

MCMC Data Association and Sparse Factorization Updating for Real Time Multitarget Tracking with Merged and Multiple Measurements

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

Abstract—In several multitarget tracking applications, a target may return more than one measurement per target and interacting targets may return multiple merged measurements between targets. Existing algorithms for tracking and data association, initially applied to radar tracking, do not adequately address these types of measurements. Here, we introduce a probabilistic model for interacting targets that addresses both types of measurements simultaneously. We provide an algorithm for approximate inference in this model using a Markov chain Monte Carlo (MCMC)-based auxiliary variable particle filter. We Rao-Blackwellize the Markov chain to eliminate sampling over the continuous state space of the targets. A major contribution of this work is the use of sparse least squares updating and downdating techniques, which significantly reduce the computational cost per iteration of the Markov chain. Also, when combined with a simple heuristic, they enable the algorithm to correctly focus computation on interacting targets. We include experimental results on a challenging simulation sequence. We test the accuracy of the algorithm using two sensor modalities, video, and laser range data. We also show the algorithm exhibits real time performance on a conventional PC.

Index Terms—Markov chain Monte Carlo, QR factorization, updating, downdating, Rao-Blackwellized, particle filter, multitarget tracking, merged measurements, linear least squares, laser range scanner.

1 INTRODUCTION

EXISTING probabilistic algorithms for tracking and data association, initially applied to radar tracking, do not adequately address merged measurements between interacting targets or multiple measurements per individual target. They assume that 1) a target can generate at most one measurement at every time step and 2) a measurement could have originated from at most one target [33], [31]. While reasonable for radar tracking, these assumptions are often violated in multitarget tracking applications in computer vision and robotics.

Two types of computer vision algorithms that are commonly used in visual tracking systems are subject to such merged and multiple measurements. *Background subtraction* algorithms return multiple blob centroids per target and, during close interactions, often return a merged blob centroid for two or more targets [41], [14]. *Interest point detectors* return a cloud of multiple measurements around a target and, as shown in the example of Fig. 1, some of these measurements are explained by multiple targets.

Like a digital video camera, a *laser range scanner* is also subject to merged and multiple measurements. This type of device is a popular sensor in robotics applications, where they are used for localization, mapping, and tracking [37], [29], [4]. One individual target typically returns several

range measurements, and during interactions many of the range measurements can be accounted for by merged targets, as illustrated in Fig. 1e.

In this paper, we introduce a probabilistic model and an approximate inference method for tracking with multiple measurements per target and merged measurements between targets. Because Markov chain Monte Carlo (MCMC) methods are effective at addressing difficult combinatorial problems both in theory and, in practice, we provide an algorithm for approximate inference in this model using an MCMC-based particle filter [32], [20], [9], [22], [23].

The MCMC-based particle filter must be used with some care to obtain an efficient and practical algorithm. In the filter, we reduce the computational cost of the predictive prior to a constant by using an idea developed for auxiliary variable particle filters [32]. Additionally, we avoid sampling over the large continuous state space of the targets by Rao-Blackwellizing the Markov chain [30], [3]. In the Rao-Blackwellized (RB) sampling scheme, an integral over the continuous parameter is analytically computed by solving a linear least squares problem during each iteration.

Our proposed Markov chain leverages sparse factorization updating and downdating techniques developed in linear algebra [16], [10]. These techniques enable the chain to incrementally construct, revise, and solve the linear least squares problem required for the RB sampling scheme. By avoiding the expense of solving the entire least squares problem, the computational cost of one iteration of the chain is significantly reduced. Moreover, the algorithm takes advantage of the sparsity of the least squares problem. Combined with a simple heuristic, the algorithm has the useful property that it focuses much of the computation on interacting targets.

• The authors are with the College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332.
E-mail: {zkhan, tucker, dellaert}@cc.gatech.edu.

Manuscript received 9 Sept. 2005; revised 21 Apr. 2006; accepted 8 May 2006; published online 12 Oct. 2006.

Recommended for acceptance by S.-C. Zhu.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0490-0905.

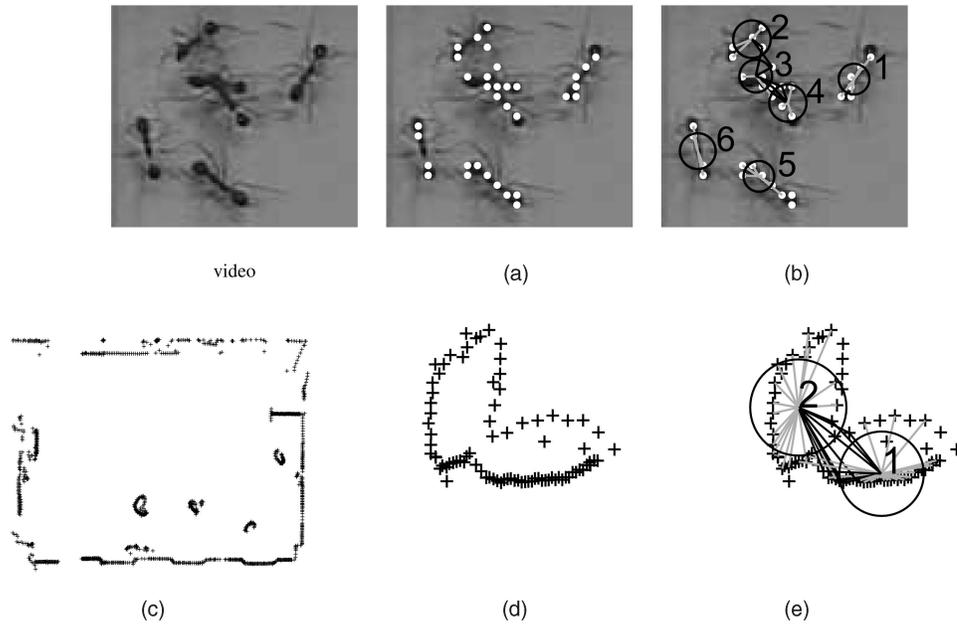


Fig. 1. (a) In this work, we address measurement data where targets may be assigned multiple measurements to a target and merged measurements can be shared between targets. The \cdot designates a measurement position. (b) The algorithm solves the data association problem: it assigns targets to measurements, simultaneously accounting for merged and multiple measurements. The light gray and black lines designate multiple and merged data associations, respectively. The black circles show the covariance ellipse around the estimated target positions. (c), (d), and (e) The algorithm is applicable to a wide range of multitarget tracking problems. Here, we track interacting targets from aligned laser range scans. Range measurements are designated by the $+$.

2 RELATED WORK

In the design of multitarget tracking systems, occlusions, interactions, and missed detections present a major challenge. Efforts to address these problems have generally focused on two areas: measurement models and motion models. Measurement models attempt to handle how these targets might appear during these difficult cases. Whereas, motion models predict target position through these difficult tracking scenarios. For several multitarget tracking applications, these challenging cases must be handled with real time requirements on the runtime of the algorithm.

Interactions and occlusions are particularly challenging when targets are identical in appearance. The trackers will tend to coalesce on the best fitting target. Several different types of solutions have been proposed to address this coalescence problem. One of the most obvious is the addition of 3D information [19], [48].

When 3D information is unavailable, low-level features such as contours and optical flow have been exploited. Methods such as the “probabilistic exclusion principle” focus on assigning measurement ownership to the correct target using differences in contour information [28], [2], [18]. Similarly, low-level information such as optical flow has been utilized to differentiate occluded targets [36]. Low-level appearance information combined with dynamic layer representation constraints has also been shown to be effective in tracking through occlusions [42].

Further robustness to occlusions can be gained when low-level information is combined with a part-based appearance model of a target. Specifically, articulated models of targets such as people model more of the shape and topological changes of a target [39], [35]. By exploiting this additional modeling information, the methods are more robust to occlusions [34]. While appealing, articulated models come at a significant computational cost.

To maintain real-time performance, several methods rely on detection measurements obtained from a background subtraction or change detection algorithms. To address interactions in this context, measurement models have been modified to model merged measurements between target. Explicitly modeling these shared measurements has been shown to be more effective tracking through occlusions [14], [26], [5], [27]. This is the approach we use in this work.

The tracker coalescence problem has also been addressed using constraints on how targets move since appearance information may not allow differentiation of individual targets. Traditional data association methods have used kinematic information to predict target position through occlusions [33]. But, the approach becomes less effective in applications where the target motion is hard to predict. Consequently, motion models have been augmented with Markov random fields priors to deal with interactions by either adding a constraint that targets rarely overlap or creating an optimization problem where two targets compete for data accounting for target appearance [23], [47], [40]. Similarly, a heuristic approach involving periodic clustering of samples was introduced into a particle filter to prevent target coalescence [46]. When targets cannot be separated as individuals, probabilistic data association methods have been extended to groups of targets where the state vector includes an estimate of how many targets participate in a group [13].

Traditional data association approaches have used kinematic information to deal with missed detections [33]. This is the approach we use in this work. Whereas, multiple hypothesis tracking methods delay the decision on data association to gain robustness to missed detections and unpredictable motion [7]. Similarly, Markov chain methods that use a sliding window over a period of time have been

shown effective at dealing with missed detections of closely moving targets [31].

