Particle Filtering for Multitarget Detection and Tracking

Chris Kreucher[1,3], Mark Morelande[2], Keith Kastella[3], Alfred O. Hero III[1]
[1]Department EECS, The University of Michigan, Ann Arbor, MI
[2]The University of Melbourne, Melbourne, Australia
[3]General Dynamics Advanced Information Systems, Ann Arbor, MI
ckreuche@umich.edu, mrmore@ee.mu.oz.au, Keith.Kastella@gd-ais.com, hero@eecs.umich.edu

Abstract— This paper presents a particle filter approach to recursively estimating the joint multitarget probability density (JMPD) for the purposes of simultaneous multitarget detection and tracking. The JMPD is a conditional probability density that characterizes uncertainty in both target state and target number given the measurements. Estimation of the JMPD presents a formidable computational challenge due to the high dimensionality of the state space needed to explicitly model the correlations between target states and between target states and target number. We address this challenge with an importance density that is measurement directed and which adaptively factorizes the problem into a set of smaller sub-problems when possible. We demonstrate the algorithm on a set of real targets whose motion is taken from a set of military battle exercises.

TABLE OF CONTENTS

1 INTRODUCTION 1
2 MATHEMATICAL FORMULATION : THE JOINT MULTITARGET PROBABILITY DENSITY 2
3 NUMERICAL IMPLEMENTATION : THE PARTICLE FILTER IMPLEMENTATION OF JMPD 6
4 SIMULATION RESULTS 12
5 CONCLUSIONS 12
6 ACKNOWLEDGEMENTS 13

1. INTRODUCTION

This paper presents a unified approach to multitarget detection and tracking which is based on recursive estimation of the joint multitarget probability density (JMPD). The JMPD is a single probabilistic entity that captures uncertainty about the number of targets present in the surveillance region as well as their individual states. As a nonlinear filter, the JMPD technique does not require assumptions of linearity or Gaussianity in either the temporal or the measurement update. The implementation presented in this paper improves and expands those of [19][20] by addressing the detection and tracking problem simultaneously.

Since the JMPD is a very high dimensional entity, sophisticated numerical techniques are required for tractable implementation of this technique. We advocate a particle filtering approach, where each particle is a sample from the JMPD. Each particle therefore contains an estimate of both the number of targets and the states of each individual target. Taken as a collection, the particles represent the uncertainty in both target number and states of the individual targets. We use a particle proposal scheme that is measurement directed for the evolution of existing targets, the introduction of new targets, and removal of existing targets.

The multitarget tracking problem has been traditionally addressed with techniques such as multiple hypothesis tracking (MHT) and joint probabilistic data association (JPDA) [3][2][5]. Both techniques work by translating a measurement of the surveillance area into a set of detections by thresholding. The detections are then either associated with existing tracks, used to create new tracks, or deemed false alarms. Typically, Kalman-filter type algorithms are used to update the existing tracks with the new measurements after association. The challenge, of course, is to determine the correct association between measurements and targets.

Others have approached the problem from a fully Bayesian perspective. Stone [33] develops a mathematical theory of multiple target tracking from a Bayesian point of view. Srivistava, Miller [22], and Kastella [18] also did early work in this area. These methods have the advantage that they are able to handle pre-thresholded measurements, as well as arbitrary models of kinematics.

Recently, some researchers have applied particle filter based strategies to the problem of multitarget tracking. In [11], Hue and Le Cadre use a particle filter based on the probabilistic multiple hypothesis tracker (PMHT) introduced by Streit [36]. Considerable attention is given to dealing with the measurement to target association issue. Others have done work which amounts to a blend between JPDA and particle filtering [14][6].

The BraMBLe [12] system, the independent partition particle filter (IPPF) of Orton and Fitzgerald [30] and the work of Maskell [27] consider multitarget tracking via particle filtering from a purely Bayesian perspective. Measurement-to-target association is not done explicitly; it is implicit within the Bayesian framework. This work has focussed on a tractable implementation of ideas in [33].

0-7803-8870-4/05/$20.00 (c) 2005 IEEE
IEEEAC paper # 1440

Mahler [23][26][29] has developed an approach to multitarget tracking based on random sets called "finite-set statistics" (FISST). Recently, FISST has been combined with particle filtering methods for multitarget tracking [37][34]. To date, these implementations have been limited to small scale problems for computational reasons. While developed independently ([17][18][18]), the JMPD method can be derived using the mathematics of random sets and expressed in the FISST framework (see [29]). As discussed there, JMPD can be traced back to the Event-Averaged Maximum Likelihood Estimation (EAMLE) work of Kastella [17][18] and many earlier works, e.g. [13][28][32].

The main contribution of this paper is the description of a multiple target tracker that recursively estimates the entire joint multitarget probability density using particle filtering methods for simultaneous multitarget detection and tracking. As such it as an extension and improvement to the work in [19][20]. The particle filter implementation uses a measurement directed importance density that includes both persistent targets (targets that remain from time step to time step) and target addition/removal. For persistent targets, we build on the independent partition idea of [30]. For target addition and removal, the importance density uses a measurement directed method of deciding which areas of the surveillance region are more likely to have had a target arrive or leave.

The rest of this paper is organized in the following manner. In Section 2, we introduce the notion of the joint multitarget probability density and show how the rules of Bayesian Filtering are applied to produce a recursive filtering procedure. We give the particle filter based estimation of the JMPD in Section 3. We detail therein the adaptive sampling strategy applied to automatically factor the JMPD when targets are behaving independently, while appropriately handling targets that are coupled. This automatic factorization is key to the computational tractability of this implementation. In Section 4, we give simulation results detailing the performance of the particle filter based multitarget tracker. Finally, we conclude in Section 5 with a brief summary and discussion.

2. MATHEMATICAL FORMULATION : THE JOINT MULTITARGET PROBABILITY DENSITY

In this section, we give the details of a Bayesian method of multitarget tracking predicated on recursive estimation of the Joint Multitarget Probability Density (JMPD). Others have studied Bayesian methods for tracking multiple targets [33][22][4]. We provide here a comprehensive framework for multitarget detection, tracking, and identification that includes unknown and time varying target number followed by a tractable implementation strategy based on particle filtering.

Mahler [26][24][25] advocates a related approach based on random sets which he calls "finite-set statistics" (FISST). Since FISST and the JMPD approach attack some of the same problems, many of the concepts that appear here such as multitarget motion models and multitarget measurement models also appear in the work of Mahler et. al [24]. The JMPD technique does not require the random set formalism of FISST; in particular, in contrast to the random set approach, the JMPD technique adopts the view that likelihoods and the joint multitarget probability density are conventional Bayesian objects to be manipulated by the usual rules of probability and statistics. Therefore, the JMPD approach described here makes no appeal to random sets or related concepts such as Radon-Nikodym derivatives.

The concept of JMPD was discussed in [25][16][15], where a method of tracking multiple targets that moved between discrete cells on a line was presented. We generalize the discussion here to deal with targets that have N-dimensional continuous valued state vectors and arbitrary kinematics. In many of the model problems, we are interested in tracking the position (x, y) and velocity $(\dot{x}, \dot{y})$ of multiple targets and so we describe each target by the four dimensional state vector $[x, \dot{x}, y, \dot{y}]'$. If we were interested in estimating target mode or target identification, these states would be added to the feature vector.

A simple schematic showing three targets (Targets A, B, and C) moving through a surveillance area is given in Figure 1. There are two target crossings, a challenging scenario for multitarget trackers.

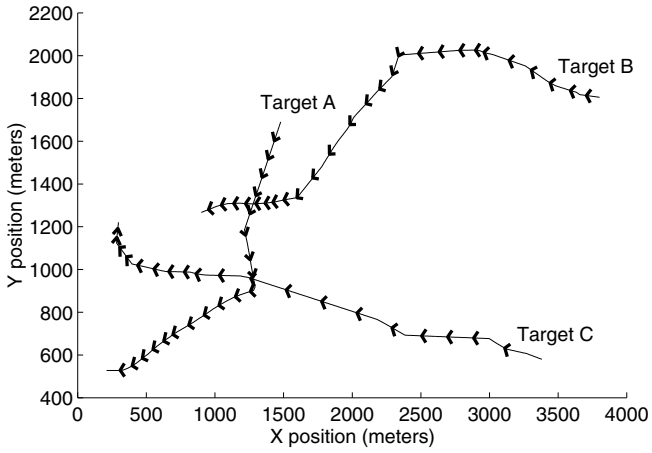


Figure 1. A schematic showing the motion of three of the ten targets in the simulation scenario, which is based on real collected target trajectories. The target paths are indicated by the lines, and direction of travel by the arrows. There are two instances where the target paths cross (i.e. are at the same position at the same time).

Recursive estimation of the JMPD provides a means for tracking an unknown number of targets in a Bayesian setting. The statistical model employed uses the joint multitarget conditional probability density

$$p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k | \mathbf{Z}^k) \quad (1)$$

as the probability density for exactly T targets with states

$\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{T-1}, \mathbf{x}_T$ at time k based on a set of past observations $\mathbf{Z}^k$. The number of targets T is a variable to be estimated simultaneously with the states of the T targets. The observation set $\mathbf{Z}^k$ refers to the collection of measurements up to and including at time k, i.e. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, ...\mathbf{z}^k\}$, where each of the $\mathbf{z}^i$ may be a single measurement or a vector of measurements made at time i.

