{(s)} \sim q(X_t) = \sum_r \pi_{t-1}^{(r)} \prod_i P\Big(X_{it}|X_{i(t-1)}^{(r)}\Big) \quad (9)$$

and weight the samples according to the following augmented likelihood expression:

$$\pi_t^{(s)} = P\Big(Z_t|X_t^{(s)}\Big) \prod_{ij\in E} \psi\Big(X_{it}^{(s)}, X_{jt}^{(s)}\Big). \quad (10)$$

The detailed steps of the joint MRF particle filter are summarized as Algorithm 1.

Algorithm 1 Joint MRF Particle Filter

At each time step t, the posterior over the target states at time $t-1$ is represented by a set of N weighted samples $\{X_{t-1}^{(r)}, \pi_{t-1}^{(r)}\}_{r=1}^N$. We then create N new particles by importance sampling, using the joint motion model (9) as the proposal distribution:

1. **Importance Sampling Step**: Repeat N times:
   a. Pick a sample $X_{t-1}^{(r)}$ with probability $\pi_{t-1}^{(r)}$.
   b. Apply the motion model $P(X_{it}|X_{i(t-1)}^{(r)})$ to each individual target to obtain a newly proposed joint state $X_t^{(s)}$.
   c. Assign a weight $\pi_t^{(s)}$ to the state $X_t^{(s)}$ according to the MRF-augmented likelihood (10). The interaction factor (7) penalizes configurations where targets overlap.
2. Return the new weighted sample set $\{X_t^{(s)}, \pi_t^{(s)}\}_{s=1}^N$.

Similarly, the MRF motion model can be incorporated into the variable target particle filter by again treating the interaction term as an additional factor in the importance weight

$$\pi_t^{(s)} = P\Big(Z_t|k_t^{(s)}, X_{k_t}^{(s)}\Big) \prod_{ij\in E} \psi\Big(X_{it}^{(s)}, X_{jt}^{(s)}\Big). \quad (11)$$

The variable target joint MRF particle filter algorithm is summarized as Algorithm 2.

Algorithm 2 Variable Target Joint MRF Particle Filter
At each time step t, the posterior over the target states at time $t-1$ is represented by a set of N weighted samples $\{k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)}, \pi_{t-1}^{(r)}\}_{r=1}^{N}$. We then create N new particles by importance sampling, using the variable target proposal distribution (5):

1. **Importance Sampling Step**: Repeat N times:
   a. Pick a sample $(k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)})$ with probability $\pi_{t-1}^{(r)}$.
   b. Determine which targets enter, stay, or leave, according to $P(k_t|k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)})$. Apply the motion model $P(X_{it}|X_{i(t-1)}^{(r)})$ to each target $i \in k_S$. Sample the position of new targets from $P(X_{k_E})$ to obtain a new state $(k_t^{(s)}, X_{k_t}^{(s)})$.
   c. Assign a weight $\pi_t^{(s)}$ to the new state according to the MRF likelihood (11).
2. Return the new weighted sample set $\{k_t^{(s)}, X_{k_t}^{(s)}, \pi_t\}_{s=1}^{N}$.

3.2 Limitations of Importance Sampling

While the above is theoretically sound, the joint MRF particle filter is not well suited for multitarget tracking. The filter suffers from exponential complexity in the number of tracked targets. If too few particles are used, all but a few importance weights will be near zero. The Monte Carlo approximation (8), while asymptotically unbiased, will have high variance. These considerations render the joint filter unusable, in practice, for more than three or four targets [13].

4 An MCMC-Based Particle Filter

As a second contribution, we show how one can efficiently sample from the posterior distribution $P(X_t|Z^t)$ over the joint target state X_t using Markov chain Monte Carlo (MCMC) sampling [35], [36]. In effect, we replace the inefficient importance sampling step of a straightforward particle filter implementation by a more efficient MCMC sampling step. In Section 4, we show that this drastically improves tracking results given the same amount of computation: The MCMC-based filter is able to cope with the high-dimensional state spaces involved, whereas the joint particle filter is not.

We base this new approach on a different Monte Carlo approximation of the posterior, in terms of *unweighted* samples. In particular, instead of using weighted samples obtained using importance sampling, we now represent the posterior $P(X_{t-1}|Z^{t-1})$ at time $t-1$ as a set of N unweighted samples $P(X_{t-1}|Z^{t-1}) \approx \{X_{t-1}^{(r)}\}_{r=1}^{N}$. Consequently, by analogy with (8), we obtain the following Monte Carlo approximation to the exact Bayesian filtering distribution

$$\begin{aligned} P(X_t|Z^t) \approx & \, cP(Z_t|X_t) \\ & \times \prod_{ij \in E} \psi(X_{it}, X_{jt}) \\ & \times \sum_r \prod_i P\left(X_{it}|X_{i(t-1)}^{(r)}\right), \end{aligned} \tag{12}$$

which also incorporates the MRF interaction potential. Now, instead of importance sampling, which is inefficient in high-dimensional state spaces, we use MCMC to sample from (12) at each time step. The sampling procedure results in a *unweighted* particle approximation for the posterior $P(X_t|Z^t) \approx \{X_t^{(s)}\}_{s=1}^{N}$.

4.1 Markov Chain Monte Carlo Sampling

As mentioned in Section 2.3, MCMC methods work by defining a Markov Chain over the space of configurations X, such that the stationary distribution of the chain is equal to a target distribution $\pi(X)$. The Metropolis-Hastings (MH) algorithm [37] is one way to simulate from such a chain. In the present case, the target distribution π is the approximate posterior (12) over joint target configurations X_t and, at each time step t, we use the MH algorithm to generate a set of samples from it. The pseudocode for the MH algorithm in this context is as follows [36].

Algorithm 3 Metropolis-Hastings Algorithm
Start with an arbitrary initial configuration $X_t^{(1)}$, then iterate for $s = 1..N-1$:

1. Propose a new assignment X_t' by sampling from the *proposal density* $Q(X_t'; X_t^{(s)})$.
2. Calculate the *acceptance ratio*

$$a = \frac{P(X_t'|Z^t)}{P(X_t^{(s)}|Z^t)} \frac{Q(X_t^{(s)}; X_t')}{Q(X_t'; X_t^{(s)})}. \tag{13}$$

3. If $a \geq 1$, then accept X_t' and set $X_t^{(s+1)} \leftarrow X_t'$. Otherwise, we accept X_t' with probability a, and reject otherwise. In the latter case, we keep the current state, i.e., $X_t^{(s+1)} \leftarrow X_t^{(s)}$.

It is standard practice to discard a number of initial "burn-in" samples, say B of them, to allow the MH algorithm to converge to the stationary distribution. In addition, to reduce the correlation between samples, it is customary to "thin out" the samples by only keeping a subset of them, taken at regular intervals. In our case, limiting the number of samples that are used to approximate (12) is also computationally advantageous, as is explained below. Please see [36] for a detailed discussion of these and other practical considerations surrounding MCMC.