In addition to the challenges associated with occlusions, interactions, and missed detections, multitarget tracking systems often have real time requirements. In surveillance, the position of targets is often needed as soon as it becomes available to determine if a target is friend or foe [21], [17]. Similarly, a robot must respond to the predicted position and trajectories of targets from a laser scan within seconds [37]. For the specific applications in the study of animal behavior that we address, tools that facilitate behavioral data collection must allow easy and quick acquisition, modification, and annotation of tracking data to gain widespread use [1], [2], [25].

Consequently, we address the specific problem of tracking interacting targets which are essentially identical in appearance in real time. We assume only target position information is desired. To meet the real-time requirements, we use a model in which measurements are a noisy cloud of detections around a target. We handle occlusions and interactions with a combined approach. We use a kinematic motion model to predict target position in these difficult cases. In addition, we exploit both merged and multiple measurements to compute accurate estimates of each target's current and predicted position.

3 A PROBABILISTIC TRACKING MODEL

In this section, we describe a probabilistic model for tracking that addresses the problem of multiple measurements per target and merged measurements between targets. We assume that there are N targets, where N is fixed, and write their joint state as X_t . At each time step we have M measurements Z_t , where M can change at each time step, governed by a data-association vector J_t (explained in detail below).

First, we specify the joint distribution $P(Z_{1:t}, X_{0:t}, J_{1:t})$ over the actual measurements $Z_{1:t}$, data associations $J_{1:t}$, and states $X_{0:t}$ of the targets between time steps 0 to t ,

$$P(Z_{1:T}, X_{0:T}, J_{1:t}) = P(X_0) \prod_{t=1}^T P(X_t|X_{t-1}) \\ \times P(Z_t|J_t, X_t)P(J_t),$$

where we assumed that the target motion is Markov, each measurement set Z_t is conditionally independent given the current state X_t , and X_t depends only on the previous time step. Since measurements arrive in random order, the actual state X_t of the targets does not provide us with any information on the data association. Consequently, we also assume that the prior over data associations $P(J_t)$ does not depend on the target state

$$P(Z_t, J_t|X_t) = P(Z_t|J_t, X_t)P(J_t).$$

It is convenient to write inference in this model recursively via the Bayes filter. The objective is to infer the current position X_t of the targets given all of the measurements $Z_{1:t}$ observed so far. In particular, the posterior distribution $P(X_t|Z_{1:t})$ over the joint state X_t of all present targets given all observations $Z_{1:t} = \{Z_1, Z_t\}$ up to and including time t is updated according to the recursive formula

$$P(X_t|Z_{1:t}) = k \sum_{J_t} P(X_t, J_t|Z_{1:t-1}) \\ = k \sum_{J_t} P(Z_t|J_t, X_t)P(J_t) \\ \times \int_{X_{t-1}} P(X_t|X_{t-1})P(X_{t-1}|Z_{1:t-1}), \quad (1)$$

where k is a normalizing constant.

In the sections below, we concentrate on deriving an expression for the posterior $P(X_t|Z_{1:t})$ on both X_t and the data association J_t , by providing further details on the *motion model* $P(X_t|X_{t-1})$ and the *measurement model* $P(Z_t|J_t, X_t)$. In Section 4, we will deal with the case of unknown data associations.

3.1 The Motion Model

For the motion model, we assume a standard linear-Gaussian model. That is, we assume that the initial joint state is Gaussian

$$P(X_0) = \mathcal{N}(X_0; m_0, V_0),$$

where m_0 is the mean and V_0 is the corresponding covariance matrix. In addition, we assume that targets move according to a linear model with additive Gaussian noise,

$$P(X_t|X_{t-1}) = \mathcal{N}(X_t; AX_{t-1}, \Gamma), \quad (2)$$

where Γ is the prediction covariance and A is a linear prediction matrix. We model the motion of each target independently, i.e., the linear prediction matrix has a sparse block-diagonal structure:

$$A = \begin{bmatrix} A_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & A_{NN} \end{bmatrix}.$$

3.2 The Measurement Model

As shown in Fig. 2, we represent a data association vector J_t by a bipartite graph

$$J_t = (X_{(1:N)t}, Z_{(1:M)t}, E_t),$$

which consists of a set of target nodes $X_{(1:N)t}$ and measurement nodes $Z_{(1:M)t}$, a subset of which are connected by a set of edges E_t .

Given the data association J_t , we can divide the measurements into clutter and observations, respectively,

$$P(Z_t|X_t, J_t) = P(Z_{c,t}|J_t)P(Z_{o,t}|J_t, X_t). \quad (3)$$

Formally, this can be modeled as a function $(Z_{c,t}, Z_{o,t}) = f(J_t, Z_t)$. We assume that each clutter measurement, i.e., an unassigned measurement like Z_{1t} in Fig. 2, is independently and uniformly generated over the field of view. Consequently, the clutter model is a constant C proportional to the number of clutter measurements $|Z_{c,t}|$:

$$P(Z_{c,t}|J_t) = \frac{|Z_{c,t}|}{C}.$$

The constant C is related to the size of the field of view—in a 720×480 image, $C = 720 \cdot 480$.

To model the observations, we map the data association to a *sparse* measurement matrix $H = g(J_t)$ in a Gaussian observation model

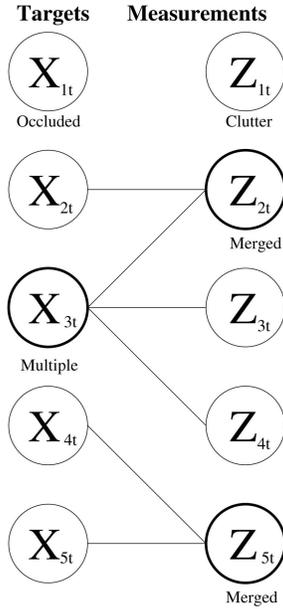


Fig. 2. The space of data associations include all bipartite graphs between targets and measurements. Measurements can be shared between targets and a target can be assigned multiple measurements.

$$P(Z_{o,t}|J_t, X_t) = \mathcal{N}(Z_{o,t}; HX_t, \Sigma).$$

The columns of the matrix H correspond to individual target states and the rows observed measurements $Z_{o,t}$. For every edge connected to a single measurement node, a measurement matrix $H_{i,j}$ is placed in the corresponding row and column. A typical example where 2D target states are estimated from 2D measurements is shown in Fig. 3. Here, the measurement matrix is a 2×2 identity matrix.

Several additional details need to be taken into account when constructing H . For merged measurements where there are several edges connecting the measurement to multiple targets, the identity matrix is multiplied by the inverse of the number of edges and then placed in the corresponding row and column. When we multiply by the inverse of the number of targets, we assume that merged measurements occur in the middle of merged targets. We find for a number of applications this approximation is sufficient because the assumption is not a strict constraint on the position of the merged measurement. The Gaussian noise model Σ_{ii} on a merged measurement allows for some variation in its position. When a target is not assigned a measurement, we apply the motion model (2) to the unassigned target. The mean of position at the next time step is the predicted position of the target. The covariance around the mean position is expanded by the prediction covariance Γ to capture that we are less certain of the target's position since no measurement was assigned.

As far as the measurement covariance matrix Σ is concerned, we assume that each measurement is generated independently and we once again obtain a block-diagonal structure:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Sigma_{MM} \end{bmatrix}.$$

In summary, by using a Gaussian measurement model, we model the measurements as noisy clouds of detections

$$H = \begin{bmatrix} 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \end{bmatrix}$$

$$X_t = \begin{bmatrix} x_{1t} \\ x'_{1t} \\ x_{2t} \\ x'_{2t} \\ x_{3t} \\ x'_{3t} \\ x_{4t} \\ x'_{4t} \\ x_{5t} \\ x'_{5t} \end{bmatrix} \quad Z_{o,t} = \begin{bmatrix} z_{2t} \\ z'_{2t} \\ z_{3t} \\ z'_{3t} \\ z_{4t} \\ z'_{4t} \\ z_{5t} \\ z'_{5t} \end{bmatrix} \quad Z_{c,t} = \begin{bmatrix} z_{1t} \\ z'_{1t} \end{bmatrix}$$

Fig. 3. H shows the sparse measurement matrix for the bipartite graph in Fig. 2. X_t shows the joint target state. Both the individual measurements and individual targets are two-dimensional $[x_{it}, x'_{it}]^T$ and $[z_{it}, z'_{it}]^T$, respectively. These 2D measurements can be an (x, y) position on an image. The observed measurements $Z_{o,t}$ and the clutter measurements $Z_{c,t}$ are shown as well.

around targets. This is designed to yield an efficient algorithm for real-time inference in this model using the computational techniques described in Section 5.1.