Each of the state vectors $\mathbf{x}_t$ in the joint multitarget probability density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k|\mathbf{Z}^k)$ is a vector quantity and may (for example) be of the form $[x, \dot{x}, y, \dot{y}]$. We refer to each of the T target state vectors $\mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_{T-1}, \mathbf{x}_T$ as a partition of the multitarget state $\mathbf{X}$. For convenience, the density will be written more compactly in the traditional manner as $p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$, which implies that the state-vector $\mathbf{X}$ represents a variable number of targets each possessing their own state vector. We will often drop the time superscript k for notational simplicity when no confusion will arise.

As an illustration, some examples illustrating the sample space of p are

- $p(\emptyset, T = 0|\mathbf{Z})$, the posterior probability density for no targets in the surveillance volume
- $p(\mathbf{x}_1, T = 1|\mathbf{Z})$, the posterior probability density for one target with state $\mathbf{x}_1$
- $p(\mathbf{x}_1, \mathbf{x}_2, T = 2|\mathbf{Z})$, the posterior probability density for two targets with states $\mathbf{x}_1$ and $\mathbf{x}_2$

Likelihoods such as $p(\mathbf{z}|\mathbf{X}, T)$ and the joint multitarget probability density $p(\mathbf{X}, T|\mathbf{Z})$ are conventional Bayesian objects manipulated by the usual rules of probability and statistics. Thus, a multitarget system has state $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_T)$ with probability distribution $p(\mathbf{x}_1, \cdots, \mathbf{x}_T, T|\mathbf{Z})$. This can be viewed as a hybrid stochastic system where the discrete random variable T governs the dimensionality of $\mathbf{X}$. Therefore the normalization condition that the JMPD must satisfy is

$$\sum_{T=0}^{\infty} \int d\mathbf{x}_1 \cdots d\mathbf{x}_T p(\mathbf{x}_1, \cdots, \mathbf{x}_T, T|\mathbf{Z}) = 1 \ . \qquad (2)$$

Quantities of interest can be deduced from the JMPD. For example, the probability that there are exactly T targets present in the system is given by the marginal distribution

$$p(T|\mathbf{Z}) = \int d\mathbf{x}_1 \cdots d\mathbf{x}_T p(\mathbf{x}_1, \cdots, \mathbf{x}_T, T|\mathbf{Z}) \ . \qquad (3)$$

An important factor that is often overlooked in multitarget tracking algorithms is that the JMPD is symmetric under permutation of the target indices. This symmetry is a fundamental property of the JMPD which exists because of the physics of the problem and not because of mathematical construction. Specifically, the multitarget states $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2]$ and $\mathbf{X} = [\mathbf{x}_2, \mathbf{x}_1]$ refer to the same event, namely that there are two targets in the surveillance area – one with state $\mathbf{x}_1$ and one with state $\mathbf{x}_2$. This is true regardless of the makeup of the single target state vector. For example, the single target state vector may include target ID or even a target serial number and the permutation symmetry remains. Therefore, all algorithms designed to implement the JMPD are permutation invariant.

If targets are widely separated in the sensor's measurement space, each target's measurements can be uniquely associated with it, and the joint multitarget posterior density approximately factors. In this case, the problem may be treated as a collection of single target problems. The characterizing feature of multitarget tracking is that in general some of the measurements have ambiguous associations, and therefore the conditional density does not factor.

The temporal update of the posterior likelihood proceeds according to the usual rules of Bayesian filtering. The model of how the JMPD evolves over time is given by $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$ and will be referred to as the kinematic prior (KP). The kinematic prior includes models of target motion, target birth and death, and any additional prior information that may exist such as terrain and roadway maps. In the case where target identification is part of the state being estimated, different kinematic models may be used for different target types.

The time-updated (prediction) density is computed via the *model update* equation as given by eq. 4. The *measurement update* equation uses Bayes' rule to update the posterior density with a new measurement $\mathbf{z}^k$ as given by eq. 5.

This formulation allows JMPD to avoid altogether the problem of measurement to track association. There is no need to identify which target is associated with which measurement because the Bayesian framework keeps track of the entire joint multitarget density. In fact, there is no need for thresholded measurements (detections) to be used at all. A tractable sensor model merely requires the ability to compute the likelihood $p(\mathbf{z}|\mathbf{X}, T)$ for each measurement $\mathbf{z}$ received.

Kinematic Modelling : The Model $p(\boldsymbol{X}^k, T^k|\boldsymbol{X}^{k-1}, T^{k-1})$

The Bayesian framework outlined above requires a model of how the system state evolves, $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$. This includes both how the number of targets changes with time (i.e. T^k versus T^{k-1}) and how individual targets that persist over time evolve (i.e. $\mathbf{x}^k$ versus $\mathbf{x}^{k-1}$). In general, this model is chosen using the physics of the particular system under consideration. The simulation studies used herein come from a set of real ground targets recorded during a military battle exercise. Therefore, we specialize the models here to this application.

To specify the model, we need to generate an expression for $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$ which can be evaluated for any set of multitarget states and target counts $\{\mathbf{X}^k, T^k, \mathbf{X}^{k-1}, T^{k-1}\}$.

We first define a set of spatially varying priors on target arrival and death. Let $\alpha^k(\mathbf{x})$ denote the *a priori* probability

$$p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}) = \sum_{T^{k-1}=0}^{\infty} \int_{\mathbf{X}} d\mathbf{X}^{k-1} p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1}) p(\mathbf{X}^{k-1}, T^{k-1}|\mathbf{Z}^{k-1}) \tag{4}$$

$$p(\mathbf{X}^k, T^k|\mathbf{Z}^k) = \frac{p(\mathbf{z}^k|\mathbf{X}^k, T^k) p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})}{p(\mathbf{z}^k|\mathbf{Z}^{k-1})} \tag{5}$$

that a target will arrive at location $\mathbf{x}$ at time k. Similarly, let the *a priori* probability that a target in location $\mathbf{x}$ will die be denoted by $\beta^k(\mathbf{x})$. For the simulation studies, we assume temporally and spatially invariant (i.e. constant) rates for the birth and death and specify only α and β. The more general case is straightforward to implement, however it significantly complicates the notation.

For targets that persist over a time step, we model the target motion as linear and independent for each target, i.e. if the state of a target is given by $\mathbf{x} = (x, \dot{x}, y, \dot{y})$ the model is

$$\mathbf{x}_i^k = \mathbf{F}\mathbf{x}_i^{k-1} + \mathbf{w}_i^k \ , \tag{6}$$

where

$$\mathbf{F} = \begin{pmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{pmatrix} \ . \tag{7}$$

$\mathbf{w}_i^k$ is 0-mean Gaussian noise with covariance $\mathbf{Q} = diag(20, .2, 20, .2)$, which was selected based on an empirical fit to the data. In the cases where we are interested in tracking the mode or identification of targets, the state vector and $\mathbf{F}$ and $\mathbf{Q}$ matrices are augmented appropriately.

We emphasize here that Linear/Gaussian models are not a requirement of the formulation, but are used as they have been found to perform well in simulation studies with the real data. More complicated models of target motion can be inserted where appropriate without directly effecting computations in the algorithm.

Sensor Modeling : The Model $p(\mathbf{z^k}|\mathbf{X^k}, \mathbf{T^k})$

In order to implement Bayes′ Formula (eq. 5), we must compute the measurement likelihood $p(\mathbf{z}|\mathbf{X}, T)$. There are two approaches to modeling the likelihood, which we refer to as the "associated measurement" model and the "association-free" model. In both models, the sensor produces a sequence of scans at discrete instants in time. Each scan is a set of measurements produced at the same instant. The difference between the models lies in the structure of the scans.

In the associated measurement model, an observation vector consists of M measurements, denoted $\mathbf{z} = (z_1, \ldots, z_M)$. $\mathbf{z}$ is composed of threshold exceedances, i.e. valid detections and false alarms. Each valid measurement is generated by a single target and is related (possibly non-linearly) to the target state. False alarms have a known distribution independent of the targets (usually taken as uniform over the observation space) and the targets have known detection probability P_d (usually constant for all targets). The origin of each measurement is unknown. If measurement m is generated by target t, then it is a realization of a measurement random process H, i.e. $z_m \sim H_t(\mathbf{x}_t, w_t)$.

In its usual formulation, the associated measurement model precludes the possibility of two different targets contributing to a single measurement. This model predominates most current tracking, data fusion and sensor management work. The practical advantage of this model is that it breaks the tracking problem into two disjoint sub-problems: data association and filtering. The filtering problem is usually treated using some kind of Kalman filter. The disadvantages are a restricted sensor model and the difficult combinatorial problem of associating observations to filters. The associated measurement model was initially conceived in order to cast the problem into a form in which the Kalman filter can be applied, which is understandable in light of the enormous success the Kalman filter has enjoyed.

In contrast, nonlinear filtering methods allow much greater flexibility regarding the way measurements are modeled. As a result, we are free to employ an association-free sensor model in the work presented here. This type of model has been used in track-before-detect algorithms, in the "Unified Data Fusion" work of Stone et. al [33] and in the grid-based sensor management work of [16]. There are several advantages to the association-free method. First, it requires less idealization of the sensor physics and can readily accommodate issues such as merged measurements, side-lobe interference amongst targets and velocity aliasing. Second, it eliminates the combinatorial bottleneck of the associated-measurement approach. Finally, it simplifies the processing of unthresholded measurements to enable improved tracking at lower target SNR.

As motivation, we consider a few of the sensor types encountered in tracking and surveillance applications. First, an imaging sensor may observe a collection of unresolved point objects. The imager returns a collection of 1- or 2-dimensional pixel intensities. The output of each pixel is related to the integrated photon count in that pixel which is in turn determined by the background rate and how many targets are present within the pixel during the integration interval, and their locations within the pixel. This is represented numerically as either a positive integer or real number. Depending on the nature of the optics and their impulse response function, one or more pixels may respond to a target. Furthermore, multiple

targets can contribute to the output of a single pixel, violating the assumptions of the associated measurement model.