4.2 Proposal Density

Careful design of the proposal density plays an important role in the success of an MCMC algorithm. To that end, we use a proposal density Q that selects a single target m, chosen from all targets with uniform probability, and updates its state X_{tm} *only* by sampling from a *single target proposal density* $Q(X_{mt}'; X_{mt})$. This "one target at a time" scheme can be formalized by:

$$Q(X_t'; X_t) \triangleq \begin{cases} \frac{1}{n} Q(X_{mt}'; X_{mt}) & \text{if } X_{-mt}' = X_{-mt} \\ 0 & \text{otherwise.} \end{cases}$$

Here, n is the number of targets, and X_{-mt} denotes the joint state of all of the targets excluding target m. As an example, the single target proposal density $Q(X_{mt}'; X_{mt})$ could simply perturb the target state according to a zero-mean normal distribution. This is the approach we take for ant-tracking in Section 6. In general, $Q(X_{mt}'; X_{mt})$ must meet two simple requirements: 1) we must be able to evaluate the density and 2) we must be able to efficiently sample from it.

The resulting acceptance ratio can be written as a product of three ratios:

$$a = \frac{P(Z_t|X_{mt}')}{P(Z_t|X_{mt})} \frac{\widetilde{P}(X_t'|Z^{t-1})}{\widetilde{P}(X_t|Z^{t-1})} \frac{Q(X_{mt}; X_{mt}')}{Q(X_{mt}'; X_{mt})}. \tag{14}$$

In the above, $\widetilde{P}(X_t|Z^{t-1})$ is the sample approximation to the predictive prior from (12):

$$\widetilde{P}(X_t|Z^{t-1}) \triangleq \prod_{ij\in E} \psi(X_{it}, X_{jt}) \sum_r \prod_i P\Big(X_{it}|X^{(r)}_{i(t-1)}\Big). \quad (15)$$

This proposal density design is computationally efficient: By only perturbing the state of one target per iteration, most factors in the acceptance ratio (14) cancel. By caching the previous likelihood evaluation $P(Z_t|X_{mt})$ for target m, only one likelihood evaluation is necessary for each iteration of the MH algorithm. In comparison, the joint particle filter requires a likelihood evaluation for each target per sample. Limiting the number of these evaluations is of great interest, as in many vision applications the likelihood includes a complex model of target appearance.

The cost of evaluating the predictive prior $\widetilde{P}(X_t|Z^{t-1})$ grows linearly with the number of samples N used to approximate the posterior $P(X_{t-1}|Z^{t-1}) \approx \{X^{(r)}_{t-1}\}^N_{r=1}$. However, careful examination again reveals that many factors remain constant if we perturb one target at a time.

4.3 Summary

The proposal density gives the MCMC-based particle filter the appealing property that the filter behaves as a set of individual particle filters when the targets are not interacting, but efficiently deals with complicated interactions when targets approach each other. The steps of the MCMC-based tracking algorithm we propose are detailed as Algorithm 4.

Algorithm 4 MCMC-Based Particle Filter

At each time step t, the posterior over target states at time $t-1$ is represented by a set of N *unweighted* samples $\{X^{(r)}_{t-1}\}^N_{r=1}$. We then create N new samples by MCMC sampling:

1. Initialize the MCMC sampler: Randomly select a sample $X^{(r)}_{t-1}$, move all targets $X^{(r)}_{i(t-1)}$ in the selected sample according to their motion models, and use the result as the initial state of the X_t Markov chain. For each target X_{it} in X_t, evaluate and cache the likelihood.
2. **MCMC Sampling Step**: Repeat $(B + MN)$ times, where B is the length of the burn-in period and M is the length of the thinning interval:
   a. Sample from the *proposal density*: Randomly pick a target m and propose a new state X'_{mt} for this target *only* by sampling from the single target proposal $Q(X'_{mt}; X_{mt})$.
   b. Compute the acceptance ratio (14).
   c. If $a \geq 1$, then accept X'_{mt}: Set the the mth target in X_t to X'_{mt} and update the cached likelihood. Otherwise, accept with probability a. If rejected, leave X_t unchanged.
3. As an approximation for the current posterior $P(X_t|Z^t)$, we return the new sample set $\{X^{(s)}_t\}^N_{s=1}$, obtained by storing every Mth sample after the initial B burn-in iterations above.

5 Handling a Variable Number of Targets Using RJMCMC

As a third contribution, we show how the proposed MCMC-based particle filter can be modified to handle a variable number of targets. We replace the MCMC sampling step of the filter with a reversible-jump MCMC (RJMCMC) sampling step, an extension of MCMC to variable dimensional state spaces [14], [15].

As with the MCMC-based particle filter, we use an unweighted sample approximation of the variable target posterior $P(k_{t-1}, X_{k_{t-1}}|Z^{t-1}) \approx \{k^{(r)}_{t-1}, X^{(r)}_{k_{t-1}}\}^N_{r=1}$ at time $t-1$. In the variable target case, each sample now includes a set of identifiers k_{t-1}. We obtain the following Monte Carlo approximation to the exact Bayes filter (2):

$$\begin{aligned} P(k_t, X_{k_t}|Z^t) \approx\ & cP(Z_t|k_t, X_{k_t}) \\ & \times \prod_{ij\in E} \psi(X_{it}, X_{jt}) \\ & \times \sum_r P\Big(k_t, X_{k_t}|k^{(r)}_{t-1}, X^{(r)}_{k_{t-1}}\Big), \end{aligned} \quad (16)$$

where the motion model $P(k_t, X_{k_t}|k^{(r)}_{t-1}, X^{(r)}_{k_{t-1}})$ factors according to (3). The RJMCMC sampling procedure, explained in detail below, will result in an *unweighted* particle approximation for the posterior $P(k_t, X_{k_t}|Z^t) \approx \{k^{(s)}_t, X^{(s)}_{k_t}\}^N_{s=1}$ at time t.

5.1 Reversible-Jump MCMC

In RJMCMC, the Metropolis-Hastings algorithm is modified to define a Markov chain over a variable dimension state space [14], [15]. The algorithm starts the chain in an arbitrary configuration $(k_t, X_{k_t}) \in \mathcal{X}$. It then selects a *move type* m from a finite set of moves that either increase the dimensionality of the state (birth), decrease it (death), or leave it unchanged. A new state (k'_t, X'_{k_t}) is then proposed using the corresponding proposal density $Q_m(k'_t, X'_{k_t}; k_t, X_{k_t})$. A move that changes the dimensionality of the state is referred to as a *jump*, as it "jumps" between state spaces of different dimensionality. Crucially, every jump has to have a corresponding reverse jump defined, e.g., a birth move should have a corresponding death move, hence the name "reversible-jump" MCMC. Denote the reverse proposal density as $Q_{m'}(k_t, X_{k_t}; k'_t, X'_{k_t})$, where m' is the reverse jump corresponding to m. The acceptance ratio is then modified as follows

$$a = \frac{P(k'_t, X'_{k_t}|Z^t)}{P(k_t, X_{k_t}|Z^t)} \frac{p_{m'}}{p_m} \frac{Q_{m'}(k_t, X_{k_t}; k'_t, X'_{k_t})}{Q_m(k'_t, X'_{k_t}; k_t, X_{k_t})}, \quad (17)$$

which will achieve detailed balance at the desired target density $P(k_t, X_{k_t}|Z^t)$.

The key component of RJMCMC is the reversibility of proposals that vary the dimensionality of the state space. To assure reversibility in the target tracking scenario, for every proposal that adds a target there must be a corresponding proposal that removes the target. Under those circumstances, detailed balance will be achieved and the chain will converge to the desired stationary distribution (16).

To achieve this, we restrict ourselves to proposals that add or remove a single target. Note that this restriction does not in any way affect the eventual convergence of the chain to the target density: The entire state space can be explored by sequentially adding and/or deleting targets to reach any of the k_t-indexed subspaces in the union space $\mathcal{X}$. In addition, by only dealing with adding or removing targets, we do not need to include a Jacobian in (17) as found in other descriptions of the reversible-jump sampler, as in our case the Jacobian is unity.