4 SEQUENTIAL INFERENCE

In this section, we provide an algorithm for approximate inference in this model using an MCMC-based particle filter. MCMC methods approximate a probability distribution by a set of samples drawn from the distribution. They are also effective at addressing difficult combinatorial problems in theory and, in practice, [32], [20], [9]. In a typical MCMC-based particle filter formulation, one starts by inductively assuming that the posterior distribution over the joint state of the targets at the previous time step is approximated by a set of S samples

$$P(X_{t-1}|Z_{1:t-1}) \approx \left\{ X_{t-1}^{(s)} \right\}_{s=1}^S. \quad (4)$$

Given this representation, we obtain the following Monte Carlo approximation of the Bayes filter (1):

$$P(X_t|Z_{1:t}) \approx k \sum_{J_t} P(Z_t|J_t, X_t) P(J_t) \times \frac{1}{S} \sum_{s=1}^S P(X_t|X_{t-1}^{(s)}). \quad (5)$$

This becomes the target distribution from which we sample data associations and target states using a Markov chain at each time step.

A straightforward implementation of (5) is intractable. As a consequence, we propose several improvements that together provide an efficient, practical algorithm.

4.1 An Auxiliary Variable Sampler

We inductively assume that we can approximate the posterior $P(X_{t-1}|Z_{1:t-1})$ by the following mixture of Gaussians

$$P(X_{t-1}|Z_{1:t-1}) \approx \frac{1}{S} \sum_{s=1}^S \mathcal{N}(X_t; m_{t-1}^{(s)}, V_{t-1}^{(s)}).$$

If we substitute this approximation into the Bayes filter (1), we obtain

$$P(X_t|Z_{1:t}) \approx \frac{k}{S} \sum_{J_t} P(Z_t|X_t, J_t) P(J_t) \times \sum_{s=1}^S \int_{X_{t-1}} P(X_t|X_{t-1}) \mathcal{N}(X_t; m_{t-1}^{(s)}, V_{t-1}^{(s)}). \quad (6)$$

Because the target motion model is linear-Gaussian, the predictive density over X_t for each value of the mixture indicator s can be calculated analytically

$$\int_{X_{t-1}} P(X_t|X_{t-1}) \mathcal{N}(X_t; m_t^{(s)}, V_t^{(s)}) = \mathcal{N}(X_t; Am_{t-1}^{(s)}, Q_t^{(s)}), \quad (7)$$

where the mean is the predicted position $Am_{t-1}^{(s)}$ and $Q_t^{(s)} = A(V_{t-1}^{(s)} + \Gamma)A^T$ is the prediction covariance. Hence, the predictive prior $P(X_t|Z_{1:t-1})$ on the current state is also a mixture of Gaussians

$$P(X_t|Z_{1:t-1}) \approx \frac{1}{S} \sum_{s=1}^S \mathcal{N}(X_t; Am_{t-1}^{(s)}, Q_t^{(s)}) \quad (8)$$

and the *sequential* Monte Carlo approximation to the target posterior (6) becomes

$$P(X_t|Z_{1:t}) \approx k \sum_{J_t} P(Z_t|X_t, J_t) P(J_t) \times \frac{1}{S} \sum_{s=1}^S \mathcal{N}(X_t; Am_{t-1}^{(s)}, Q_t^{(s)}). \quad (9)$$

A single evaluation of (9) is intractable due the large summation over the space of data associations J_t combined with the summation over the mixture indicator s . To address this problem, we make a second Monte Carlo approximation

$$P(X_t|Z_{1:t}) \approx k \sum_{w=1}^W P(Z_t|J_t^{(w)}, X_t^{(w)}) \times P(J_t^{(w)}) \mathcal{N}(X_t^{(w)}; Am_{t-1}^{(s')}, Q_t^{(s')})$$

using a set of sampled states, data associations, and mixture indicators $\{X_t^{(w)}, J_t^{(w)}, s^{(w)}\}_{w=1}^W$, where $s' = s^{(w)}$ is the w th sampled mixture indicator drawn from the following target density

$$\hat{\pi}(X_t, J_t, s) = k P(Z_t|J_t, X_t) P(J_t) \mathcal{N}(X_t; Am_{t-1}^{(s)}, Q_t^{(s)}). \quad (10)$$

Note that including the mixture indicator in the target density is similar to the technique used in auxiliary variable particle filters to remove the computational dependence of the target ratio on the number of samples [32].

4.2 A Rao-Blackwellized Target Density

The second improvement comes from the fact that drawing samples from the continuous state space X_t is unnecessary: we can analytically marginalize out the current state X_t . This has two benefits: it yields a computationally efficient Markov chain sampler using the techniques described in Section 5.1 and it reduces the variance of the Monte Carlo estimate of the joint state [30], [3], [24].

In particular, we use the terms for the likelihood from Section 3.2 and obtain a Rao-Blackwellized target density

$$\pi(J_t, s) = P(Z_{o,t}|J_t) P(J_t) \times \int_{X_t} \mathcal{N}(Z_{o,t}; HX_t, \Sigma) \mathcal{N}(X_t; Am_{t-1}^{(s)}, Q_t^{(s)}). \quad (11)$$

The key observation here is that the product of the likelihood and the predictive prior

$$q(X_t) = \mathcal{N}(Z_{o,t}; HX_t, \Sigma) \mathcal{N}(X_t; Am_{t-1}^{(s)}, Q_t^{(s)})$$

is proportional to a Gaussian. As a result, the integral over X_t is analytically tractable and is also Gaussian. The W samples $\{J_t^{(w)}, s^{(w)}\}_{w=1}^W$ drawn from the Rao-Blackwellized target density can be used to construct a new mixture of Gaussians over the current state

$$P(X_t|Z_{1:t}) = \frac{1}{W} \sum_{w=1}^W \mathcal{N}(X_t; m_t^{(w)}, V_t^{(w)}),$$

where $m_t^{(w)}$ is the mean and $V_t^{(w)}$ is the covariance of the target state at the current time step.

We compute the values $m_t^{(w)}$ and $V_t^{(w)}$ by integrating over X_t , making use of the fact that the integral of any function $q(X_t)$ proportional to a Gaussian is equal to the maximum of that function X_t^* times a proportionality constant:

$$\int_{X_t} q(X_t) = \sqrt{|2\pi V_t|} q(X_t^*).$$

Here, V_t is the covariance at X_t^* . In our case, the function $q(X_t)$ is the product of the likelihood and the predictive prior

$$q(X_t) = \mathcal{N}(Z_t; HX_t, \Sigma) \mathcal{N}(X_t; Am_{t-1}^{(s)}, Q_t^{(s)}).$$

Because both terms are linear Gaussian, the maximum of $q(X_t)$ is the minimum of the following linear least squares problem.

$$X_t^* = \min_{X_t} \left\| B^{-1} \begin{bmatrix} I \\ H \end{bmatrix} X_t - \begin{bmatrix} Am_{t-1}^{(s)} \\ Z_{o,t} \end{bmatrix} \right\|_2, \quad (12)$$

where I is the identity matrix and B is the Cholesky factor of the combined measurement and prior covariance

$$B^T B = \begin{bmatrix} Q_t^{(s)} & 0 \\ 0 & \Sigma \end{bmatrix}.$$

The Cholesky factor has a block diagonal structure

$$B = \begin{bmatrix} B^{(s)} & 0 & 0 \\ 0 & B_{11} & 0 \\ 0 & 0 & \ddots \end{bmatrix}$$

since we assume measurements are independent $B_{ii}^\top B_{ii} = \Sigma_{ii}$. The measurement covariance also factors separately $Q_t^{(s)} = B_{(s)}^\top B_{(s)}$.

We find the solution of the linear least squares problem (12), where the system matrix Λ and the right-hand side (rhs) b can be written as follows:

$$\Lambda = B^{-1} \begin{pmatrix} I \\ H \end{pmatrix} \quad b = B^{-1} \begin{pmatrix} Am_{t-1}^{(s)} \\ Z_{o,t} \end{pmatrix} \quad (13)$$

by computing the QR factorization of the system matrix

$$\Lambda = Q \begin{bmatrix} R \\ 0 \end{bmatrix}.$$

The orthogonal Q factor is applied to the rhs

$$\begin{bmatrix} z \\ v \end{bmatrix} = Q^\top b.$$

Using the resulting vector $Q^\top b$ and the upper triangular R factor, we compute the residual ρ , the least squares solution X_t^* and the covariance V_t at the solution

$$X_t^* = R^{-1} z \quad \rho = \|v\|_2 \quad V_t = R^{-\top} R.$$

In addition, the determinant of the covariance $|V|$ can be computed in linear time $O(N)$ time from the R factor

$$|V_t| = |R^{-\top} R| = \prod_{i=1}^N \frac{1}{r_{i,i}^2},$$

where $r_{i,i}$ is the i th entry along the diagonal of the R factor.

The value of $q(X_t^*)$ is computed using the residual ρ of the least squares problem. It is the following function of the residual:

$$\frac{\exp(-\rho^2)}{\sqrt{|2\pi\Sigma| |2\pi Q_t^{(s)}|}}.$$

In summary, we compute the final Rao-Blackwellized target distribution $\pi(\cdot)$ by solving (12), computing the residual at X_t^* , and combining it with a prior term over a data association and a clutter model

$$\pi(J_t, s) = kP(J_t) \frac{|Z_{c,t}|}{C} \frac{\sqrt{|2\pi V_t|}}{\sqrt{|2\pi\Sigma| |2\pi Q_t^{(s)}|}} \exp(-\rho^2). \quad (14)$$

We construct the mean at the current time step using the optimal state $m_t^{(w)} = X_t^*$ and the covariance term $V_t^{(w)} = V_t$ from the R factor of the QR factorization.