Another commonly used sensor type is radar. In a ground moving target indicator (GMTI) radar, a collection of pulses is emitted, their returns are collected and integrated over some coherent processing interval (CPI) [35]. The output of successive CPIs may also be averaged non-coherently. During the integration interval, the radar antenna is directed at some fixed or slowly varying bearing. The integrated pulse data is processed to obtain the reflectivity as a function of range and range-rate at that average bearing. Depending on the nature of the integration process, the return amplitude may be envelope detected or it may be available in complex form. Given the ubiquity of modern digital signal processing, radar data is usually available somewhere within the radar system as an array indexed by discrete range, range-rate and bearing values.

With this as background motivation, we present the association-free model. We compute the measurement likelihood $p(\mathbf{z}|\mathbf{X},T)$, which describes how sensor output depends on the state of all of the targets in the surveillance region. A sensor scan consists of M pixels, and a measurement $\mathbf{z}$ consists of the pixel output vector $\mathbf{z}=[z_1,\ldots,z_M]$, where z_i is the output of pixel i. In general, z_i can be an integer, real, or complex valued scalar, a vector or even a matrix, depending on the sensor. If the data are thresholded, then each z_i will be either a 0 or 1. Note that for thresholded data, $\mathbf{z}$ consists of both threshold exceedances and non-exceedances. The failure to detect a target at a given location can have as great an impact on the posterior distribution as a detection.

We model pixel measurements as conditionally independent so

$$p(\mathbf{z}|\mathbf{X},T)=\prod_i p(z_i|\mathbf{X},T) \tag{8}$$

Independence between the measurements given the state is often approximately true, and modeling as such often provides a nice simplification. However, conditional independence amongst the measurements is not a necessary part of this framework. Occasions where the physics of the situation imply sensor returns are dependent warrant a more general sensor model. This will not change the framework given here, only the implementation of the likelihood $p(\mathbf{z}|\mathbf{X},T)$.

Let $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_T]$ and let $\chi_i(\mathbf{x}_t)$ denote the indicator function for pixel i, defined as $\chi_i(\mathbf{x}_t)=1$ when a target in state $\mathbf{x}_t$ projects into sensor pixel i (i.e. couples to pixel i) and $\chi_i(\mathbf{x}_t)=0$ when it does not. Observe a pixel can couple to multiple targets and single target can contribute to the output of multiple pixels, say, by coupling through side-lobe responses. The indicator function for the joint multitarget state is constructed as the logical disjunction

$$\chi_i(\mathbf{X},T)=\bigvee_{t=1}^{T}\chi_i(\mathbf{x}_t) \tag{9}$$

The set of pixels that couple to $\mathbf{X}$ is

$$i_{\mathbf{X}}=\{i|\chi_i(\mathbf{X},T)=1\} \tag{10}$$

For the pixels that do not couple to $\mathbf{X}$, the measurements are characterized by the background distribution, denoted $p_0(z_i)$. With this, eq. (8) becomes

$$p(\mathbf{z}|\mathbf{X},T)= \prod_{i\in i_{\mathbf{X}}}p(z_i|\mathbf{X},T)\prod_{i\notin i_{\mathbf{X}}}p_0(z_i)\propto\prod_{i\in i_{\mathbf{X}}}\frac{p(z_i|\mathbf{X},T)}{p_0(z_i)} \tag{11}$$

Equation (12) allows for fairly general modeling of a pixelized sensor response to a collection of targets including non-linear effects due to multiple targets contributing to a single pixel. One limitation is aggregations of targets only couple to the union of pixels that the individual targets couple to. If a pair of targets have some type of nonlinear coupling that results in a contribution to a pixel that they do not couple to individually, this is not captured in the model. This is likely to be a very small effect in most situations, so we choose to ignore it here.

We further idealize the sensor as having a box-car resolution cell in position coordinates. We assume that the sensor scans a fixed rectangular region consisting of $N_x\times N_y$ contiguous pixels. The x- and y- ground-plane projection of each pixel is Δ_x and Δ_y. The sensor response within pixel i is uniform for targets in i and vanishes for targets outside pixel i. It is convenient to define the occupation number $n_i(\mathbf{X})$ for pixel i as the number of targets in $\mathbf{X}$ that lie in i. The single target signal-noise-ratio (SNR), assumed constant across all targets, is denoted λ. We assume that when multiple targets lie within the same pixel their amplitudes add non-coherently (this will be an accurate model for unresolved optical targets and radar targets not moving as a rigid body). Then the effective SNR when there are n targets in a pixel is $\lambda_n=n\lambda$ and we may use $p_n(z_i)$ to denote the pixel measurement distribution (note that the background distribution is obtained by setting $n=0$).

With these modeling assumptions, the measurement distribution in pixel i depends only on its occupation number and eq. (12) becomes

$$p(\mathbf{z}|\mathbf{X},T)\propto\prod_{i\in i_{\mathbf{X}}}\frac{p_{n_i(\mathbf{X}),T}(z_i)}{p_0(z_i)} \tag{12}$$

To complete the specification of the sensor model, we must give its dependence on SNR. Many models are plausible, depending on the detailed nature of the sensor. We have elected to use Rayleigh-distributed measurements. This distribution corresponds to envelope detected signals under a Gaussian model, and has been used for example to model interfering targets in a monopulse radar system and to model clutter and target returns in turbulent environments. Rayleigh models are also often used for diffuse fading channels. In the unthresholded case, this is

$$p_n(z)=\frac{z}{1+n\lambda}\exp\left(-\frac{z^2}{2(1+n\lambda)}\right) \tag{13}$$

When the tracker only has access only to thresholded measurements, we use a constant false-alarm rate (CFAR) model for the sensor. If the background false alarm rate is set at P_f, then the detection probability when there are n targets in a pixel is

$$P_{d,n} = P_f^{\frac{1}{1+n\lambda}} \tag{14}$$

This extends the usual relation $P_d = P_f^{\frac{1}{1+\lambda}}$ for thresholded Rayleigh random variables at SNR λ [2].

3. NUMERICAL IMPLEMENTATION : THE PARTICLE FILTER IMPLEMENTATION OF JMPD

We now turn to the development of a particle filter approximation to the Joint Multitarget Probability Density (JMPD). Even for modest problems, the sample space of the JMPD is large since it contains all possible configurations of state vectors $\mathbf{x}_t$ for all possible values of T. Earlier implementations of JMPD given by Kastella [16] approximated the density by discretizing on a grid. The computational burden in this scenario makes evaluating realistic problems intractable, even when using the simple model of targets moving between discrete locations in one-dimension. In fact, for a fixed approximation error, the number grid cells needed grows as L^T, where L is the number of discrete locations the targets may occupy and T is the number of targets.

Thus, to estimate the JMPD in a computationally tractable manner, a more sophisticated approximation method is required. We find that a particle filter (PF) based implementation of JMPD breaks the computational logjam and allows us to investigate more realistic problems.

The Single Target Particle Filter

Before detailing the particle filter implementation of JMPD, we first review standard single target particle filtering. Particle filtering is a method of approximately solving the prediction and update equations by simulation [1][9], where samples from the target density are used to represent the density and are propagated through time.

To implement a single target particle filter, the single target density of interest, $p(\mathbf{x}|\mathbf{Z})$, is approximated by a set of N_{part} weighted samples (particles):

$$p(\mathbf{x}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta_D(\mathbf{x} - \mathbf{x}_p) \tag{15}$$

Where δ_D represents the usual Dirac delta function.

The model update eq. (4) and the measurement update eq. (5) are simulated by the following three step recursion, summarized in Table 1.

First, the particle locations at time k are generated using the particle locations $\mathbf{x}_p$ at time $k-1$ and the current measurements $\mathbf{z}^k$ by sampling from an importance density, denoted $q(\mathbf{x}^k|\mathbf{x}^{k-1}, \mathbf{z}^k)$. The design of the importance density is a well studied area [7], as the choice of the importance density can have a dramatic effect of the efficiency of the particle filter algorithm. It is known that the optimal importance density (OID) is given by $p(\mathbf{x}^k|\mathbf{x}^{k-1}, \mathbf{z}^k)$, but this density is typically prohibitively difficult to sample from. In practice, oftentimes the importance density is chosen just to be the kinematic prior $p(\mathbf{x}^k|\mathbf{x}^{k-1})$. However, more sophisticated choices of importance density lead to better results for a fixed number of particles. As we will see in the multitarget case, approximating the OID (rather than simply using the kinematic prior) becomes crucial as problem dimension increases.

Second, particle weights are updated according to the weight equation, which involves the likelihood, the kinematic model, and the importance density [1].

$$w_p^k = w_p^{k-1} \frac{p(\mathbf{z}^k|\mathbf{x}_p^k) p(\mathbf{x}_p^k|\mathbf{x}_p^{k-1})}{q(\mathbf{x}_p^k|\mathbf{x}_p^{k-1}, \mathbf{z}^k)} \tag{16}$$

When using the kinematic prior as the importance density, the weight equation reduces to simply $w_p^k = w_p^{k-1} * p(\mathbf{z}^k|\mathbf{x}_p^k)$.

Finally, a resampling step is used to prevent particle degeneracy. Without resampling, the variance of the particle weights increases with time, yielding a single particle with all the weight after a small number of iterations [8]. Resampling may be done on a fixed schedule or based on the weight variance.

The particle filter algorithm that uses the kinematic prior as the importance density and resamples at each time step is called sampling importance resampling (SIR) in the literature. The algorithm is illustrated in Table 1.