Reflecting the split of the identifier set k_t into entering targets k_E and targets that stay k_S in Section 2.1.2, we introduce two reversible pairs of proposal types, respectively, adding/deleting newly entering targets mediated by a target detector, and proposals that decide on targets staying/leaving the field of view, based on their current state.

5.2 Detector-Mediated Proposals

We assume the existence of a noisy target detector that, at each time step, returns a set of identifiers k_d of detected targets along with their estimated state X_{k_d}. Intuitively, by using the image information, the detector provides a more informed way of adding targets, i.e., it is data-driven [38]. We use the detector output to drive two types of proposals.

5.2.1 Add

If a detected target a is not yet in the identifier set k_t, propose adding it. In particular, we randomly select an identifier a with uniform probability $1/|k_d \backslash k_t|$, where $k_d \backslash k_t$ is the set of targets that have been detected but not yet added to k_t, and then append $(\{a\}, X_a)$ to the sampler state (k_t, X_{k_t}). We can write this proposal succinctly as

$$Q_A(k'_t, X'_{k_t}; k_t, X_{k_t}) = \begin{cases} \frac{1}{|k_d \backslash k_t|} & \text{if } (k'_t, X'_{k_t}) = \\ & (k_t, X_{k_t}) \cup (\{a\}, X_a) \\ 0 & \text{otherwise,} \end{cases}$$

where the union operator $\cup$ is defined in the obvious way. If all detected targets have already been added, we set the probability p_A of selecting the *Add* proposal to zero.

5.2.2 Delete

As required by the reversible-jump MCMC algorithm, the *Add* proposal above needs to have a corresponding reverse jump defined, in order to potentially move the chain back to a previous hypothesis in the union space $\mathcal{X}$. This is done as follows: We randomly select an identifier d with uniform probability $1/|k_t \cap k_d|$, where $k_t \cap k_d$ is the set of detected targets that have already been added to k_t, and then remove $(\{d\}, X_d)$ from the sampler state (k_t, X_{k_t}). This proposal can be written as

$$Q_D(k'_t, X'_{k_t}; k_t, X_{k_t}) = \begin{cases} \frac{1}{|k_t \cap k_d|} & \text{if } (k'_t, X'_{k_t}) = \\ & (k_t, X_{k_t}) \backslash (\{d\}, X_d) \\ 0 & \text{otherwise,} \end{cases}$$

where the difference operator $\backslash$ is defined in the obvious way. If no detected targets were added yet, we set the probability p_D of selecting the *Delete* proposal to zero.

5.3 Stay or Leave Proposals

The *Add/Delete* proposals above enable new targets to enter the field of view at each time step, mediated by a target detector. Additionally, we need a mechanism for deciding on the fate of targets that were already represented in the previous sample set $\{k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)}\}_{r=1}^N$. In particular, we introduce the following pair of proposals.

5.3.1 Stay

If a given target is no longer present in the current sampler state (k_t, X_{k_t}), propose to re-add it and let it *stay* anyway. To that effect, we define the set $k* \triangleq (\bigcup_{r=1} k_{t-1}^{(r)}) \backslash k_t$ of identifiers were present at time $t-1$ but are not present in the set of identfiers present at current time step k_t. We then select a target s from this set with uniform probability $\frac{1}{|k*|}$, sample a new location from

$$Q(X_s; s) = \sum_{r: s \in k_{t-1}^{(r)}} P\left(X_{st} | X_{s(t-1)}^{(r)}\right)$$

and then append $(\{s\}, X_s)$ to the state (k_t, X_{k_t}). We can write this proposal succinctly as

$$Q_S(k'_t, X'_{k_t}; k_t, X_{k_t}) = \begin{cases} \frac{1}{|k*|} Q(X_s; s) & \text{if } (k'_t, X'_{k_t}) = \\ & (k_t, X_{k_t}) \cup (\{s\}, X_s) \\ 0 & \text{otherwise.} \end{cases}$$

When the set $k*$ is empty, we set the probability p_S of the *Stay* proposal to zero.

5.3.2 Leave

The corresponding reverse jump randomly selects an identifier l from the set $k_t \backslash k_d$, with uniform probability $1/|k_t \backslash k_d|$ and removes it from the the current state of the sampler:

$$Q_L(k'_t, X'_{k_t}; k_t, X_{k_t}) = \begin{cases} \frac{1}{|k_t \backslash k_d|} & \text{if } (k_t, X_{k_t}) \backslash (\{l\}, X_l) \\ 0 & \text{otherwise.} \end{cases}$$

If the set $k_t \backslash k_d$ is empty, we set the probability p_L of selecting the *Leave* proposal to zero.

5.4 Acceptance Ratios

The four dimensionality-varying proposals discussed above have the following acceptance ratios associated with them, all specialized versions of the RJMCMC ratio (17):

$$a_A = P(Z_t|X_a) \frac{\widetilde{P}(k'_t, X'_{k_t}|Z^{t-1})}{\widetilde{P}(k_t, X_{k_t}|Z^{t-1})} \frac{p_D}{p_A} \frac{|k_d \backslash k_t|}{|k'_t \cap k_d|}, \tag{18}$$

$$a_D = \frac{1}{P(Z_t|X_d)} \frac{\widetilde{P}(k'_t, X'_{k_t}|Z^{t-1})}{\widetilde{P}(k_t, X_{k_t}|Z^{t-1})} \frac{p_A}{p_D} \frac{|k_t \cap k_d|}{|k_d \backslash k'_t|}, \tag{19}$$

$$a_S = P(Z_t|X_s) \frac{\widetilde{P}(k'_t, X'_{k_t}|Z^{t-1})}{\widetilde{P}(k_t, X_{k_t}|Z^{t-1})} \frac{p_L}{p_S} \frac{|k*|}{|k'_t \backslash k_d| Q(X'_s; s)}, \tag{20}$$

$$a_L = \frac{1}{P(Z_t|X_l)} \frac{\widetilde{P}(k'_t, X'_{k_t}|Z^{t-1})}{\widetilde{P}(k_t, X_{k_t}|Z^{t-1})} \frac{p_S}{p_L} \frac{|k_t \backslash k_d| Q(X_l; l)}{|k*'|}. \tag{21}$$

In the above, $\widetilde{P}(k_t, X_{k_t}|Z^{t-1})$ is defined as the sample approximation to the predictive prior from (16) (compare with (15)),

$$\widetilde{P}(k_t, X_{k_t}|Z^{t-1}) \triangleq \prod_{ij \in E} \psi(X_{it}, X_{jt}) \times \sum_r P(X_{k_E}) P\left(X_{k_S}|X_{k_S(t-1)}^{(r)}\right) \times P\left(k_t|k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)}\right)$$

were we have used the factored form (3) to describe the motion model. Thus, with each acceptance ratio computation, we need to sum over all samples in $\{k_{t-1}^{(r)}, X_{k_{t-1}}^{(r)}\}_{r=1}^N$. However, as before, many of the factors stay constant as only one target is added or removed at a time.

A fifth proposal type, the *Update* proposal, leaves the dimensionality of the state unchanged and has a straightforward acceptance ratio associated with it as in Section 4.2.

In all five cases, likelihood evaluations can be limited using the same caching scheme described for the MCMC filter and only one likelihood evaluation is required per sample.