5 RAO-BLACKWELLIZED MARKOV CHAIN

In this section, we describe the details on how to efficiently sample from the target density $\pi(J_t, s)$ (14) in a Markov chain Monte Carlo sampler. Here, we use the Metropolis Hastings (MH) algorithm to construct a Markov chain that has this density as its stationary distribution [15]. The most important component of an MCMC sampler is how to propose new hypotheses (J_t, s) from old ones. We use one of two simple proposals: we either 1) update the auxiliary variable s or 2) add/remove an edge from the data association J_t . Using these proposals together we can explore the entire state space, and the MH algorithm then assures that the resulting Markov

chain achieves detailed balance and converges to the target density $\pi(J_t, s)$. As opposed to data driven proposals [45] which can be computationally costly, the proposals we select are purely random and were chosen primarily for their computational efficiency.

The primary computational cost of a Markov chain comes from the evaluation of the target ratio at each iteration. In most applications, a single evaluation of the target ratio takes constant time. In our case, a single evaluation of the target ratio involves finding a solution to a linear least squares problem whose computational complexity is $O(MN^2)$, where M is the number of measurements and N is the number of targets. As a consequence, we propose using matrix factorization updating and down-dating techniques to reduce the computational cost of a single evaluation of the target ratio to $O(N^2)$. This reduction in computational cost comes from the fact these techniques construct, modify and solve the linear least squares problem incrementally. The computation used to construct the old R factor is utilized in the construction of the new R' factor. To handle these incremental computations, we also keep track of the the least-squares problem associated with the data association. In each iteration of the Markov chain, we not only propose a new auxiliary variable s' and data association J'_t , but we additionally compute a new R' factor and corresponding rhs vector z' .

5.1 Updating and Downdating for Efficiency

Specifically, updating and down-dating techniques enable the addition and removal of rows and measurements from a linear least squares problem [16], [10]. Updating and down-dating techniques are based on the observation that Q matrix is formed by accumulating a sequence of orthogonal transformations G_i that zero columns in the system matrix to form an upper triangular R matrix. We update the R factor and rhs vector z by forming an augmented matrix with the new row h and rhs element b to be added to the matrix. Next, we apply a sequence of orthogonal transformations G_i^\top to zero the new row h in the augmented R factor

$$G_N^\top \dots G_2^\top G_1^\top \begin{bmatrix} R & z \\ h & b \end{bmatrix} = \begin{bmatrix} R' & z' \\ 0 & \eta \end{bmatrix}.$$

If the orthogonal transformations are Givens rotations, the row h and rhs element b are added to the R factor and the rhs vector z is updated accordingly. When we use hyperbolic rotations, the row h and rhs element b are removed. In this case, the factorization is said to be down-dated. We use η to compute the new residual $\rho' = \sqrt{\rho^2 + \eta^2}$ and $\rho' = \sqrt{\rho^2 - \eta^2}$ for updating and down-dating, respectively. Updating and down-dating returns an updated R' factor, z' rhs vector, and ρ' residual. We repeat the updating and down-dating procedure, row by row, to add and remove multiple rows and rhs elements. Rows are replaced in the factorization by performing a sequence of updating operations, to add the new rows and rhs elements, followed by a sequence of down-dating operations to remove the rows and rhs elements.

Because both Givens and hyperbolic rotations zero one element at a time of the row h , they can also take advantage of the *sparsity* of the new row h and the R factor. The off diagonal elements of R are zero when there exists no correlation between the positions of two targets in the joint state X_t^* . This occurs because the R factor is the inverse of the Cholesky

factorization of the covariance $V_t = R^{-\top}R$. As targets interact, the off diagonal blocks of the covariance matrices will fill with nonzero correlations and, so, will the off diagonal terms of the R factor. To help reduce the computational cost, we use the following heuristic: When two targets i and j are more than γ threshold apart, the off diagonal blocks V_{ij} and V_{ji} are set to zero in the covariance matrix. For instance, if target $i = 1$ and target $j = N$ are greater than γ distance apart, the covariance matrix is modified as follows:

$$\begin{bmatrix} V_{11} & 0 & \mathbf{V}_{1N} \\ 0 & \ddots & 0 \\ \mathbf{V}_{N1} & 0 & V_{NN} \end{bmatrix} \rightarrow \begin{bmatrix} V_{11} & 0 & \mathbf{0} \\ 0 & \ddots & 0 \\ \mathbf{0} & 0 & V_{NN} \end{bmatrix}.$$

A similar technique is often used in the simultaneous mapping and localization literature to reduce computational complexity [43]. With this heuristic, the algorithm acquires the nice property that a measurement can be assigned to a target in time $O(I^2)$, where I is the number of interacting targets. Next, we describe how to use these techniques in conjunction with the two proposals used by the Markov chain.

5.2 Auxiliary Variable Proposal

The first type of proposal, the *auxiliary variable proposal*, is selected with probability p_s and simply selects a new auxiliary variable s' uniformly at random (uar). Modifying the auxiliary variable s' necessitates only replacing the rows of the prior term in the factorization—the only rows that depend on s . To derive which rows need to be replaced in the factorization, we examine the rhs of the least squares problem (13) and multiply both the system matrix Λ and the rhs b by the inverse of the Cholesky factor B^{-1} .

In this matrix multiplication, we exploit the special block structure of the inverse Cholesky factor B^{-1} and see that the inverse of the Cholesky factor of the covariance of the previous sample $B_{(s)}^{-1}$ must be replaced with the new sample Cholesky factor $B_{(s')}^{-1}$ in the system matrix Λ , and the corresponding rows in the rhs $B_{(s)}^{-1}Am_{t-1}^{(s)}$ must be replaced by a new covariance weighted prediction $B_{(s')}^{-1}Am_{t-1}^{(s')}$. The replacement is conducted by first adding the rows of the new sample Cholesky factor and weighted prediction and then down-dating the rows of the old sample Cholesky factor and weighted prediction.

Next, this proposal is accepted with the following probability

$$a = \min \left(\frac{|Z'_{c,t}| \sqrt{|2\pi V'_t|} \sqrt{|2\pi \Sigma|} |2\pi Q_t^{(s)}| \exp(-\rho'^2)}{|Z_{c,t}| \sqrt{|2\pi V_t|} \sqrt{|2\pi \Sigma'|} |2\pi Q_t^{(s')}| \exp(-\rho^2)}, 1 \right).$$

Note that the normalizing constant k cancels in the target ratio. Moreover, we assume the prior over data associations $P(J_t)$ is uniform. Consequently, we can treat it as a constant and the term also cancels in the target ratio. In order to avoid computing the inverse Cholesky factor $B_{(s)}^{-1}$ and the weighted prediction $B_{(s)}^{-1}Am_{t-1}^{(s)}$ at each iteration of the chain, we compute each of these values before the first iteration.

5.3 Edge Proposal

The second type, the *edge proposal*, is selected with probability $(1 - p_s)$, and selects a measurement Z_{it} and a target X_{jt} uar. If an edge exists between the target and the measurement the proposal removes the edge, otherwise it adds an edge. The

$$\Sigma_{ii} = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix} \quad B_{ii}^{-1} = \begin{bmatrix} \frac{1}{\sigma_x} & 0 \\ 0 & \frac{1}{\sigma_y} \end{bmatrix}$$

(a)

$$H_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(b)

$$\Lambda'_i = \begin{bmatrix} 1/\sigma_x & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\sigma_y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad b'_i = \begin{bmatrix} z_{1t}/\sigma_x \\ z'_{1t}/\sigma_y \end{bmatrix}$$

(c)

Fig. 4. (a) For the example in Fig. 2, we assume measurements have covariance Σ_{ii} and a corresponding inverse Cholesky factor B_{ii}^{-1} . (b) H_i lists the rows that must be added to the sparse measurement matrix H in Fig. 3 to assign the clutter measurement one to target one. (c) Λ'_i and b'_i show the rows and rhs vector that must be added to the least squares problem to conduct this assignment.

edge proposal is based on the idea that, given a measurement and the set of targets assigned to that measurement, we can completely construct the rows corresponding to that measurement in the sparse measurement matrix H . We treat measurements that were clutter in the previous iteration of the Markov chain differently from measurements that were assigned to a target in the previous iteration.