Table 1. SIR Single Target Particle Filter

1. For each particle p, $p = 1, \cdots, N_{part}$, sample $\mathbf{x}_p^k \sim q(\mathbf{x}^k|\mathbf{x}^{k-1}, \mathbf{z}^k) = p(\mathbf{x}|\mathbf{x}_p^{k-1})$
2. Compute $w_p^k = w_p^{k-1} * p(\mathbf{z}|\mathbf{x}_p)$ for each p
3. Normalize w_p to sum to 1, $w_p \leftarrow w_p / \sum_{i=1}^{N_{parts}} w_i$.
4. Resample N_{part} particles with replacement from $\mathbf{x}_p$ based on the distribution defined by w_p

SIR Multitarget Particle Filtering

To implement the JMPD recursions via a particle filter, we similarly approximate the joint multitarget probability density $p(\mathbf{X}, T|\mathbf{Z})$ by a set of N_{part} weighted samples. A particle now becomes more than just the estimate of the state of a target; it incorporates both an estimate of the states of all of the targets as well as an estimate of the number of targets.

As we write the multitarget state vector for T targets as

$$\mathbf{X} = [\mathbf{x}_1, \ \mathbf{x}_2, \ \ldots, \ \mathbf{x}_{T-1}, \ \mathbf{x}_T] \ , \tag{17}$$

the particle state vector will be written as

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \ \mathbf{x}_{p,2}, \ \ \ldots \ \ \mathbf{x}_{p,T_p}] \ . \tag{18}$$

The notation we use here is that a particle $\mathbf{X}_p$ has T_p targets, where T_p can be any non-negative integer. With δ_D denoting the Dirac delta, we define

$$\delta(\mathbf{X} - \mathbf{X}_p) = \begin{cases} 0 & T \neq T_p \\ \delta_D(\mathbf{X} - \mathbf{X}_p) & \text{otherwise} \end{cases} \quad (19)$$

Then the particle filter approximation to the JMPD is given by a set of particles X_p and corresponding weights w_p as

$$p(\mathbf{X}, T|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \quad (20)$$

where $\sum_{p=1}^{N_{parts}} w_p = 1$.

The joint multitarget probability density $p(\mathbf{X}, T|\mathbf{Z})$ is defined for all possible numbers of targets, $T = 0, 1, 2, \cdots$. As each of the particles $\mathbf{X}_p$, $p = 1...N_{part}$ is a sample drawn from the JMPD $p(\mathbf{X}, T|\mathbf{Z})$, a particle $\mathbf{X}_p$ may have $0, 1, 2, \cdots$ partitions, each partition corresponding to a different target. Different particles in the approximation may correspond to different estimates of the number of targets in the surveillance region. In practice, the maximum number of targets a particle may track is truncated at some large finite number T_{max}.

We will denote the t^{th} partition of particle p by $\mathbf{X}_{p,t}$, i.e. $\mathbf{X}_{p,t}$ refers to the estimate of the t^{th} target state made by particle p. Since a partition corresponds to a target, the number of partitions that a particle has is that particle's estimate of the number of targets in the surveillance area.

With these definitions, the SIR particle filter extends directly to JMPD filtering, as shown in Table 2. This method simply proposes new particles at time k using the particles at time $k-1$ and the target kinematic model eq. (4) and the weight update becomes

$$w_p^k = w_p^{k-1} \frac{p(\mathbf{z}^k|\mathbf{X}_p^k) p(\mathbf{X}_p^k|\mathbf{X}_p^{k-1})}{q(\mathbf{X}_p^k|\mathbf{X}_p^{k-1}, \mathbf{z}^k)} \quad (21)$$

As in the single target case, since the model of target kinematics is used to propose particles, the weight equation (16) simplifies to become the measurement likelihood, $p(\mathbf{z}|\mathbf{X}_p)$.

Targets entering or leaving the surveillance region are accounted for as the proposed particle $\mathbf{X}_p^k$ may have either fewer targets or more targets than $\mathbf{X}_p^{k-1}$ (i.e. $T_p^k = T_p^{k-1} - 1$ or $T_p^k = T_p^{k-1} + 1$). This algorithm is illustrated in Table 2.

The Inefficiency of the SIR Method

The SIR multitarget particle filter has the benefit of being simple to describe and easy to implement. These benefits, however, are erased since using the kinematics requires an enormous amount of particles for successful tracking. In fact, SIR is so numerically inefficient that problems of any realistic size are intractable.

Table 2. SIR Multitarget Particle Filter

1. For each particle p, p = $1, ..., N_{part}$, Sample $(\mathbf{X}_p, T_p)$ from $q(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1}, \mathbf{z}^k) = p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$
2. Compute w_p^k = $w_p^{k-1} * p(\mathbf{z}|\mathbf{X}_p)$ for each p
3. Normalize w_p to sum to 1, $w_p \leftarrow w_p / \sum_{i=1}^{N_{parts}} w_i$.
4. Resample N_{part} particles with replacement from $\mathbf{X}_p$ based on w_p

Assume for discussion that the sensor is pixelated, returning energy in one of C sensor cells as discussed in Section 2. Target birth may occur in any unoccupied cell at any time step. Target death may occur in any occupied cell at any time step. One method of handling this would be to have a very large number of particles, capable of encoding all possibilities of the next state, i.e. no new target, one new target (in each of the possible unoccupied cells), or one less target (in each of the occupied cells) and still retain the particle diversity required for efficient filtering. Since the state space contains many possible locations for the new target (e.g. a 100x100 sensor grid), the straightforward method would require an enormous number of particles to include the possible permutations of targets removed and added.

Furthermore, even in the (artificial) case where there is no birth and death target proposals using the kinematics are too inefficient to be useful on realistic problems. Consider the case where there are M targets in the surveillance region. In order for a particle to be a good estimate of the multitarget state, all M partitions must be proposed to good locations. Without knowledge of the measurements, the probability of an individual target being proposed to a good location is much less than 1. Therefore, as the number of targets grows, the number of particles required to perform good tracking with high probability grows exponentially.

Both of these problems can be remedied by using an importance density that more closely approximates the optimal importance density (i.e. uses the current measurements to direct particle proposals to higher likelihood multitarget states). Therefore, rather than using the kinematics $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$ for proposal, we will carefully design an importance density that more closely approximates the optimal density $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1}, \mathbf{z}^k)$.

Importance Density Design for Target Birth and Death

In order to reach the efficiency required for tractable detection of multiple targets, we advocate a measurement directed sampling scheme for target birth and death. Specifically, we keep an existence grid (separate from the particles and tied to the sensor grid) which contains the probability for t targets in cell i at time k given the measurements $\mathbf{Z}^k$, $p_i(t^k|\mathbf{Z}^k)$. We consider only two possible values for t: 0 (no target in cell i) and 1 (a target exists in cell i). Therefore, the existence grid is merely a single vector of floating point numbers, one for each sensor cell.

The existence grid cells are initialized with a prior probability, $p_i(t^0|\emptyset)$, which may be spatially varying. The probability of target existence in each cell is propagated forward in time via

$$p_i(t^{k+1}|\mathbf{Z}^k) = \int p_i(t^{k+1}|t^k)p_i(t^k|\mathbf{Z}^k)dt^k \quad . \tag{22}$$

Where $p_i(t^{k+1}|t^k)$ is the model of time evolution of target number. Specifically, it encapsulates the probability of 1 target at time $k+1$ given there were 0 targets at time k, the probability of 1 target at time $k+1$ given there was 1 target at time k, and so on. According to the model outlined above, new targets arrive at the rate α and leave at the rate of β. This model completely specifies the transition density. Since t can only take on one of two values, this integral becomes a simple summation that is easily computable.

The existence grid is updated according to Bayes' rule when new measurements, $\mathbf{z}^{k+1}$, come in as

$$p_i(t^{k+1}|\mathbf{Z}^{k+1}) = \frac{p_i(t^{k+1}|\mathbf{Z}^k)p_i(\mathbf{z}^{k+1}|t^{k+1})}{p_i(\mathbf{z}^{k+1})} \quad . \tag{23}$$

These update procedures result in an existence grid that is separate from the particles and which contains a probability of target existence.

To handle target birth, new targets are preferentially added in locations according to the rate dictated by $p_i(\textit{target}^k|\mathbf{Z}^k)$ rather than at the nominal rate given by $\alpha(\mathbf{x})$. This is a bias which will be removed during the weight update process so that the Bayesian recursions are still exactly implemented. This implementational technique allows particles to be used more efficiently as new targets are only added in highly probable areas. Similarly, to handle target death, targets are preferentially removed at the rate dictated by $p_i(\textit{no target}^k|\mathbf{Z}^k)$ rather than the nominal rate given by β. Again, this bias is removed during the weight update so that the Bayesian recursions are still exactly implemented.