5.5 Summary

The detailed steps of the RJMCMC-based tracking algorithm we propose are summarized as Algorithm 5. The MCMC-based particle filter, in which the target number is fixed, can be thought of as a specialization of the RJMCMC-based particle filter. We obtain an MCMC-based filter when the probabilities of proposals that add or remove targets are set to zero.

Algorithm 5 RJMCMC-Based Particle Filter

At each time step t, the posterior over target states at time $t-1$ is represented by an set of unweighted samples $\{k_{t-1}^{(r)} X_{k_{t-1}}^{(r)}\}_{r=1}^{N}$. We then create N new samples by RJMCMC sampling:

1. Initialize the RJMCMC sampler using the same procedure as in Algorithm 4.
2. **RJMCMC Sampling Step**: Repeat $(B+MN)$ times, with B and M are defined as before: Propose a new state (k_t', X_{k_t}'), depending on the randomly selected proposal type:
   - *Add*: Randomly add a detected target a with probability $1/|k_t \backslash k_d|$ to the current state. Compute the acceptance ratio according to (18).
   - *Delete*: Randomly select a target d with probability $1/|k_t \cap k_d|$ and delete it from the current state. Compute the acceptance ratio according to (19).
   - *Stay*: Randomly select a target s to with probability $1/|k*|$, sample its location from $Q(X_s; s)$, and re-add it. Compute the acceptance ratio according to (20).
   - *Leave*: Randomly select a target l with probability $1/|k_t \backslash k_d|$, and delete it from the current state. Compute the acceptance ratio according to (21).
   - *Update*: Use the same proposal density and acceptance ratio as in Algorithm 4.

   If $a \geq 1$, then accept the proposed state $(k_t, X_{k_t}) \leftarrow (k_t', X_{k_t}')$. Otherwise, we accept it with probability a, and reject it otherwise.
3. As an approximation to the current posterior $P(k_t, X_{k_t}|Z^t)$, return the new sample set $\{k_t^{(s)}, X_{k_t}^{(s)}\}_{s=1}^{N}$, obtained by storing every Mth sample after the initial B burn-in iterations.

6 Experimental Details and Results

We evaluated our approach by tracking through two long video sequences of two different species of roaming ants and present both quantitative results as well as a graphical comparison of the different tracker methodologies. In the first sequence, the number of targets was fixed and, in the second, the number of targets varied. The two sequences were sufficiently similar such that in the evaluation the appearance and measurement models were the same for both sequences, although the specific parameters differed. The sequences were chosen to emphasize that the variable target component, instead of a radical departure in appearance and motion models, accounted for the performance of the RJMCMC filter.

The first test sequence consisted of 10,400 frames recorded at a resolution of 720×480 pixels at 30 Hz, showing 20 ants of the species *Aphaenogaster cockerelli* roaming about a closed arena (see Fig. 1). The ants themselves are about 1 cm long and move about the arena as quickly as 3 cm per second. Interactions occur frequently and can involve five or more ants in close proximity. In these cases, the motion of the animals is difficult to predict. After pausing and touching one another, they often walk rapidly sideways or even backward. This experimental domain provides a substantial challenge to any multitarget tracker.

The second test sequence consisted of 10,000 frames taken at 15 Hz, also at a resolution of 720×480 pixels, recording a colony of *Leptothorax curvispinosus*, a different species of ant, in the process of selecting a new artificial nest site. This species is smaller, approximately 0.5 cm. The motion of these ants differs slightly from the ants in the first sequence. The ants may enter and exit from the center of the artificial nest (see Fig. 4). The numerous interactions that the individual ants undergo in this selection process are believed to be related to how the colony 'decides" to emigrate into a new nest.

We evaluated a number of different trackers with respect to a baseline tracking sequence. As no ground truth was available, we obtained the baseline sequence by running a very slow but accurate tracker and correcting any mistakes it made by hand. When we observed a tracker failure, we reinitialized the positions of the targets by hand and resumed tracking. Next, the trajectories were further processed to remove any remaining errors in the position of targets. The trajectories and videos can be downloaded at ftp://ftp.cc.gatech.edu/pub/groups/borg/pami05.

6.1 Modeling Ants

An individual ant's state at time t is modeled using a 3DOF state vector $X_{it} = [x_{it}, y_{it}, \theta_{it}]'$, where i is the ant's identifier, (x_{it}, y_{it}) its position and θ_{it} its orientation. In all test runs, the targets were initialized based on the first frame of the "ground truth" baseline sequence.

The appearance *likelihood model*, given that the cameras were stationary in all cases, uses a combination of an appearance template and a background model. The template is a rectangular region with width w and height h. We factored the measurement model assuming that the foreground $(F=1)$ pixels are independent of the background $(F=0)$ pixels. As a consequence, the likelihood only needs to be evaluated over pixels from a rectangular region with position and orientation X_{it} for target i, obtained by an inverse warp W^{-1} from the image:

$$P(Z_t|X_t) \propto \prod_i P(Z_t|X_{it}) = \prod_i \frac{P(W^{-1}(Z_t, X_{it})|F=1)}{P(W^{-1}(Z_t, X_{it})|F=0)}.$$

Because the ants periodically deform, we modeled the appearance and background using t distributions to provide robustness to outliers [39]. We model the appearance of the target $W^{-1}(Z_t, X_{it})|F=1 \sim \mathcal{T}(\mu, \Sigma, \nu)$ as a multivariate-t distribution with mean μ, diagonal covariance Σ, and degrees of freedom $\nu = 4$. The background is modeled with mean μ' and variance image Σ' distributed according to a multivariate-t distribution with degrees of freedom $\nu' = 4$. Because each pixel is independent, the background distribution can be obtained by applying the warp to the mean and variance images

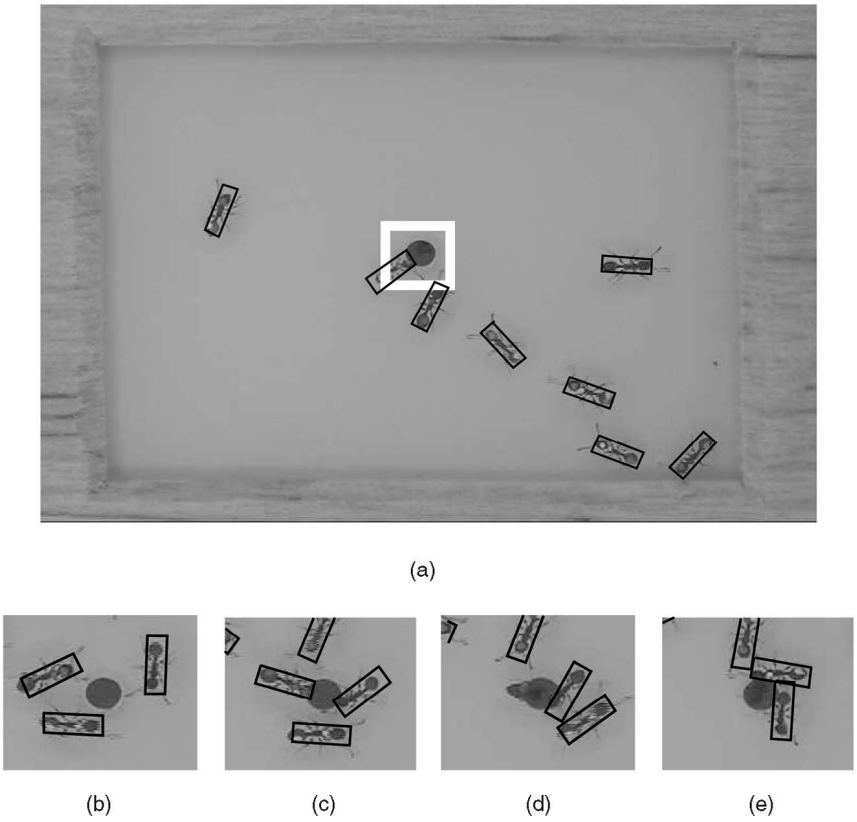


Fig. 4. (a) The MCMC-based particle filter can be extended to track variable numbers of interacting targets. Here, the ant *Leptothorax curvispinosus* may enter and exit from the circular entrance at the center of artificial nest site (highlighted by the square). The interactions that individuals undergo are believed to play an important role in nest site selection. (b) Frame 1806, (c) Frame 1810, (d) Frame 1821, and (e) Frame 1839 shows a sequence where an individual exits the nest site.