5.3.1 Clutter Measurement

When we assign a clutter measurement, we add an edge to the bipartite graph representation of the data association J_t . As a consequence, we need to add rows to the measurement matrix H that map the assigned target's state X_{jt} to the assigned measurement Z_{it} . To construct the rows of H , we use the mapping detailed in Section 3.2. If we examine the system matrix, specifically, the structure of the inverse Cholesky factor B^{-1} , we see that we also must multiply the new rows by the inverse Cholesky factor B_{ii}^{-1} of the measurement covariance Σ_{ii} . For example, Fig. 4 shows the rows that must be added to the least squares problem (13) to assign the clutter measurement one to target one. The result of the updating operation provides us with a new R' factor and rhs vector z' . We accept this proposal with the following probability

$$a = \min \left(\frac{|Z'_{c,t}| \sqrt{|2\pi V'_t|} \sqrt{|2\pi \Sigma|} \exp(-\rho'^2)}{|Z_{c,t}| \sqrt{|2\pi V_t|} \sqrt{|2\pi \Sigma'|} \exp(-\rho^2)}, 1 \right). \quad (15)$$

5.3.2 Assigned Measurement

When a measurement is assigned to a target in the previous iteration of the Markov chain, we must construct the new rows Λ'_i and b'_i and down-date the old rows, Λ_i and b_i in the least squares problem (13). To construct these rows, we use the mapping described in Section 3.2. In this case as well, we must multiply the system matrix by the inverse Cholesky factor B_{ii}^{-1} of the measurement covariance. For instance, Fig. 5 shows the new rows required to assign measurement five to target two in the example shown Section 3.2. This proposal is accepted according to the probability

$$a = \min \left(\frac{\sqrt{|2\pi V'_t|} \sqrt{|2\pi \Sigma|} \exp(-\rho'^2)}{\sqrt{|2\pi V_t|} \sqrt{|2\pi \Sigma'|} \exp(-\rho^2)}, 1 \right).$$

$$\begin{aligned}
H'_i &= \begin{bmatrix} 0 & 0 & 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 & 0 \\ 0 & 0 & 0 & 1/3 & 0 & 0 & 0 & 1/3 & 0 & 1/3 \end{bmatrix} \\
&\quad (a) \\
H_i &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 0 & 1/2 \end{bmatrix} \\
&\quad (b) \\
\Lambda'_i &= \begin{bmatrix} 0 & 0 & \frac{1}{3\sigma_x} & 0 & 0 & 0 & \frac{1}{3\sigma_x} & 0 & \frac{1}{3\sigma_x} & 0 \\ 0 & 0 & 0 & \frac{1}{3\sigma_y} & 0 & 0 & 0 & \frac{1}{3\sigma_y} & 0 & \frac{1}{3\sigma_y} \end{bmatrix} \quad b'_i = \begin{bmatrix} z_{5t}/\sigma_x \\ z'_{5t}/\sigma_y \end{bmatrix} \\
&\quad (c) \\
\Lambda_i &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2\sigma_x} & 0 & \frac{1}{2\sigma_x} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2\sigma_y} & 0 & \frac{1}{2\sigma_y} \end{bmatrix} \quad b_i = \begin{bmatrix} z_{5t}/\sigma_x \\ z'_{5t}/\sigma_y \end{bmatrix} \\
&\quad (d)
\end{aligned}$$

Fig. 5. For the example in Fig. 2, (a) shows the the rows H'_i that must be added to the the measurement matrix H in Fig. 3 to assign measurement five to targets two, four, and five. (b) Before we add these rows, we remove the rows H_i that previously assigned measurement five to targets four and five. (c) Λ'_i and b'_i are the rows that must be added to the least squares problem (13). (d) Λ_i and b_i show the rows that must be downdated from the least squares problem. We assume the measurement covariance follows Fig. 4a.

This target ratio calculation is typically conducted in log-space to avoid overflow. When a measurement is no longer assigned a target, an edge is removed, we only downdate the least squares problem and accept according to (15).

6 ALGORITHM SUMMARY

Putting all these improvements together, we arrive at the following RBPF tracking algorithm:

1. Start with a set of S hybrid Gaussian samples

$$P(X_{t-1}|Z_{1:t-1}) \approx \left\{ \mathcal{N}\left(X_{t-1}; m_{t-1}^{(s)}, V_{t-1}^{(s)}\right) \right\}_{s=1}^S,$$

which approximate the posterior over the target state at the previous time step $P(X_{t-1}|Z_{1:t-1})$.

2. Apply the prediction equation (7) to obtain a hybrid sample approximation of the predictive prior $P(X_t|Z_{1:t-1})$

$$P(X_t|Z_{1:t-1}) \approx \left\{ \mathcal{N}\left(X_t; Am_{t-1}^{(s)}, Q_t^{(s)}\right) \right\}_{s=1}^S.$$

3. Run the Rao-Blackwellized Markov chain described in Section 5. After running for a given number of iterations, the Markov chain will return an upper-triangular R factor and rhs vector z . This process can be repeated W times to obtain the R factors and rhs vectors needed to form the new hybrid sample set in the next step.
4. Solve the upper triangular system $Rm_t = z$ and form the covariance matrix $V_t = R^{-T}R$ for each run of the Markov chain to obtain a new set of hybrid samples

$$P(X_t|Z_{1:t}) \approx \left\{ \mathcal{N}\left(X_t; m_t^{(w)}, V_t^{(w)}\right) \right\}_{w=1}^W.$$

The hybrid sample set approximates the posterior distribution over the current state $P(X_t|Z_{1:t})$.

6.1 Computational Heuristics

We apply two common heuristics to obtain some additional gains in efficiency. First, we gate the measurements based on a covariance ellipse around each target, a technique commonly used in radar tracking [33], [6]. Targets are only assigned to measurements within β standard deviations of the predicted position of the target. Second, we initialize the chain at each time step by assigning each measurement uar to a nearby target. A target is nearby if the measurement is gated for that target. The result of this procedure produces a more likely initial data association which also shortens the mixing time of the chain.

7 EXPERIMENTAL RESULTS

To show the effectiveness of the algorithm, we examined its performance on 1) a challenging simulation sequence and 2) on two real data sets—one using vision and one using a laser range finder—where ground truth data was available. All of the data sets and videos described below are available from <http://borg.cc.gatech.edu/biotracking/experimental-data.html>.

We tested three variants of the tracker. The first was restricted to matchings: one measurement per target and one target per measurement. The second allowed only for multiple measurements per target. The third allowed the sampler to propose both multiple measurements and merged measurements. We also examined the running time of the algorithm. The results and specific parameter choices are detailed in the section below.

7.1 Simulation

First, we tested the algorithm on a simulation sequence where two accelerating targets move in close proximity, see Fig. 6. The sequence represents a substantial challenge because both targets generate multiple measurements. When the target are in close proximity, distinguishing which target generated a particular measurement is quite difficult.

The success rates of each variant of the algorithm over a total of 500 simulation runs are summarized in Table 1. In the simulation, the field of view was set to $x = [0, 75] y = [-25, 25]$. The ground truth trajectory was obtained by generating the noiseless trajectory of targets that were initially placed at $x_0 = 0 y_0 = \pm 20.75$ with initial velocity and acceleration $v_{0,x} = 4.4 a_y = 0.5 v_y = \pm 4.2$ moving according to a constant acceleration model with a unit time step $\Delta t = 1$. The trajectory was $T = 18$ units in length. Next, measurements were generated at each position. The number of measurements was sampled from a Poisson distribution with mean $\lambda_x = 20$ and the positions of the measurements were sampled from a 2D Gaussian with variance $\sigma_z^2 = 9.0I_2$ centered at the target position. Additionally, clutter measurements were introduced uniformly over the field of view. The number of clutter measurements was sampled from a Poisson with mean $\lambda_c = 30$. If targets were off by more than 10 units at the last time step, the simulation sequence was marked a failure.

7.2 Interacting Targets

Next, we tested all three variants of the algorithm on a challenging ground truth sequence of 20 ants, *Aphaenogaster cockerelli*, in a small container. The analysis of behavioral trajectory data has important implications in the study of animal behavior [1], [25], [12], [11] and this particular

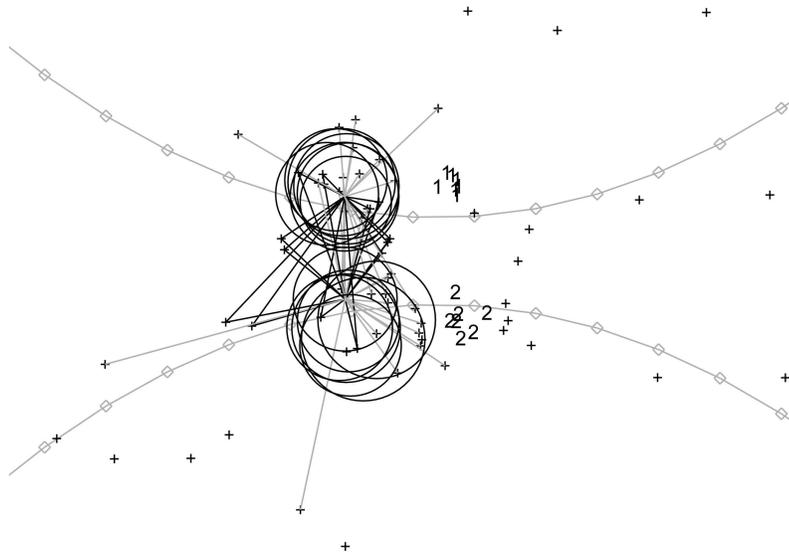


Fig. 6. This figure shows a frame from the simulation sequence. \diamond designates the ground truth. The circles designated the covariance ellipses of the hybrid samples. The data association for one sample is plotted. The dark lines and light gray lines designate merged measurements and multiple measurements, respectively.

sequence was obtained as part of a larger research project to analyze and model multiagent system behavior. 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, see Fig. 7.