Let the number of possible locations a target could be added in the surveillance region at time k be denoted e^k. Furthermore, denote the existence grid weight in cell i at time k by g_i^k. Finally, for a set of integers Z let $\mu_m^Z(1,j),\ldots,\mu_m^Z(m,j)$, $j \in \{1,\ldots,\binom{|Z|}{m}\}$ denote the jth combination of m integers from Z. Then, if particle p adds a targets preferentially in cells j_1 and removes b targets preferentially from cells j_2, the bias correction factor (to be used in particle weight update) is given by

$$m_p = \frac{\lambda_a \rho_b}{\nu_{a,j_1}\kappa_{b,j_2}} \tag{24}$$

where λ and ρ are defined using the prior addition and removal probabilities as

$$\lambda_a = (1-\alpha)^{e_k}\Big(\frac{\alpha}{1-\alpha}\Big)^a \tag{25}$$

$$\rho_b = (1-\beta)^{T^{k-1}}\Big(\frac{\beta}{1-\beta}\Big)^b \tag{26}$$

and ν and κ are defined using the addition and removal rates used by the filter as

$$\nu_{a,j} = \prod_{i\in A^k}(1-g_i^k)\prod_{l=1}^{a} g^k_{\mu_a^{A^k}(l,j)} \Big/ \Big(1-g^k_{\mu_a^{A^k}(l,j)}\Big) \tag{27}$$

$$\kappa_{b,j} = \prod_{i=1}^{T^{k-1}}\big(1-\tau_i^k\big)\prod_{l=1}^{b}\tau^k_{\mu_b^{T^{k-1}}(l,j)} \Big/ \Big(1-\tau^k_{\mu_b^{T^{k-1}}(l,j)}\Big) \tag{28}$$

where

$$\tau_i^k = \beta\frac{1-g^k_{v_i^k}}{1/T^{k-1}\sum_{l=1}^{T^{k-1}}(1-g^k_{v_i^k})} \tag{29}$$

Importance Density Design for Persistent Targets

The drawback to using the kinematic prior for persistent targets is that the fact that the state vector represents many targets is not explicitly taken advantage of. Targets that are far apart in measurement space behave independently and should be treated as such. Furthermore, similar to that of the uniformed birth/death proposal, the current measurements are not used when proposing new particles. These two considerations taken together result in an inefficient use of particles and therefore require large numbers of particles to successfully track.

To overcome these deficiencies, we have developed alternative particle proposal techniques which bias the proposal process towards the measurements and allow for factorization of the target state when permissible. These strategies propose each partition (target) in a particle separately, and form new particles as the combination of the proposed partitions. We describe two methods here, the independent partitions (IP) method of [30] and the coupled partitions (CP) method. The basic idea of both CP and IP is to construct particle proposals at the partition level, incorporating the measurements so as to bias the proposal towards the optimal importance density. We show that each has benefits and drawbacks and propose an adaptive partition (AP) method which automatically switches between the two as appropriate. All of the methods are performed only on the persistent targets, and the algorithm is done in conjunction with the addition and removal of targets as described in the preceding section. Therefore the bias compensation term derived from addition/removal of targets, m_p, is part of each algorithm.

The permutation symmetry of the JMPD must be carefully accounted for when using these advanced sampling schemes. The CP method proposes particles in a permutation invariant manner, however it has the drawback of being computationally demanding. When used on all partitions individually, the IP method is not permutation invariant. Our solution is to perform an analysis of the particle set to determine which partitions require the CP algorithm, and which partitions may be proposed via the IP method. This analysis leads to the AP method of proposal which is permutation invariant.

Independent-Partition (IP) Method—The independent partition (IP) method given by Orton [30] is a convenient way to propose particles when part or all of the joint multitarget density factors. The Independent-Partition (IP) method proposes a new partition independently as follows. For a partition t, each particle at time $k-1$ has it's t^{th} partition proposed via the kinematic prior and weighted by the measurements. From this set of N_{part} weighted estimates of the state of the t^{th} target, we select N_{part} samples with replacement to form the t^{th} partition of the particles at time k.

Note that the importance density $q(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$ is no longer simply the model of target kinematics $p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1})$ as in the SIR Multitarget particle filter. Therefore, the weight given by eq. (16) does not simply become the likelihood $p(\mathbf{z}^k|\mathbf{X}^k, T^k)$. There is a bias which prefers to select partitions in accordance with the likelihood of the partition. To account for this sampling scheme, the biases corresponding to each particle for each target, $b_{p,t}$, are retained to use in conjunction with the likelihood $p(\mathbf{z}^k|\mathbf{X}^k, T^k)$ when computing particle weights. This is summarized in Table 3.

Table 3. Independent Partition Particle Filter

1. For each partition, $t = 1 \cdots T_{max}$,
(a) Propose partition t via Independent Partition Subroutine
2. Remove or add partitions as in Section 3 resulting in bias term m_p
3. Compute $w_p^k = w_p^{k-1} * \frac{p(\mathbf{z}|\mathbf{X}_p)}{m_p \prod_{t=1}^{T_p} b_{p,t}}$

Independent Partition Subroutine for Target t:
1. For each particle $p = 1, ..., N_{part}$,
(a) Sample $\mathbf{X}_{p,t}^* \sim p(\mathbf{X}_{p,t}^k|\mathbf{X}_{p,t}^{k-1})$
(b) Compute $w_p = p(\mathbf{z}|\mathbf{X}_{p,t}^*)$
2. Normalize w_p to sum to 1, $w_p \leftarrow w_p / \sum_{i=1}^{N_{parts}} w_i$.
3. For each particle $p = 1, ..., N_{part}$,
(a) Sample an index j from the distribution defined by w
(b) Set $\mathbf{X}_{p,t} = \mathbf{X}_{j,t}^*$
(c) Retain bias of sample, $b_{p,t} = w_j$

It is important to carefully account for the permutation symmetry issue discussed in Section 3 here. The IP method makes the critical assumption that partition t in each particle corresponds to the same target. Therefore, the partitions in each particle must be identically positioned before this method is applied. If IP is applied to particles that have different orderings of partitions, multiple targets will be grouped together within the same partition and erroneously used to propose the location of a single target. However, when this assumption of target/partition correspondence is valid, IP is an effective sampling strategy because it combines results for each partition across particles, resulting in improved numerical efficiency.

In the case of well separated targets, this method allows many targets to be tracked with the same number of particles needed to track a single target. Indeed, as mentioned earlier, in the case of well separated targets, the multitarget tracking problem breaks down into many single-target problems. The IP method is useful for just this case, as it allows the targets to be treated independently when their relative spacing deems that appropriate. Note, however, that this method is not applicable when there is any measurement-to-target association ambiguity. Therefore, when targets are close together in sensor space, an alternative method must be used.

Coupled Partition (CP) Proposal Method—When the posterior distributions on target position begin to overlap, we say that the corresponding partitions are coupled. In these instances, the IP method is no longer applicable, and another method of particle proposal such as Coupled Partitions (CP) must be used. An alternative method would be to use the IP strategy on groups of partitions as suggested in [30].

We apply the coupled partitions method as follows. To propose partition t of particle p, CP proposes R possible realizations of the future state using the kinematic prior. The R proposed futures are then given weights according to the current measurements and a single representative is selected. This process is repeated for each particle until the t^{th} partition for all particles has been formed. This can interpreted as an auxiliary particle filter [31] where the multiplicity R plays the role of the auxiliary variable. As in the IP method, the final particle weights must be adjusted for this biased sampling. This is summarized in table 4.

This algorithm is a modified version of the traditional SIR technique that operates on partitions individually. It improves tracking performance over SIR at the expense of additional computations.

Adaptive Particle Proposal Method—In order to mitigate the problem of additional computational cost of the CP method, and the problems with the IP method when targets are close together, we propose a hybrid solution, called the Adaptive-Partition (AP) method. The adaptive-partition method again considers each partition separately. Those partitions that are sufficiently well separated according to a given metric (see below) from all other partitions are treated as independent and proposed using the IP method. When targets are not sufficiently distant, the CP method is used.

Table 4. Coupled Partition Particle Filter

1. For each partition, $t = 1 \cdots T_{max}$
(a) Propose partition t via Coupled Partition Subroutine
2. Remove or add partitions as in Section 3 resulting in bias term m_p
3. Compute $w_p^k = w_p^{k-1} * \frac{p(\mathbf{z}|\mathbf{X}_p)}{m_p \prod_{t=1}^{T_p} b_{p,t}}$

Coupled Partition Subroutine for Target t

1. For each particle $p = 1, ..., N_{part}$,
(a) For each proposal $r = 1, ..., R$
i. Sample $\mathbf{X}^*_{p,t}(r) \sim p(\mathbf{X}^k_{p,t}|\mathbf{X}^{k-1}_{p,t})$
ii. Compute $w_r = p(\mathbf{z}|\mathbf{X}^*_{p,t}(r))$
(b) Normalize w_r to sum to 1, $w_r \leftarrow w_r / \sum_{i=1}^{R} w_i$.
(c) Sample an index j from the distribution defined by w
(d) Set $\mathbf{X}_{p,t} = \mathbf{X}^*_{p,t}(j)$
(e) Retain bias of sample, $b_{p,t} = w_j$

To determine when targets are sufficiently separated, we threshold based on distance in sensor space between the estimated state of the i^{th} partition and the j^{th} partition. Denote by $\mathbf{x}'_i$ the estimated x and y positions of the i^{th} partition (40). Notice only the spatial states are used (i.e. velocities are neglected), as these are the states that measure distance in sensor space for our model. We have computed the distance between two partitions using a Euclidian metric between the estimated centers, and the Mahalanobis metric (30), where $\hat{\Sigma}_j$ is the covariance associated with the estimate of the j^{th} partition (41).

$$r^2 = (\mathbf{x}'_i - \mathbf{x}'_j)' \hat{\Sigma}_j^{-1} (\mathbf{x}'_i - \mathbf{x}'_j) \tag{30}$$

We have additionally used a nearest neighbor type criteria, where partitions are considered coupled if any sample from partition i is closer to the center of partition j then any sample from partition j. In practice, it is found that simply using the Euclidian distance between estimated states is sufficient and less computationally burdensome. The adaptive proposal method is summarized in Table 5.