$W^{-1}(Z_t, X_{it})|F = 0 \sim \mathcal{T}(W^{-1}(\mu', X_{it}), W^{-1}(\Sigma', X_{it}), \nu')$. Both models are learned from a set of training images using EM iterations [40], [39]. To speed the image warps, the video frames were down-sampled to 180×120 pixels and the positions of the targets scaled prior to obtaining the template.

For the *motion model*, we used a normal density centered on the previous pose of target $X_{i(t-1)}$

$$X_{it}|X_{i(t-1)} = R(\theta_{t-1} + \Delta\theta)[\,\Delta x \quad \Delta y \quad 0\,]^\top + X_{i(t-1)},$$

where $[\Delta x, \Delta y, \Delta\theta] \sim [\mathcal{N}(0, \sigma_x^2), \mathcal{N}(0, \sigma_y^2), \mathcal{N}(0, \sigma_\theta^2)]$ and $R(\theta)$ a function that specifies a rotation matrix. The motion model is specified in the coordinate frame in the ant, with the head facing the y-axis. Hence, a larger variance in the y direction indicates we less certain of the ant's next position along its body.

For the *MRF interaction terms*, we used a simple linear penalty function $g(X_{it,}X_{jt}) = \gamma p(X_{it}, X_{jt})$, where p is the area of overlap between two targets, in pixels, and $\gamma = 5,000$.

6.2 Fixed Number of Targets

For the fixed number of targets case, we evaluated nine tracker/sample size combinations, the results of which are summarized in Table 1. The table shows for each combination the number of tracking failures, which was calculated by automatically identifying tracking failures when the reported position of a target deviated 50 pixels from the ground truth position. When a failure occurred, the tracker was reinitialized and tracking was resumed. Also shown is the mean and standard deviation of the per-target positional error, in pixels.

The three tracker types that were evaluated were respectively the joint MRF particle filter (joint, Algorithm 1), 20 independent particle filters (independent), and the MCMC-Based Particle Filter (MCMC, Algorithm 4). The results are qualitatively illustrated in Fig. 5 for the 2,000 samples setting. Also shown there is the number of observed actual interactions between ants for each frame, determined automatically from the ground truth sequence.

TABLE 1
Tracker Failures Observed in the
10,400 Frame Test Sequence with 20 Targets

Tracker	Samples	Failures	Per Target Error (pixels)
joint	200	606	9.99±4.35
joint	1000	491	9.52±3.60
joint	2000	407	9.33±3.42
independent	10 per target	144	5.23±1.88
independent	50 per target	79	3.09±1.33
independent	100 per target	67	2.89±1.26
MCMC	200	47	2.67±1.04
MCMC	1000	29	2.12±0.97
MCMC	2000	26	2.08±1.03

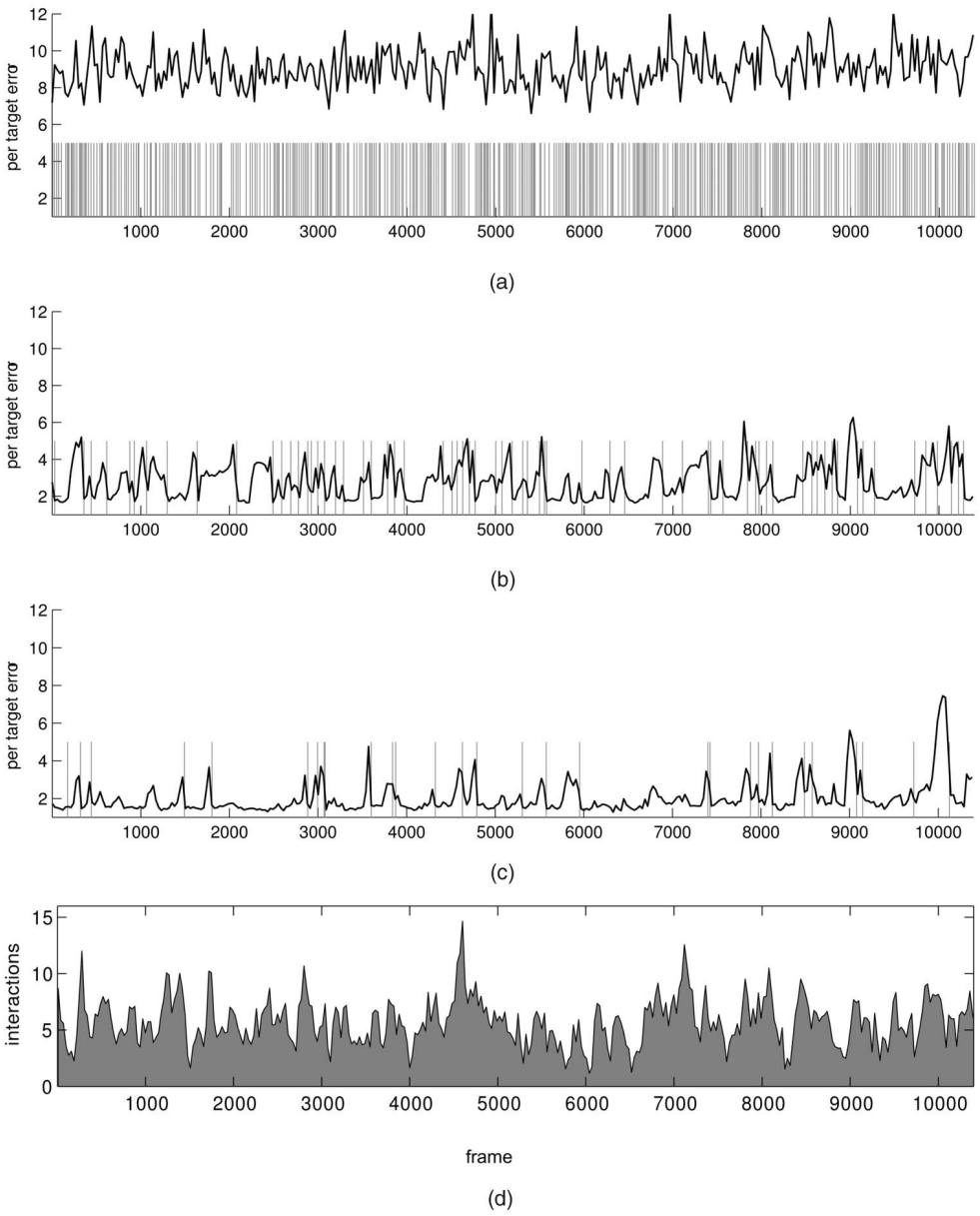


Fig. 5. (a), (b), and (c) Comparison of three trackers, each tracking 20 ants using an equivalent sample size of 2,000. Track quality is evaluated on the basis of per target error measured in pixels. Tick marks (vertical lines) show when tracking failures occur throughout the sequence. The time series plot shows average distance from ground truth in pixels (averaged per target and per second). (d) The number of interactions between ants for each frame determined from the ground truth sequence. (a) Joint particle filter, 2,000 joint particles. (b) Twenty single particle filters, 100 particles each. (c) MCMC particle filter, 2,000 particles. (d) Observed interactions.