The test sequence presents a substantial challenge for any multitarget tracking algorithm. The sequence consisted of 10,400 frames recorded at a resolution of 720×480 pixels at 30 Hz. A simple procedure where image pixels were

thresholded was used to obtain detections. The original images were blurred and down sampled twice to obtain an image that was 180×120 pixels. Pixels with the following YUV ranges were considered detections: $1 < Y < 75$, $122 < U < 128$, and $128 < V < 135$. The x, y locations on the smaller images were then scaled up to the original 720×480 image.

The number of failures detected on the ground truth sequence for all variants of the tracker are shown in Table 2. We counted a failure when a target deviated from the ground truth position by more than 60 pixels. After a failure, all of the targets were reinitialized to the ground truth position and tracking was resumed. In both the simulation and on interacting target data, we used the same parameter settings. The state space for each target included both position and velocity $X_i = [v_x, v_y, x, y]^T$. Measurements were simply a 2D position $Z = [x, y]^T$. In this specific arrangement, the measurement matrix was

$$H_{i,j} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

since the velocity components of the target state were not measured. We modeled target motion using a constant velocity model with time step $\Delta t = 0.033$. We used $S = 6$ hybrid samples. We ran the Markov chain for 333 iterations between each sample. The initial covariance was $V_0 = 32I_{4N}$. The predictive covariance was set to $\Gamma = 16I_{4N}$. The measurement noise was $\Sigma_{ii} = 32I_{4N}$. All of the covariances were

TABLE 1
Tracker Success Rates for 500 Simulation Runs

Data Association Type	Success Rate
Matchings Only	32.7%
Multiple Measurements	56.1%
Merged and Multiple Measurements	77.9%

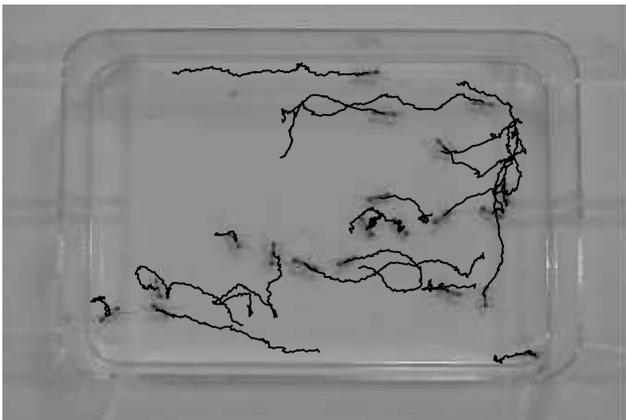


Fig. 7. A frame in the ground truth sequence of 20 ants. The black lines show a small portion of the ants' trajectories. In the video sequence, interactions occur frequently and can involve five or more ants in close proximity. See also Fig. 1.

TABLE 2
Number of Failures on 10,400 Frame Ground Truth Sequence

Data Association Type	Number of Failures
Matchings Only	40
Multiple Measurements	36
Merged and Multiple Measurements	21

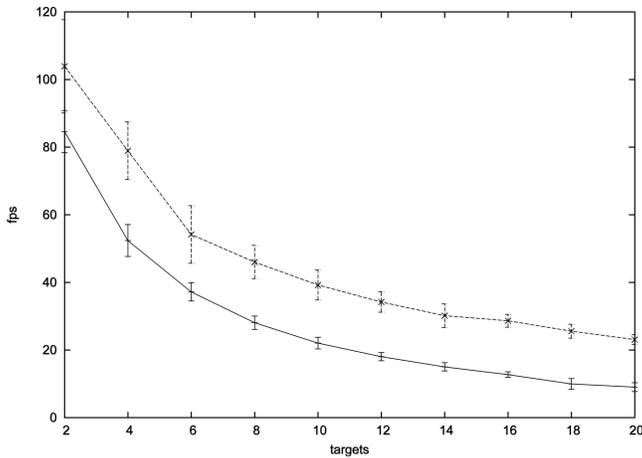


Fig. 8. Average frames per second (fps) versus the number of targets of an unoptimized implementation of the algorithm. The solid line shows performance for $S = 6$ samples and the dashed line shows performance for unimodal distribution $S = 1$. Error bars show one standard deviation from the mean. See text for details on the implementation.

specified in pixels. The gating threshold was $\beta = 3$ and off diagonal covariance blocks were zeros when targets were more than $\gamma = 200$ pixels apart. The auxiliary variable proposal was chosen with probability $p_s = 0.05$.

7.3 Run Time

To examine the run time of the algorithm we plotted the average frame rate on a Pentium 4-M 3 GHz versus the number of targets tracked on the first 200 frames of the test sequence. N nearby targets were chosen randomly. The frame rate did not include the time to process the image or update the display. A freely available sparse LDL factorization code was used to compute the Cholesky factorizations [8]. The average frame rate is shown in Fig. 8.

We also compared the average frame rate including image processing time to our previous work on the same video sequence. As shown in Table 3, the current work performs eight times faster for 20 targets when the hybrid sample set has $S = 6$ samples and has five fewer tracking failures on the ground truth sequence. When the distribution over the target state is unimodal $S = 1$, the algorithm performs 23 times faster at nearly the same accuracy. Note that our previous work additionally tracks target orientation. This additional

TABLE 3
Average Frames per Second (fps) Including Image Processing Time

Algorithm	fps	Number of Failures
Current Work $S = 1$	23.03 ± 0.87	24
Current Work $S = 6$	8.75 ± 0.55	21
[23]	1.04 ± 0.21	26

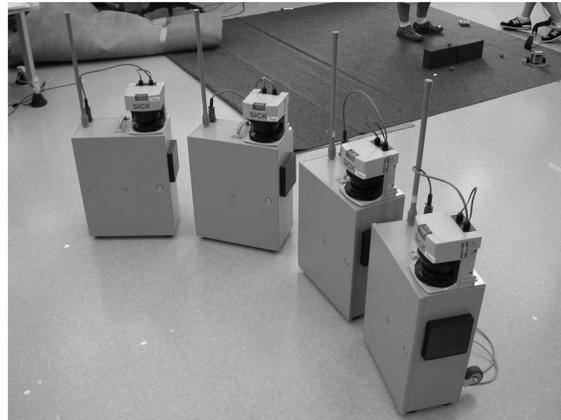


Fig. 9. The laser tracking system consists of four stationary laser range scanner units. The system is designed to track people and objects. In this second tracking domain, we test the algorithm’s robustness to missed detections.

complexity accounts partially for the slower frame rate and may also account for the extra failures [23].

7.4 Missed Detections

Further, to investigate the algorithm’s robustness to missed detections we applied it to a second sensor modality—SICK laser range data. In this case, the tracking system consists of four stationary laser range scanners to track people and objects. The four units are shown in Fig. 9. The system uses an Iterative Closest Point algorithm to align individual scans. Next, stationary background objects are removed using an background subtraction algorithm as shown in Fig. 10. The remaining measurements are used for multitarget tracking.

The final tracking system must be robust to missed detections. In this application, missed detections occur

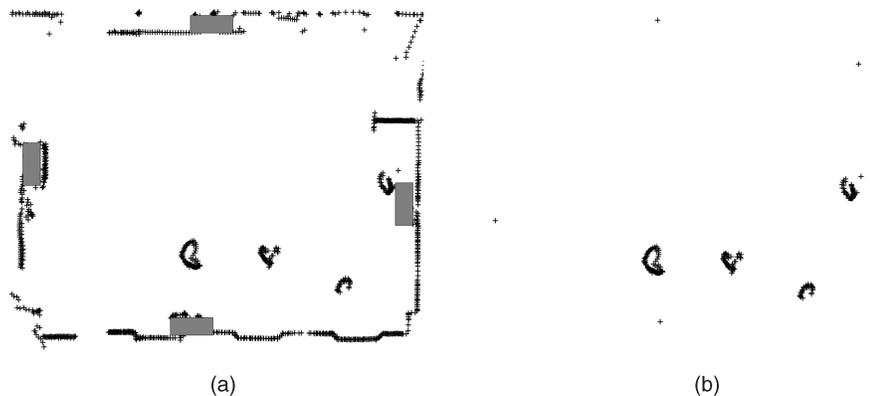


Fig. 10. (a) An Iterative Closest Point algorithm aligns individual laser scans from each of four lasers. The approximate positions of each laser unit are shown as dark gray rectangles. (b) The system uses a background subtraction algorithm to remove stationary objects from the environment.