Table 5. Adaptive Proposal Method

1. For each partition $t = 1 : T_{max}$
(a) $d(t) = \min_{j \neq t} ||\mathbf{x}'_t - \mathbf{x}'_j||$
(b) if $d(t) > \tau$
Propose partition t using IP method
(c) else
Propose partition t using CP method
2. Remove or add partitions as in Section 3 resulting in bias term m_p
3. For each particle $p = 1, ..., N_{part}$
$w_p^k = w_p^{k-1} * \frac{p(\mathbf{z}|\mathbf{X}_p)}{m_p \prod_{t=1}^{T_p} b_{p,t}}$

A further refinement to the CP method improves performance. In this method, those partitions that are deemed to be coupled are clustered according to the method of section 3. This results in "partitions" that contain multiple targets – some with 2 targets, some with 3 targets, etc. Then instead of proposing each target individually, the clustered pairs (triplets, etc.) of targets are proposed all at once. This method is summarized in Table 6. Note that the idea of a partition containing multiple targets is also present in the work of Orton [30], although adaptively deciding partition boundaries and partition clustering is new to this work.

Table 6. Modified Adaptive Proposal Method

1. Cluster targets into C groups
2. For each group $c = 1 : C$
(a) if group C has one entry,
Propose group c using IP method
(b) else
Propose group c using CP method
3. Remove or add partitions from selected particles as in Section 3
4. For each particle $p = 1, ..., N_{part}$
$w_p^k = w_p^{k-1} * \frac{p(\mathbf{z}|\mathbf{X}_p)}{m_p \prod_{c=1}^{C} b_{p,c}}$

Permutation Symmetry and Partition Sorting

As discussed throughout the preceding sections, the permutation symmetry associated with the JMPD discussed in Section 2 is directly inherited by the particle filter representation of the JMPD. Each particle contains many partitions (as many as the number of targets it estimates exist in the surveillance region) and the permutation symmetry of JMPD is visible through the fact that the relative ordering of targets may change from particle to particle. We refer to the permutation symmetry in this context as partition swapping.

The fact that partitions are in different orders from particle to particle is of no consequence when the object of interest is an estimate of the joint multitarget density. Each particle contributes the correct amount of mass in the correct location to the multitarget density irrespective of the ordering of its partitions.

However, the IP scheme requires that particles be identically ordered. Furthermore, estimating the multitarget states from the particle filter representation of JMPD must also be done in a way that is invariant to permutations of the particles. Therefore, before estimating target states, we permute the particles so that each particle has the targets in the same order. We use the K-means algorithm [10] to cluster the partitions of each particle, where the optimization is done across permutations of the particles. This is a very light computational burden in practice for two reasons. First, those partitions that are not coupled are already consistently ordered and need not be involved in the clustering procedure. Second, since this re-ordering occurs at each time step, those partitions that are coupled are nearly ordered already, and so one iteration of the K-means algorithm is typically enough to find the best permutation.

The details of the K-means algorithm are as follows. First, we state the notion of permutation symmetry precisely. Suppose a particle has T_p partitions labeled $t = 1 \cdots T_p$. A permutation π_p is a reshuffling of the labels, $\pi_p : i \rightarrow \pi_p(i)$. So a particle defined

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \cdots, \mathbf{x}_{p,T_p}] \tag{31}$$

Under the permutation π_p is reordered to

$$\mathbf{X}_p = [\mathbf{x}_{p,\pi_p(1)}, \mathbf{x}_{p,\pi_p(2)}, \cdots, \mathbf{x}_{p,\pi_p(T_p)}] \tag{32}$$

Denote by π a set of permutations for each particle, $\pi_p, p = 1 \cdots N_{part}$. We define the mean of the tth partition under the permutation π as

$$\bar{\mathbf{X}}_t(\pi) = \sum_{p=1}^{N_{parts}} w_p \mathbf{X}_{p,\pi_p(t)} \tag{33}$$

where it is understood that the summation is taken over only those particles that have partition t, and the weights are appropriately normalized to this subset. Further, define the χ^2 statistic as

$$\chi^2(\pi) = \sum_{p=1}^{N_{parts}} \sum_{t=1}^{T_p} w_p (\mathbf{X}_{p,\pi_p(t)} - \bar{\mathbf{X}}_t(\pi_p))^2 \tag{34}$$

To reorder the particles, the goal is to find the set of permutations π that minimize χ^2, i.e.

$$\hat{\pi} = \min_{\pi} \chi^2(\pi) \tag{35}$$

The K-means algorithm is a well known method of approximately solving problems of this type. An initial permutation π is assumed and perturbations about that value are made to descend and find the locally optimal π. As mentioned earlier, re-ordering is done at each iteration of the algorithm, so the initial ordering is typically very close to the globally optimal ordering. Therefore, the K-means algorithm typically converges to the global optimum after a very small number of iterations (often 1). The algorithm is given in Table 7.

Table 7. K-means Algorithm Optimizing Over Partition Orderings

1. Initialize with π = current ordering of partitions
2. Compute $\bar{\mathbf{X}}_t(\pi)$ for $t = 1 \cdots T_p$ using (33)
3. For each particle p, permute the particle (update π_p) to yield

$$\pi_p \leftarrow \arg\min_{\pi_p} \sum_{t=1}^{T_p} (\mathbf{X}_{p,\pi_p(t)} - \bar{\mathbf{X}}_t(\pi_p))^2$$

4. If no particles have changed permutation from π, quit. Otherwise set $\pi = (\pi_1, \cdots, \pi_p, \cdots, \pi_{N_{part}})$ and go to 2

Notice that if the K-means algorithm fails to return the globally best reshuffling, this is not a serious problem. The main effect is that the CP algorithm will be used more than is minimally necessary. This results in increased computation but no performance degradation. A secondary effect is that target estimates will be slightly incorrect. There will be only a minor error relative to the sensor resolution because a local minimum will at worst mix partitions that are very close together in sensor space.

Estimation

Estimates of various quantities may be easily made using the particles.

Equation (3) gives the expression for computing the probability there are exactly T targets in the surveillance volume from the JMPD. To extract this from the particle filter approximation, first define the indicator variable $I_p(T)$ for $p = 1...N_{parts}$,

$$I_p(T) = \begin{cases} 1 & \text{if } T_p = T \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

Then the probability of T targets in the surveillance volume, $p(T|\mathbf{Z})$, is approximated by

$$p(T|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} I_p(T) w_p \tag{37}$$

Hence, the estimate of the probability that there are T targets in the surveillance volume is merely the sum of the weights of the particles that have T partitions. Note that the particle weights, w_p, are normalized to sum to unity for all equations given in this section.

To compute the estimated state and covariance of target i, we first define a second indicator variable $\tilde{I}_p(i)$ that indicates if particle p has a partition corresponding to target i. This is necessary as each particle is a sample drawn from the JMPD and hence may have a different number of partitions (targets):

$$\tilde{I}_p(i) = \begin{cases} 1 & \text{if partition } i \text{ exists in particle } p \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

Note that the sorting procedure of Section 3 has identified an ordering of particles to allow $\tilde{I}_p(i)$ to be determined. Furthermore, we define the normalized weights to be

$$\hat{w}_p = \frac{w_p \tilde{I}_p(i)}{\sum_{l=1}^{N_{part}} \tilde{I}_l(i) w_l} \tag{39}$$

So $\hat{w}_p$ is the relative weight of particle p, with respect to all particles containing a partition corresponding to target i. Then the estimate of the state of target i is given by

$$\hat{\mathbf{X}}(i) = E[\mathbf{X}(i)] = \sum_{p=1}^{N_{part}} \hat{w}_p \mathbf{X}_{p,i} \tag{40}$$

which is simply the weighted summation of the position estimates from those particles that are tracking target i. The

covariance is given similarly as

$$\hat{\Lambda}(i) = \sum_{p=1}^{N_{part}} \hat{w}_p(\mathbf{X}_{p,i} - \hat{\mathbf{X}(i)})(\mathbf{X}_{p,i} - \hat{\mathbf{X}(i)})' \qquad (41)$$

The indicator function $\tilde{I}_p(i)$ ensures that the summations in eq. (40) and eq. (41) are taken over only those particles that are tracking target i. The permutation symmetry issue mentioned earlier comes to the forefront here. Notice that without a clustering on the partitions, it is not necessarily true that partition i of particle j is tracking the same target that partition i of particle $j+1$ is tracking. Therefore, before evaluation of eq. (40) or eq. (41), the clustering procedure discussed in Section 3 is performed.

Resampling

In the traditional method of resampling, after each measurement update N_{part} particles are selected with replacement from $\mathbf{X}_p$ based upon particle weights w_p. The result is a set of N_{part} particles with uniform weight that approximate the density $p(\mathbf{X}|\mathbf{Z})$. Particles that do not correspond to measurements are eliminated – in particular, particles whose T_p value is not supported by measurements (too many or too few targets) are selected with low probability.

The particular resampling that was used in this work is systematic resampling [1]. This resampling strategy is easily implemented, runs in order N_{parts}, is unbiased, and minimizes the Monte Carlo variance. Many other resampling schemes and modifications are presented in the literature [7]. Of these methods, we have found that adaptively choosing at which time steps to resample [21] based on the number of effective particles leads to improved performance while reducing compute time. All results presented herein use the method of [21] to determine which times to resample and use systematic resampling [1] to perform resampling. We have also found that Markov Chain Monte Carlo (MCMC) moves using a Metropolis-Hastings scheturneme [7] leads to slightly improved performance in our application.

4. SIMULATION RESULTS

We illustrate the performance of our multitarget tracking scheme by considering the following model scenario involving ten moving ground targets and an airborne sensor.