Orientation was not included in the evaluation because of instances where the ground truth orientation was not available. In several points in the video, the ants occasionally crawled up the side of the container, such that their heads are facing directly toward the camera. In this case, a ground truth orientation was difficult or impossible to determine.

Below are the specific implementation choices made for the fixed number of targets case:

- The joint state X_t of targets is a vector containing each ant's state X_{it}.
- The motion model parameters used were $\sigma_x^2 = 4, \sigma_y^2 = 8,\ \sigma_\theta^2 = 0.4$.
- *MCMC parameters*: We discard the first 25 percent of the samples returned by the MH algorithm to allow the sampler burn-in. We propagate only $N = 10$ samples to the next time step to reduce the computational cost of the acceptance ratio and limit correlation between samples.
- We used a *single target proposal density* that updated the state according to

$$X'_{it}|X_{i(t-1)} = R(\theta_{t-1} + \Delta\theta)[\Delta x \quad \Delta y \quad 0]^\top + X_{i(t-1)},$$

where

$$[\Delta x, \Delta y, \Delta\theta] \sim [\mathcal{N}(0, \sigma_{p,x}^2), \mathcal{N}(0, \sigma_{p,y}^2), \mathcal{N}(0, \sigma_{p,\theta}^2)]$$

with parameters $\sigma_{p,x}^2 = 2, \sigma_{p,y}^2 = 2$, and $\sigma_{p,\theta}^2 = 0.2$.

TABLE 2
Tracking Errors on a 10,000 Frame Sequence with Entering and Leaving Targets

		Failures		
Tracker	Samples	Position	Number	Total
variable joint	500	71	199	270
variable joint	750	55	195	250
variable joint	1000	55	181	236
RJMCMC	500	6	29	35
RJMCMC	750	7	24	31
RJMCMC	1000	2	19	21

6.3 Variable Number of Targets

Table 2 shows the number of tracking failures for all for all the tracker/sample size combinations we evaluated in the variable number of targets case. Here, we recorded *two* types of failures: 1) failures in reported position and 2) an incorrect number of targets. To allow for variation in the time the target was detected and added to the current state, a number failure was recorded when the incorrect number of targets was reported for more than a second. As before, a position failure was recorded when a target was not found within 50 pixels of a ground truth target and orientation was not included in the evaluation.

The two trackers that were evaluated were, respectively, the variable target joint MRF particle filter (joint, Algorithm 2), and the RJMCMC-Based Particle Filter (RJMCMC, Algorithm 5). Fig. 6 shows the qualitative evaluation of these results, this time for the 1,000 samples setting. Both the positional errors and number errors are graphed separately for clarity.

The additional implementation details for the variable target case are given below:

- The distribution over new identifiers $P(k_t|k_{t-1}, X_{k_{t-1}})$ was modeled as a series of leave events and enter events. The position of entering targets $P(X_{k_E})$ was uniform around the nest site entrance. This choice captured the fact that the ants are restricted to enter and leave from a hole in the center of of the artificial nest site (see Fig. 4). The probability a leave event was set to one when the ant was positioned at nest entrance and 0.8 in the close proximity of the entrance. An enter event was generated by adding an individual detected target with probability 0.1.
- Targets were detected using a noisy, but fast, color detector to locate ant-colored regions in an image in the proximity of the nest entrance [41].
- *Parameters*: The following parameters were changed from the fixed target number case because the species and, thus, both the size and the behavior of the ants in question changed. We determined empirically on a small test sequence that the following parameters performed well: $\sigma_x^2 = 2, \sigma_y^2 = 6$, $\sigma_\theta^2 = 0.4$, $\sigma_{p,x}^2 = 1.5$, $\sigma_{p,y}^2 = 1.5$, $\sigma_{p,\theta}^2 = 0.2$.
- *Proposal Probabilities*: The probabilities of the *Add*, *Delete*, *Stay*, *Leave*, *and Update* proposal types were set to 0.15, 0.15, 0.05, 0.05, and 0.6, respectively. The values of these parameters were determined empirically to perform well on a short test sequence.

7 Conclusions and Future Work

From the quantitative results in Table 1 and the qualitative comparison in Fig. 5 for the fixed number of targets case, we draw the following conclusions:

1. The traditional joint particle filter is unusable for tracking this many targets. The track quality is very low and number of errors reported is very high.
2. The MRF motion prior improves tracking during interactions. The MCMC-based trackers perform significantly better than running independent particle filters with a comparable number of samples, both in track quality and failures reported.
3. The MCMC-based particle filter requires far fewer samples to adequately track the joint target state.

Fig. 7 shows an example of a failure mode of the MCMC-based tracker. These occur when our assumption that targets do not overlap is violated. In these cases, it is unclear that any data association method offers a solution. A more complicated joint likelihood model might be helpful in these cases.

In the variable number of targets case, we draw the following conclusions from the quantitative results in Table 2 and the qualitative comparison in Fig. 6:

1. The traditional variable target joint particle filter is unusable for tracking variable numbers of targets. For a fixed number of samples, the tracker is unable to accurately determine the number of targets when the target number increases greater than 5.
2. The RJMCMC-based tracker responds more quickly to changes in the number of targets than the variable target joint filter. Even though the variable target joint particle filter shows slightly better performance when there are fewer than five targets, many of the position errors occur when the number of targets changes. The RJMCMC tracker is able to handle many of these changes in target number.
3. The RJMCMC-based particle filter efficiently handles changing numbers of targets. Fewer samples are needed to track the joint state of the targets even when many targets have entered the field of view.

Fig. 8a shows an example failure mode of the RJMCMC tracker. This failure occurred when the ants engaged in a carrying behavior. The two targets are incorrectly identified as a single target. Tracking performance might be improved in these cases by using a switching variable that detects and correctly handles this behavior. Additionally, in Fig. 8b, individuals that straddle the nest entrance are incorrectly identified as having left the nest site. A more complex model of how targets leave the field of view might better account for this straddling behavior.

In conclusion, the results show that Markov random field motion priors are quite successful at eliminating a large number of interaction-related tracking errors. In addition, MCMC-based filters perform vastly better than a traditional particle filtering approaches in handling the resulting, highly coupled joint state space that is induced by the use of the MRF.