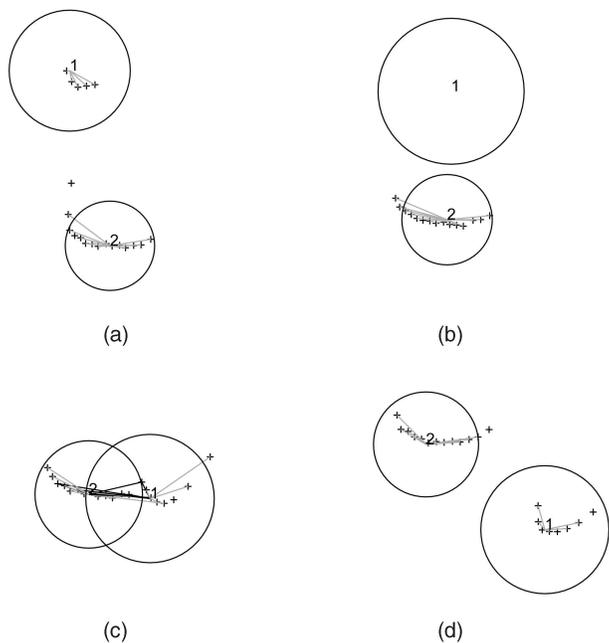


Fig. 11. This figure shows an example of tracking through an interaction and a series of missed detections using laser range data. The light gray and black lines designate multiple and merged data associations, respectively. The black circles show the covariance ellipse around the estimated target positions. Laser range measurements are designated by +.

when the laser is occluded by obstacles or other targets. An example of a missed detection sequence is shown in Fig. 11. It is important that these cases be addressed as the deployed version of the laser tracking system will have regions that are covered only by a single laser. These regions will be particularly prone to missed detections.

To test the algorithm's performance through missed detections we tested with a *single* laser, which presents an extreme case and is the toughest situation for the tracking system. We tracked $N = 5$ people with a single laser through 4,752 scans (specifically, laser ID 0 in the data

set). The five people were asked to periodically bump into each other to introduce close interactions.

The results were good. For the data set, 164 missed detections were counted by a human observer. Any cases where all of the measurements accounting for a single target disappeared for several scans were counted as missed detections. The system failed 15 times through the sequence for a success rate of 90.85 percent.

The details were as follows: we modeled target motion using a constant velocity model with time step $\Delta t = 0.03$. We used a unimodal distribution over the target state since the background subtraction algorithm removed most of the clutter $S = 1$. As a consequence, the sample proposal probability was set to $p_s = 0$. We ran the Markov chain for 4,000 iterations and took the last sample. The initial covariance was $V_0 = 0.05I_{4N}$. The predictive covariance was set to $\Gamma = 0.05I_{4N}$. The measurement noise was $\Sigma_{ii} = 0.1I_{4N}$. All of the covariances were specified in meters. The gating threshold was $\beta = 4.5$ and the clutter probability was $z_x \sim \text{Unif}(8.7)$ and $z_y \sim \text{Unif}(11.5)$ over the 8.7m by 11.5m dimension of the tracking area.

7.5 Occlusions

Finally, we also tested the algorithm's robustness through partial and full occlusions on a set of video sequences in which two targets move on separate planes. In the selected video sequences, a colony of *Leptothorax curvinoposis* ants is in the process of nest emigration. The artificial nest located in the field of view consists of a cavity constructed out of balsa wood and a top covering pane of glass. Ants may enter the nest in the entrance at the top of the image, but they may also walk on the glass covering of the cavity. As a result, there are two planes in which the ants may move—the top glass and the floor of the nest cavity. It is desirable that the system is robust to occlusions, partial and full, caused by ants moving over one another. An example of an occlusion is shown in Fig. 12a.

As before, a simple procedure where image pixels were thresholded was used to obtain detections. The original images were blurred and down sampled twice to obtain an image that was 180×120 pixels. Pixels with the following

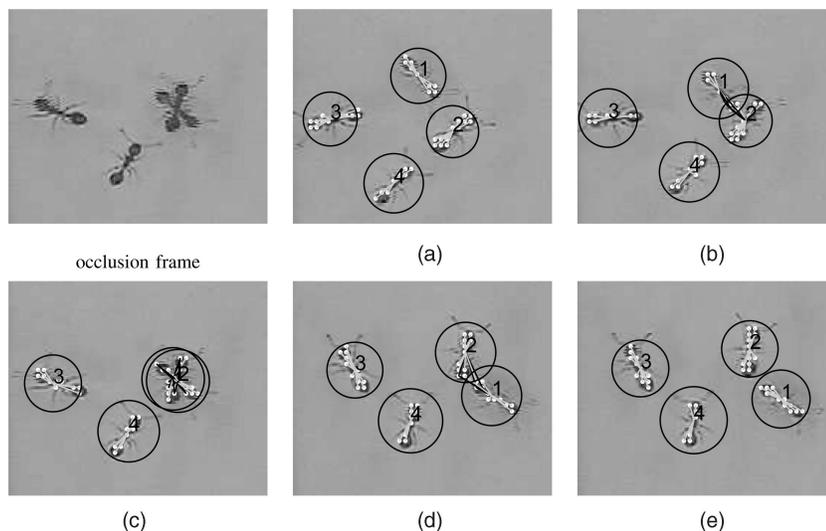


Fig. 12. The occlusion event that occurred in sequence 1. Here, two targets pass over one another creating an occlusion in (c). The tracker successfully tracks through this occlusion (a), (b), (c), (d), and (e). Measurements are designated by +. The light gray and black lines designate multiple and merged data associations, respectively. The black circles show the covariance ellipse around the estimated target positions.

YUV ranges were considered detections: $39 < Y < 101$, $116 < U < 125$, and $128 < V < 136$. The x, y locations on the smaller images were then scaled up to the original 720×480 image. We modeled target motion using a constant velocity model with time step $\Delta t = 0.1$. We used a unimodal distribution over the target state $S = 1$. We ran the Markov chain for 4,000 iterations and took the last sample. The initial covariance was $V_0 = 32I_{4N}$. The predictive covariance was set to $\Gamma = 4I_{4N}$. The measurement noise was $\Sigma_{ii} = 150I_{4N}$. All of the covariances were specified in pixels. The gating threshold was $\beta = 3$ and the clutter probability was uniform over the field of view, $z_x \sim \text{Unif}(0, 720)$ and $z_y \sim \text{Unif}(0, 480)$.

We tested the algorithm by randomly selecting 16 video clips of occlusion events. The algorithm successfully tracked through 12 of the 16 sequences. Specifically, the algorithm failed on sequences 5, 8, 12, and 14. All sequences, as well as the trajectories, are available on our Web site for viewing and testing.

8 CONCLUSIONS

The algorithm performs accurately on challenging simulation data and on real data sets with interacting targets. In simulation, modeling both merged and multiple measurements eliminates track switching during close interactions. As shown in Table 1, the algorithm performs best over the simulation runs when merged measurements are modeled. Table 2 shows that, on challenging video sequence with interacting targets, modeling merged measurements reduces the number of tracking failures that occur during interactions. The algorithm is effective in two sensor modalities—laser range data as well as video. We specifically show that the algorithm is robust missed detections in laser range data. In Section 7.5, we also show that modeling merged measurements effectively handles occlusions in challenging video sequences.

Moreover, the algorithm exhibits real time performance on a conventional PC. Fig. 8 shows an unoptimized implementation runs at approximately 23Hz for 20 targets for $S = 1$ a unimodal distribution over the target state and 10Hz for a more complex multimodal distribution $S = 6$. The performance is due to a compelling property of the algorithm: A measurement can be assigned to a target in time proportional to the number of targets interacting with that target squared. The algorithm has this property because the incremental least squares techniques exploit the sparsity of the R factor, which is maintained by zeroing off diagonal blocks of the covariance matrix when targets are no longer interacting.

Another, complementary approach to improving performance of the algorithm is the use of a data-driven proposal distribution [44], [38], [48]. By exploiting low-level information, such as edge and appearance information in images, the Markov chain could quickly explore parts of the target distribution that have high probability. An interesting direction for future work would be to utilize this information while maintaining the real time performance of the algorithm.

In addition to modifying the proposal distributions, we intend to explore how we can modify the inference algorithm to address tracking and data association over a sliding window. Performing data association over a longer period of time may help eliminate many of the failures observed in Table 2. Moreover, it might be possible to include more complex trajectory primitives such as curves

and lines that represent target maneuvers over time. The RB Markov chain would efficiently fit these primitives to the trajectory data in the least squares sense.

To the best of our knowledge, this work is the first time MCMC and sparse factorization updating and downdating techniques have been combined to obtain a practical, real time algorithm. We believe this general approach will yield a number of computationally efficient algorithms for addressing a large class of problems, in particular, linear least squares problems where correspondences are unknown.

ACKNOWLEDGMENTS

This work was funded under US National Science Foundation Award IIS-0219850, "ITR: Observing, Tracking and Modeling Social Multi-Agent Systems." Frank Dellaert was partially supported by US National Science Foundation CAREER grant, award number IIS-0448111, and Tucker Balch was partially supported by US National Science Foundation CAREER grant, award number IIS-0347743. The authors would like to thank the anonymous reviewers, Ananth Ranganathan and Eric Vigoda, for their insightful comments. The laser data was collected with the help of Adam Feldman and Jorge Lage Saguier. They would also like to thank Stephen Pratt for the *Leptothorax curvispinosus* video sequence.