The ten ground targets used in this simulation move in a $5000m \times 5000m$ surveillance area. Targets are modeled using the four-dimensional state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]$. The target motion in the simulation is taken from a set of recorded data based on GPS measurements of vehicle positions collected as part of a battle training exercise at the Army's National Training Center. This battle simulation provides a large number of real vehicles, including army HMMWVs, armored personnel carriers, tanks, and the like. The vehicles follow a prescribed trajectory over natural terrain. Based on an empirical fit to the data, we found that a nearly constant velocity model (see eq. (4)) was adequate to model the behavior of the vehicles for these simulation studies and is therefore used in all experimental results presented herein.

We utilize the idealized sensor described in Section 2. The sensor scans a fixed rectangular region of 50×50 pixels, where each pixel represents a $100m \times 100m$ area on the ground plane. The sensor returns Rayleigh-distributed measurements in each pixel, depending on the number of targets that occupy the pixel. We use thresholded measurements with $P_d = .5$ in all cases.

The filter was initialized with no knowledge, i.e. 0 targets and a uniform density over the surveillance region of target existence. As measurements come in, targets are added according to the measurement directed scheme outlined earlier. Targets that persist from time to time are tracked according to the nearly constant velocity model given earlier.

We measure the performance of the algorithm in two ways. First, we compare the estimated number of targets to the true number of targets, where the estimated number of targets at time k is defined as

$$\hat{r}^k = \sum_{r=0}^{\infty} r \int_{\mathbf{X}} d\mathbf{X} p(\mathbf{X}, r|\mathbf{Z}) \approx \sum_{p=1}^{n} w_p r_p \qquad (42)$$

Second, we use the ground truth to calculate the number of actual targets that are successfully tracked by the filter. For each of the hypothesized target t, we have an estimate of the target state as

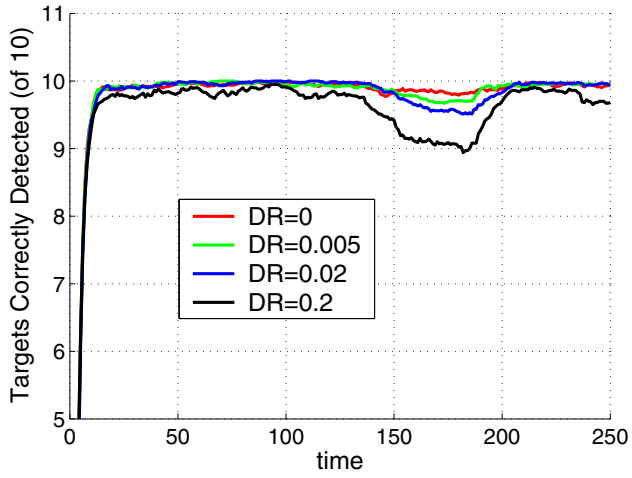
$$\hat{x}_t^k = \int \mathbf{x}_t d\mathbf{x}_1 \cdots \mathbf{x}_t p(\mathbf{x}_1 \cdots \mathbf{x}_T|\mathbf{Z}) \approx \sum_{p=1}^{n} \hat{w}_p \mathbf{X}_{p,t} \qquad (43)$$

where $\hat{w}_p$ is normalized to sum to one over all particles that contain partition t. The target estimates are then matched up with the ground truth to give a measure of how many true targets are being successfully tracked, which we denote $r_{tracked}$. Note this measure captures both targets that should have been detected but weren't as well as targets that were successfully detected but then poorly tracked.

These two measures allow for determination of the number of false targets initiated as well as the number of true targets not under track. Figures 2 through 4 show the performance of the algorithm as different parameters are varied.

5. CONCLUSIONS

This paper has presented a method for simultaneous detection and tracking of multiple moving targets based on recursive estimation of the joint multitarget probability density using particle filtering methods. The importance density used by the particle filter is adaptive and measurement directed. We have shown the efficacy of the method using a collection of ten real moving ground targets and simulated Rayleigh distributed thresholded measurements.



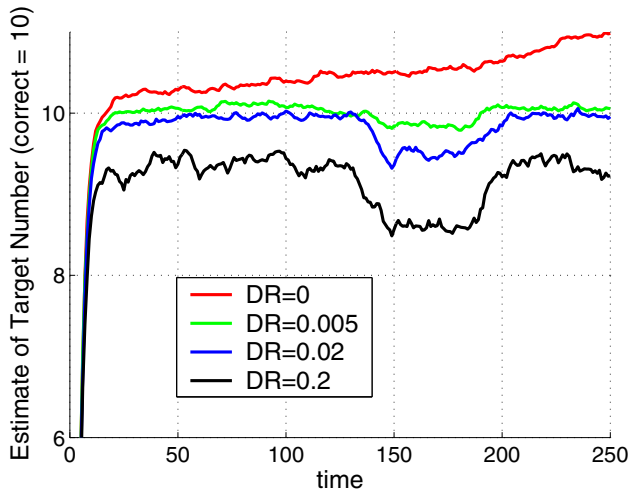
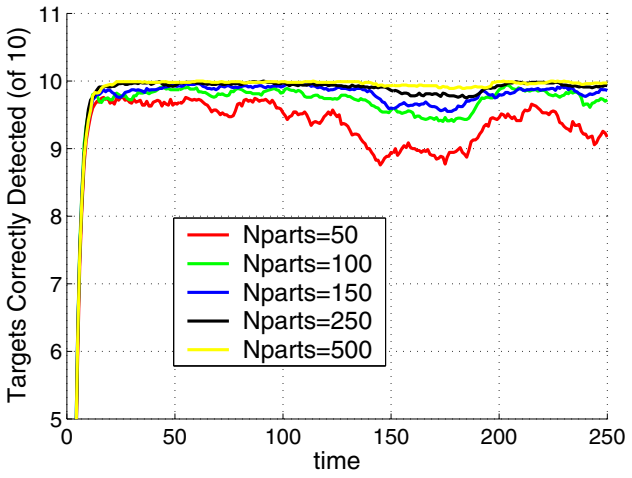


Figure 2. Performance of the tracker as the death rate (DR) parameter β is varied. Death rate of .005 performs best. The drop seen in the curve at time 150-200 is a location where targets cross, and often times one is removed when the death rate is too high.



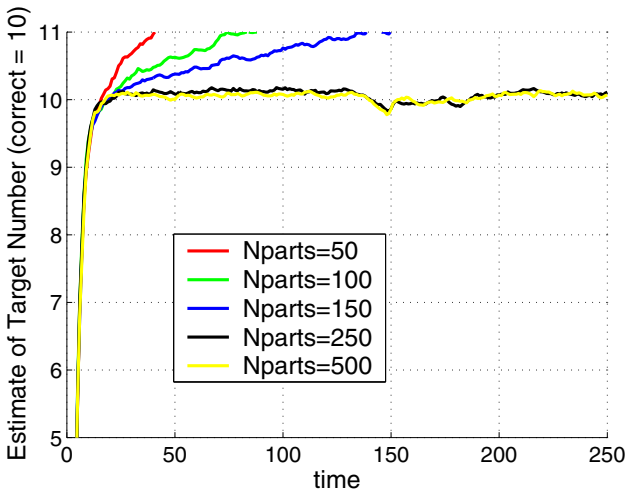


Figure 3. The performance of the multitarget tracker as the number of particles used is varied. 250 particles is sufficient to detect and track the ten real targets as they move through the surveillance region.

6. Acknowledgements

This work was supported by the USAF Contract No. F33615-02-C-1199, AFRL contract SPO900-96-D-0080, and ARO-DARPA MURI Grant DAAD19-02-1-0262. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Air Force.

References

[1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 174-188, 2002.

[2] Y. Bar-Shalom, *Multitarget Multisensor Tracking: Advanced Applications*, Artech House, 1990.

[3] S. Blackman, *Multiple-Target Tracking with Radar Applications*, Dedham, MA: Artech House, 1986.

[4] R. E. Bethel and G. J. Paras, "A PDF Multisensor Multitarget Tracker," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 34, no. 1, January 1998, pp. 153-168.

[5] Y. Bar-Shalom and W. D. Blair, *Multitarget-Multisensor Tracking: Applications and Advances, Volume III*, Boston: Artech House, 2000.

[6] H. A. P. Blom and E. A. Bloem, "Joint IMMPDA Particle Filter", *Proceedings of the 6th International Conference on Information Fusion*, Cairns, Queensland, Australia, July 2003.

[7] A. Doucet, N. de Freitas, and N. Gordon, *Sequential*

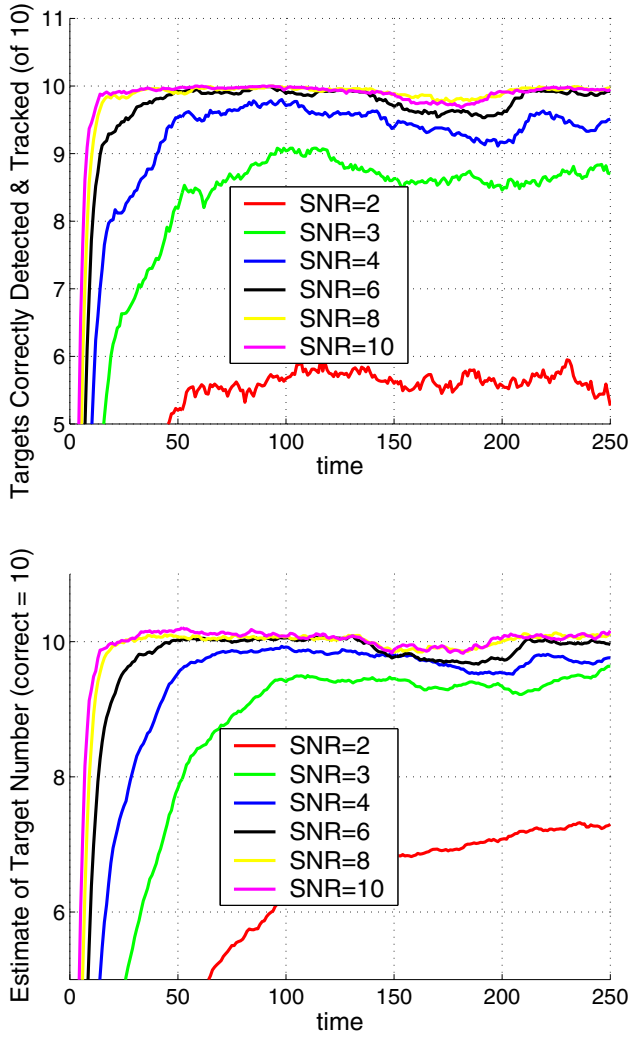


Figure 4. The performance of the multitarget tracker as the SNR is varied. This simulation uses 250 particles and $\beta = .005$ for each sensor cell.