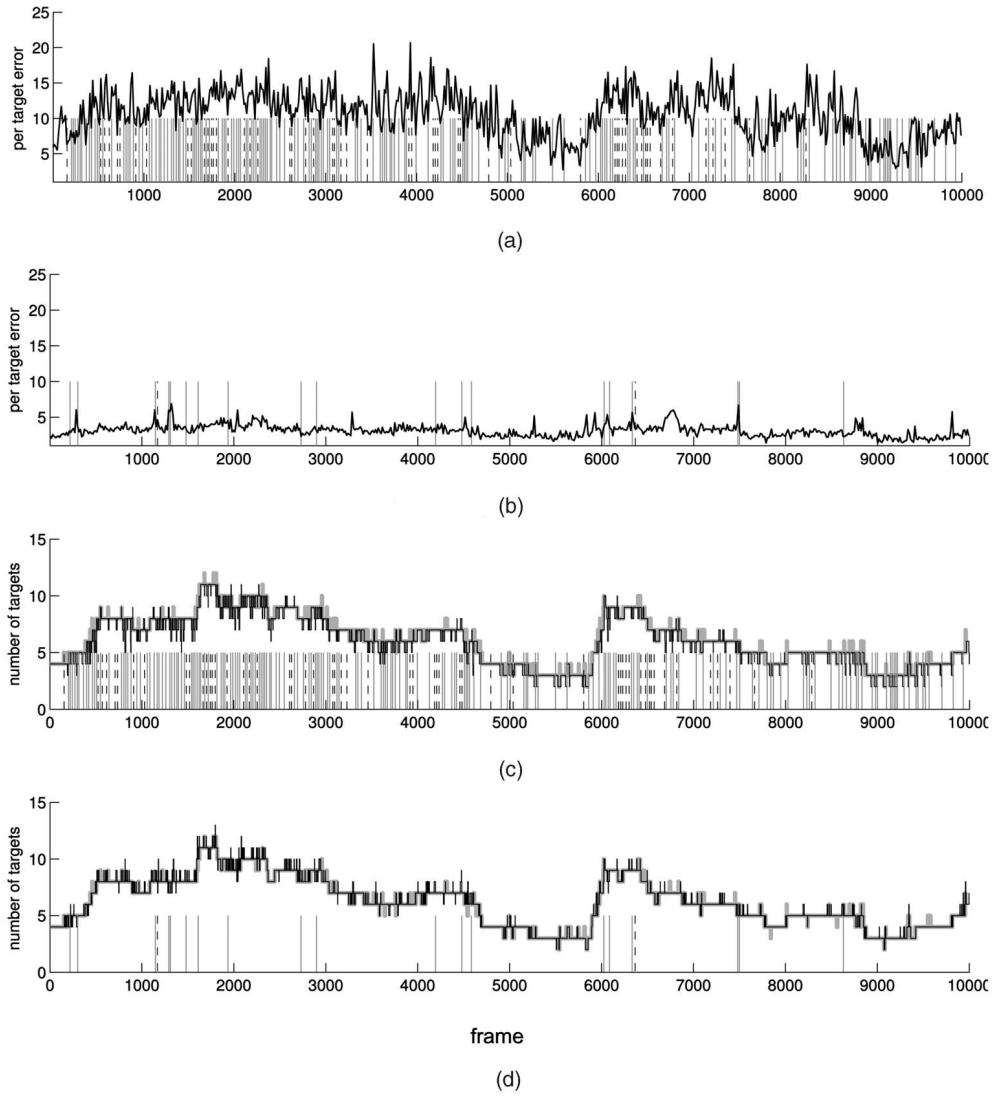


Fig. 6. (a) and (b) show comparison of track quality measured as per target error in pixels obtained from the variable target particle filter with the RJMCMC-based particle filter. Target position failures are shown as solid tick marks. The time series plot shows average distance from ground truth in pixels (averaged per target and per second). (c) and (d) compare the target number estimates returned by the variable target filter and the RJMCMC-filter. The dashed tick marks indicate a target number failure. The ground truth number of targets is shown as a thick gray line. The reported number of targets is superimposed on the ground truth number as a thin, black line. (a) Variable target joint filter, 1,000 particles. (b) RJMCMC, 1,000 particles. (c) Variable target joint filter, 1,000 particles. (d) RJMCMC, 1,000 particles.

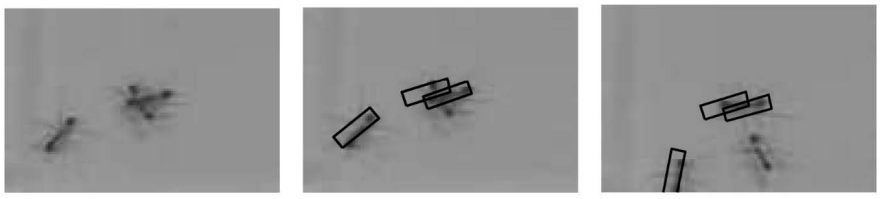
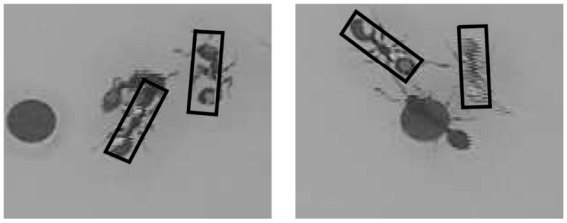


Fig. 7. Shows two examples of failure modes in MCMC-based MRF particle filter. These occur when the assumption that targets do not overlap is violated. (a) Frame 3054. Two targets undergoing extensive overlap. (b) Frame 3054. The tracker reports the incorrect position for the overlapped ant. (c) Frame 3072. The resulting tracker failure.

Finally, the use of reversible-jump MCMC successfully extended this to the case where the number of targets varies over time, a situation that is prevalent in practice.

In future work, we intend to examine methods for automatically learning behavioral models from the tracking data. In this context, switching models are of great interest because the behavioral state of a target can give important clues as to whether an interaction will take place and what type of interaction it will be. This is particularly true in the insect-tracking domain. We expect that such learned models will further improve tracking performance by more closely modeling target interactions.



(a) (b)

Fig. 8. Shows two examples of failure modes in the RJMCMC-based tracker. These occur when (a) an individual ant carries another ant into the nest and (b) individuals straddle the nest site entrance for several seconds. Note the dark circle in the figure is the nest site entrance.

ACKNOWLEDGMENTS

This work was funded under US National Science Foundation Award IIS-0219850, "ITR: Observing, Tracking and Modeling Social Multiagent Systems" and NSF Award 0347743, CAREER: Learning Executable Models of Social Systems. The authors would like to thank Stephan Pratt at the Princeton University Department of Ecology and Evolutionary Biology for the *Leptothorax curvispinosus* sequence. In addition, we would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