REFERENCES

- [1] T. Balch, F. Dellaert, A. Feldman, A. Guillory, C. Isbell, Z. Khan, A. Stein, and H. Wilde, "How A.I. and Multi-Robot Systems Research Will Accelerate Our Understanding of Social Animal Behavior," *Proc. IEEE*, 2005.
- [2] K. Branson and S. Belongie, "Tracking Multiple Mouse Contours (without too Many Samples)," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 1039-1046, 2005.
- [3] G. Casella and C.P. Robert, "Rao-Blackwellisation of Sampling Schemes," *Biometrika*, vol. 83, no. 1, pp. 81-94, Mar. 1996.
- [4] J.A. Castellanos and J.D. Tardos, *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic, 2000.
- [5] Y.C. Chang and Y. Bar-Shalom, "Joint Probabilistic Data Association for Multitarget Tracking with Possibly Unresolved Measurements and Maneuvers," *IEEE Trans. Automatic Control*, vol. 29, pp. 585-594, 1984.
- [6] J.B. Collins and J.K. Uhlmann, "Efficient Gating in Data Association with Multivariate Gaussian Distributed States," *IEEE Trans. Aerospace and Electronic Systems*, vol. 28, no. 3, 1992.
- [7] I.J. Cox and S.L. Hingorani, "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and Its Evaluation for the Purpose of Visual Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, 138-150, Feb. 1996.
- [8] T.A. Davis, "Algorithm 8xx: A Concise Sparse Cholesky Factorization Package," Technical Report TR-04-001, Univ. of Florida, Jan. 2004.
- [9] F. Dellaert, S.M. Seitz, C.E. Thorpe, and S. Thrun, "Structure from Motion without Correspondence," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 2000.
- [10] J.J. Dongarra, C.B. Moler, J.R. Bunch, and G.W. Stewart, *LINPACK: Users' Guide*. Soc. Industrial and Applied Math., 1979.
- [11] M. Egerstedt, T. Balch, F. Dellaert, F. Delmotte, and Z. Khan, "What Are the Ants Doing? Vision-Based Tracking and Reconstruction of Control Programs," *Proc. IEEE Int'l Conf. Robotics and Automation*, Apr. 2005.
- [12] A. Feldman and T. Balch, "Representing Honey Bee Behavior for Recognition Using Human Trainable Models," *Adaptive Behavior*, 2004.
- [13] G. Gennari and G.D. Hager, "Probabilistic Data Association for Visual Tracking of Groups," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.
- [14] A. Genovesio and J.C. Olivo-Marin, "Split and Merge Data Association Filter for Dense Multi-Target Tracking," *Proc. Int'l Conf. Pattern Recognition*, pp. 677-680, 2004.

- [15] *Markov Chain Monte Carlo in Practice*, W.R. Gilks, S. Richardson, and D.J. Spiegelhalter, eds. Chapman and Hall, 1996.
- [16] G.H. Golub and C.F. Van Loan, *Matrix Computations*. Johns Hopkins Univ. Press, 1996.
- [17] I. Haritaoglu, D. Harwood, and L.S. Davis, "W4: Real-Time Surveillance of People and Their Activities," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 809-830, 2000.
- [18] M. Isard and A. Blake, "Contour Tracking by Stochastic Propagation of Conditional Density," *Proc. European Conf. Computer Vision*, pp. 343-356, 1996.
- [19] M. Isard and J. MacCormick, "BraMBLE: A Bayesian Multiple-Blob Tracker," *Proc. Int'l Conf. Computer Vision*, pp. 34-41, 2001.
- [20] M. Jerrum and A. Sinclair, "The Markov Chain Monte Carlo Method: An Approach to Approximate Counting and Integration," *Approximation Algorithms for NP-Hard Problems*, D.S. Hochbaum, ed., chapter 12. PWS Publishing, 1997.
- [21] T. Kanade, R. Collins, A. Lipton, P. Burt, and L. Wixson, "Advances in Cooperative Multi-Sensor Video Surveillance," *Proc. DARPA Image Understanding Workshop*, pp. 3-24, 1998.
- [22] Z. Khan, T. Balch, and F. Dellaert, "An MCMC-Based Particle Filter for Tracking Multiple Interacting Targets," *Proc. European Conf. Computer Vision*, 2004.
- [23] Z. Khan, T. Balch, and F. Dellaert, "MCMC-Based Particle Filtering for Tracking a Variable Number of Interacting Targets," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1805-1918, Nov. 2005.
- [24] Z. Khan, T. Balch, and F. Dellaert, "Multitarget Tracking with Split and Merged Measurements," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.
- [25] Z. Khan, R.A. Herman, K. Wallen, and T. Balch, "A 3-D Visual Tracking System for the Study of Spatial Navigation and Memory in Rhesus Monkeys," *Behavior Research Methods, Instruments & Computers*, 2005.
- [26] T. Kirubarajan, Y. Bar-Shalom, and K.R. Pattipati, "Multiassignment for Tracking a Large Number of Overlapping Objects [and Application to Fibroblast Cells]," *IEEE Trans. Aerospace and Electronic Systems*, vol. 37, no. 1, pp. 2-21, Jan. 2001.
- [27] W. Koch and G. van Keuk, "Multiple Hypothesis Track Maintenance with Possibly Unresolved Measurements," *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 589-601, 1997.
- [28] J. MacCormick and A. Blake, "A Probabilistic Exclusion Principle for Tracking Multiple Objects," *Proc. Int'l Conf. Computer Vision*, pp. 572-578, 1999.
- [29] M. Montemerlo and S. Thrun, "Simultaneous Localization and Mapping with Unknown Data Association Using FastSLAM," *Proc. IEEE Int'l Conf. Robotics and Automation*, 2003.
- [30] K. Murphy and S. Russell, "Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks," *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, eds., Springer-Verlag, Jan. 2001.
- [31] S. Oh, S. Russell, and S. Sastry, "Markov Chain Monte Carlo Data Association for General Multiple Target Tracking Problems," *Proc. 43rd IEEE Conf. Decision and Control*, 2004.
- [32] M.K. Pitt and N. Shephard, "Auxiliary Variable Based Particle Filters," *Sequential Monte Carlo Methods in Practice*. A. Doucet, N. de Freitas, and N. Gordon, eds. Springer-Verlag, 2001.
- [33] R. Popoli and S.S. Blackman, *Design and Analysis of Modern Tracking Systems*. Artech House Radar Library, Aug. 1999.
- [34] D. Ramanan and D.A. Forsyth, "Finding and Tracking People from the Bottom Up," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003.
- [35] C. Rasmussen and G.D. 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.
- [36] O. Sanchez and F. Dibos, "Displacement Following of Hidden Objects in a Video Sequence," *Int'l J. Computer Vision*, vol. 57, no. 2, pp. 91-105, 2004.
- [37] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers, "People Tracking with a Mobile Robot Using Sample-Based Joint Probabilistic Data Association Filters," *Int'l J. Robotics Research*, vol. 22, no. 2, 2003.
- [38] S.C. Zhu, R. Zhang, and Z. Tu, "Integrating Bottom-Up/Top-Down for Object Recognition by Data Driven Markov Chain Monte Carlo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000.

- [39] L. Sigal, S. Bhatia, S. Roth, M.J. Black, and M. Isard, "Tracking Loose-Limbed People," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 421-428, 2004.
- [40] K. Smith, D. Gatica-Perez, and J.M. Odobez, "Using Particles to Track Varying Numbers of Interacting People," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2005.
- [41] C. Stauffer and W.E.L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 246-252, 1999.
- [42] H. Tao, H.S. Sawhney, and R. Kumar, "Object Tracking with Bayesian Estimation of Dynamic Layer Representations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 75-89, Jan. 2002.
- [43] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte, "Simultaneous Localization and Mapping with Sparse Extended Information Filters," *Int'l J. Robotics Research*, vol. 23, nos. 7-8, pp. 693-716, 2004.
- [44] Z. Tu, S.-C. Zhu, and H.-Y. Shum, "Image Segmentation by Data Driven Markov Chain Monte Carlo," *Proc. Int'l Conf. Computer Vision*, 2001.
- [45] Z.W. Tu and S.C. Zhu, "Image Segmentation by Data-Driven Markov Chain Monte Carlo," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 657-673, May 2002.
- [46] J. Vermaak, A. Doucet, and P. Perez, "Maintaining Multi-Modality through Mixture Tracking," *Proc. Int'l Conf. Computer Vision*, 2003.
- [47] T. Yu and Y. Wu, "Collaborative Tracking of Multiple Targets," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.
- [48] T. Zhao and R. Nevatia, "Tracking Multiple Humans in Crowded Environment," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004.



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 from 2002 to 2004. He is currently an algorithms developer at Sarnoff Corporation in Princeton, New Jersey. His research interests include computer vision, machine learning, optimization, and problems at the intersection of computer science and biology.



Tucker Balch received the BS degree 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. He is an assistant professor of computing at the Georgia Institute of Technology, Atlanta. His research focuses on behavior-based control, learning and diversity in multirobot teams, 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, an MSc degree in computer science and engineering from Case Western Reserve University in 1995, and the equivalent of an 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 60 articles in journals and refereed conference proceedings, as well as several book chapters. He is a member of the IEEE.