Monte Carlo Methods in Practice, New York: Springer Publishing, 2001.

[8] A. Doucet, "On Sequential Monte Carlo Methods for Bayesian Filtering", Dept. Eng. Univ. Cambridge, UK. Tech. Rep., 1998.

[9] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, "A Novel Approach to Non-linear and Non-Gaussian Bayesian State Estimation," *IEE Proceedings on Radar and Signal Processing*, vol. 140, pp. 107-113, 1993.

[10] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer Series in Statistics. Springer Verlag, Basel, 2001.

[11] C. Hue, J.-P. Le Cadre, and P. Perez, "Tracking Multiple Objects with Particle Filtering", *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 791-812, 2002.

[12] M. Isard and J. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker", Proceedings of the 8th International Conference on Computer Vision, 2001.

[13] E. W. Kamen, "Multiple Target Tracking Based on Symmetric Measurement Functions", *IEEE Transactions on Automatic Control*, vol. 37, no. 3, pp. 371-374, 1992.

[14] R. Karlsson and F. Gustafsson, "Monte Carlo Data Association for Multiple Target Tracking", *Proceedings of The IEE Workshop on Target Tracking : Algorithms and Applications*, The Netherlands, 2001.

[15] K. Kastella, "Discrimination Gain to Optimize Classification", *IEEE Transactions on Systems, Man and Cybernetics-Part A: Systems and Humans*, Vol. 27, no. 1, January 1997.

[16] K. Kastella, "Discrimination Gain for Sensor Management in Multitarget Detection and Tracking", IEEE-SMC and IMACS Multiconference CESA '96, vol. 1, Lille France, July 9-12 1996, pp. 167-172.

[17] K. Kastella, "A Maximum Likelihood Estimator for Report-to-Track Association", *SPIE Proceedings*, vol. 1954, pp. 386-393, 1993.

[18] K. Kastella, "Event Averaged Maximum Likelihood Estimation and Mean-Field Theory in Multitarget Tracking", *IEEE Transactions on Automatic Control*, vol. 50, no. 6, 1070-1073, June 1995.

[19] C. Kreucher, K. Kastella, A. O. Hero III, "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density", *Under review at IEEE Transactions on Aerospace and Electronic Systems*.

[20] C. Kreucher, K. Kastella, and A. O. Hero, "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density", SPIE International Symposium on Optical Science and Technology, San Diego, California, August 2003.

[21] J. Liu, and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems", Journal of the American Statistical Association, September 1998.

[22] M. I. Miller, A. Srivastava, A., and U. Grenander, "Conditional Mean Estimation via Jump-Diffusion Processes in Multiple Target Tracking/Recognition.", *IEEE Transactions on Signal Processing*, **43**, 11, 2678-2690, 1995.

[23] R. Mahler, "Global Optimal Sensor Allocation", *Proceedings of the Ninth National Symposium on Sensor Fusion*, vol. I, March 12-14, 1996, Naval Postgraduate School, Monterey CA, pp. 347-366.

[24] I. R. Goodman, R. P. S. Mahler and H. T. Nguyen, *Mathematics of Data Fusion*, Kluwer Academic Publishers, Dordrecht, 1997.

[25] S. Musick, K. Kastella, and R. Mahler, "A Practic1al Implementation of Joint Multitarget Probabilities" in I. Kadar (ed.) *Signal Processing, Sensor Fusion, and Target Recognition VII*, SPIE Vol. 3374, pp. 26-37.

[26] R. Mahler, "A Unified Foundation for Data Fusion", in F. A. Sadjadi (ed.)., *Selected Papers on Sensor and Data Fusion*, SPIE Vol. MS-124, 1996, pp. 325-345.

[27] S. Maskell, M. Rollason, N. Gordon, D. Salmond, "Efficient Particle Filtering for Multiple Target Tracking with Application to Tracking in Structured Images", *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, 2002*, Orlando, April 1-5, 2002.

[28] S. Mori, C. Y. Shong, E. Tse, and R. P. Wishner, "Tracking and Classifying Multiple Targets Without a Prior Identification", *IEEE Transactions on Automatic Control*, AC31, May 1986.

[29] S. Musick, K. Kastella, and R. Mahler, "A Practical Implementation of Joint Multitarget Probabilities.", *SPIE Proceedings*, vol. 3374, 1998, pp. 26-37.

[30] M. Orton and W. Fitzgerald, "A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters", *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 216-223, 2002.

[31] M. K. Pitt and N. Shephard, "Filtering via Simulation: Auxiliary Particle Filters", *Journal of the American Statistical Association*, vol. 94, no. 446, pp. 590-599, June 1999.

[32] H. M. Shertukde and Y. Bar-Shalom, "Tracking of crossing targets with imaging sensors", *IEEE Transactions on Aeropace and Electronic Systems*, vol. 27, no. 4, 582-592, 1991.

[33] L. D. Stone, C. A. Barlow, and T. L. Corwin, *Bayesian Multiple Target Tracking*, Artech House, 1999.

[34] H. Sidenbladh, and S-L. Wirkander, "Particle Filtering for Random Sets.",*IEEE Transactions on Aerospace and Electronic Systems*, to appear.

[35] G. W. Stimson, *Introduction to Airborne Radar*, SciTech Publishing, 2nd edition, 1998.

[36] R. L. Streit, and T. E. Luginbuhl, "Maximum Likelihood Methods for Probabilistic Multi-hypothesis Tracking.", In *Proceedings of SPIE*, vol. 2235, April 5-7, 1994

[37] B-N Vo, S. Singh, S., and A. Doucet, "Sequential Monte Carlo Implementatoin of the PHD filter for Multi-Target Tracking.", In *Proceedings of the sixth International Conference on Information Fusion*, 792-799, July 2003.

Chris Kreucher received the B.S. and M.S. degrees in Electrical and Computer Engineering from the University of Michigan-Dearborn in 1996 and 1997, respectively. Since 1997, he has been a researcher in the Exploitation Unit of General Dynamics (formerly ERIM) Advanced Information Systems Group, located in Ann Arbor, Michigan. Since 2002, he has also been a Ph.D. student in the Department of Electrical Engineering and Computer Science, working under the guidance of Professor Alfred O. Hero III. His current research interests include multitarget tracking and sensor management.

Mark R. Morelande received the B.Eng. degree in aerospace avionics from Queensland University of Technology (QUT), Brisbane, Australia, in 1997 and the Ph.D. degree in electrical engineering from Curtin University of Technology, Perth, Australia, in 2001. From November 2000 to January 2002, he was a Postdoctoral Fellow with the Centre for Eye Research, QUT. Since January 2002, he has been a Research Fellow with the Cooperative Research Centre for Sensor, Signal, and Information Processing, University of Melbourne, Parkville, Australia. His research interests include nonstationary signal analysis and target tracking with particular emphasis on multiple target tracking and application of sequential Monte Carlo methods to tracking problems.

Keith Kastella is Senior Scientist in the Exploitation Unit of General Dynamics Advanced Information Systems Group in Ann Arbor. His current research interests include multitarget tracking, sensor management and quantum information processing. He received his Ph.D. in high energy physics from SUNY Stony Brook in 1988.

Alfred O. Hero III received the B.S. in Electrical Engineering (summa cum laude) from Boston University (1980) and the Ph.D from Princeton University (1984), both in Electrical Engineering. Since 1984 he has been a Professor with the University of Michigan, Ann Arbor, where he has appointments in the Department of Electrical Engineering and Computer Science, the Department of Biomedical Engineering and the Department of Statistics. He has held visiting positions at I3S University of Nice, Sophia-Antipolis, France (2001), Ecole Normale Supérieure de Lyon (1999), Ecole Nationale Supérieure des Télécommunications, Paris (1999), Scientific Research Labs of the Ford Motor Company, Dearborn, Michigan (1993), Ecole Nationale Superieure des Techniques Avancees (ENSTA), Ecole Superieure d'Electricite, Paris (1990), and M.I.T. Lincoln Laboratory (1987 - 1989).

His research interests are in areas including: estimation and detection, statistical communications, bioinformatics, signal processing and image processing. He is a Fellow of the Institute of Electrical and Electronics Engineers (IEEE), a member of Tau Beta Pi, the American Statistical Association (ASA) , the Society for Industrial and Applied Mathematics (SIAM), and the US National Commission (Commission C) of the International Union of Radio Science (URSI). He has received the 1998 IEEE Signal Processing Society Meritorious Service Award, the 1998 IEEE Signal Processing Society Best Paper Award, and the IEEE Third Millenium Medal. In 2002 he was appointed IEEE Signal Processing Society Distinguished Lecturer.