[1] T. Balch, Z. Khan, and M. Veloso, "Automatically Tracking and Analyzing the Behavior of Live Insect Colonies," *Proc. Conf. Autonomous Agents,* 2001.
[2] D. Reid, "An Algorithm for Tracking Multiple Targets," *IEEE Trans. Automatic Control,* vol. 24, no. 6, pp. 84-90, Dec. 1979.
[3] Y. Bar-Shalom, T. Fortmann, and M. Scheffe, "Joint Probabilistic Data Association for Multiple Targets in Clutter," *Proc. Conf. Information Sciences and Systems,* 1980.
[4] T. Fortmann, Y. Bar-Shalom, and M. Scheffe, "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association," *IEEE J. Oceanic Eng.,* vol. 8, July 1983
[5] R. Deriche and O. Faugeras, "Tracking Line Segments," *Image and Vision Computing,* vol. 8, pp. 261-270, 1990.
[6] I. Cox and J. Leonard, "Modeling a Dynamic Environment Using a Bayesian Multiple Hypothesis Approach," *Artificial Intelligence,* vol. 66, no. 2, pp. 311-344, Apr. 1994.
[7] C. Rasmussen and G. Hager, "Probabilistic Data Association Methods for Tracking Complex Visual Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 560-576, June 2001.
[8] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers, "Tracking Multiple Moving Targets with a Mobile Robot Using Particle Filters and Statistical Data Association," *IEEE Int'l Conf. Robotics and Automation,* 2001.
[9] N. Gordon, D. Salmond, and A. Smith, "Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation," *IEE Proc. F,* vol. 140, no. 2, pp. 107-113, 1993.
[10] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. European Conf. Computer Vision,* pp. 343-356, 1996.
[11] J. Carpenter, P. Clifford, and P. Fernhead, "An Improved Particle Filter for Non-Linear Problems," technical report, Dept. Statistics, Univ. of Oxford, 1997.
[12] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo Localization for Mobile Robots," *IEEE Int'l Conf. Robotics and Automation,* 1999.
[13] Z. Khan, T. Balch, and F. Dellaert, "Efficient Particle Filter-Based Tracking of Multiple Interacting Targets Using an MRF-Based Motion Model," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems,* 2003.
[14] P. Green, "Trans-Dimensional Markov Chain Monte Carlo," *Highly Structured Stochastic Systems,* Oxford Univ. Press, 2003.
[15] P. Green, "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika,* vol. 82, pp. 711-732, 1995, citeseer.nj.nec.com/green95reversible.html.
[16] L.D. Stone, C.A. Barlow, and T.L. Corwin, *Bayesian Multiple Target Tracking.* Boston: Artech House, 1999.
[17] R. Popoli and S.S. Blackman, *Design and Analysis of Modern Tracking Systems.* Artech House Radar Library, Aug. 1999.
[18] M. Isard and J. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker," *Proc. Int'l Conf. Computer Vision,* pp. 34-41, 2001.
[19] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for On-Line Non-Linear/Non-Gaussian Bayesian Tracking," *IEEE Trans. Signal Processing,* vol. 50, no. 2, pp. 174-188, Feb. 2002.
[20] D. Tweed and A. Calway, "Tracking Many Objects Using Subordinate Condensation," *Proc. British Machine Vision Conf.,* 2002.
[21] J. Vermaak, A. Doucet, and P. Perez, "Maintaining Multi-Modality through Mixture Tracking," *Proc. Int'l Conf. Computer Vision,* 2003.
[22] K. Okuma, A. Taleghani, N. de Freitas, J. Little, and D. Lowe, "A Boosted Particle Filter: Multitarget Detection and Tracking," *Proc. European Conf. Computer Vision,* 2004.
[23] J. MacCormick and A. Blake, "A Probabilistic Exclusion Principle for Tracking Multiple Objects," *Proc. Int'l Conf. Computer Vision,* pp. 572-578, 1999.
[24] J. Vermaak, S.J. Godsill, and A. Doucet, "Radial Basis Function Regression Using Trans-Dimensional Sequential Monte Carlo," *Proc. 12th IEEE Workshop Statistical Signal Processing,* pp. 525-528, 2003.
[25] H. Sidenbladh and S. Wirkander, "Tracking Random Sets of Vehicles in Terrain," *Proc. Second IEEE Workshop Multi-Object Tracking,* 2003.
[26] H. Sidenbladh and S. Wirkander, "Particle Filtering for Random Sets," 2004, unpublished.
[27] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Crowded Environment," *proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.
[28] C. Sminchisescu and B. Triggs, "Kinematic Jump Processes for Monocular 3D Human Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 69-76, 2003.
[29] W. Gilks and C. Berzuini, "Following a Moving Target—Bayesian Inference for Dynamic Bayesian Models," *J. Royal Statistical Soc., Series B,* vol. 63, no. 1, pp. 127-146, 2001.
[30] S.N. MacEachern and P. Muller, "Estimating Mixture of Dirichlet Process Models," *J. Computational and Graphical Statistics,* vol. 7, pp. 223-238, 1998.
[31] C. Berzuini, N.G. Best, W. Gilks, and C. Larizza, "Dynamic Conditional Independence Models and Markov Chain Monte Carlo Methods," *J. Am. Statistical Assoc.,* vol. 92, pp. 1403-1412, 1996.
[32] C. Andrieu, M. Davy, and A. Doucet, "Sequential MCMC for Bayesian Model Selection," *Proc. IEEE Signal Processing Workshop Higher Order Statistics,* 1999.
[33] C. Berzuini and W. Gilks, "RESAMPLE-MOVE Filtering with Cross-Model Jumps," *Sequential Monte Carlo Methods in Practice,* A. Doucet, N. de Freitas, and N. Gordon, eds. New York: Springer-Verlag, 2001.
[34] S. Li, *Markov Random Field Modeling in Computer Vision.* Springer, 1995.
[35] R. Neal, "Probabilistic Inference Using Markov Chain Monte Carlo Methods," Technical Report CRG-TR-93-1, Dept. of Computer Science, Univ. of Toronto, 1993.
[36] *Markov Chain Monte Carlo in Practice,* W. Gilks, S. Richardson, and D. Spiegelhalter, eds. Chapman and Hall, 1996.
[37] W. Hastings, "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika,* vol. 57, pp. 97-109, 1970.
[38] Z. Tu and S. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 657-673, 2002.
[39] Z. Khan and F. Dellaert, "Robust Generative Subspace Modeling: The Subspace t Distribution," Technical Report GIT-GVU-04-11, GVU Center, College of Computing, Georgia Tech, 2004.
[40] K.L. Lange, R.J.A. Little, and J.M. G. Taylor, "Robust Statistical Modeling Using the *t* Distribution," *J. Am. Statistical Assoc.,* vol. 84, no. 408, pp. 881-896, 1989.
[41] J. Bruce, T. Balch, and M. Veloso, "Fast and Inexpensive Color Image Segmentation for Interactive Robots," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems,* 2000.

Zia Khan received the BS degree in computer science and the BS degree in biology from Carnegie Mellon University in 2002. He was a research scientist at the Georgia Institute of Technology's College of Computing between 2002 and 2004. He is currently pursuing the PhD degree. His research interests include computer vision, machine learning, optimization, and interesting problems at the intersection between computer science and biology.

Tucker Balch received the BS in computer science from the Georgia Institute of Technology in 1984, the MS degree in computer science from University of California, Davis in 1988, and the PhD degree in computer science from Georgia Institute of Technology in 1998. Dr. Balch is an assistant professor of computing at the Georgia Institute of Technology in Atlanta. His research focuses on behavior-based control, learning and diversity in multirobot teams, and automatic tracking and analysis of social insect behavior. Dr. Balch has published more than 70 journal and conference papers and a book *Robot Teams* (edited with Lynne Parker) in AI and robotics. He is a member of the IEEE.

Frank Dellaert received the PhD degree in computer science from Carnegie Mellon University in 2001, the MSc degree in computer science and engineering from Case Western Reserve University in 1995, and the equivalent of the MSc degree in electrical engineering from the Catholic University of Leuven, Belgium, in 1989. He is an assistant professor in the College of Computing, at the Georgia Institute of Technology. His research focuses on probabilistic methods in robotics and vision. He has applied Markov chain Monte Carlo sampling methodologies in a variety of novel settings, from multitarget tracking to the correspondence problem. Before that, with Dieter Fox and Sebastian Thrun, he introduced the Monte Carlo localization method for robot localization, which is now a standard and popular tool in mobile robotics. Professor Dellaert has published more than 50 articles in journals and refereed conference proceedings, as well as several book chapters. He is a member of the IEEE.