Multitarget Tracking using the Joint Multitarget Probability Density

CHRIS KREUCHER
KEITH KASTELLA
General Dynamics Advanced Information Systems
ALFRED O. HERO, III, Fellow, IEEE
University of Michigan

This work addresses the problem of tracking multiple moving targets by recursively estimating the joint multitarget probability density (JMPD). Estimation of the JMPD is done in a Bayesian framework and provides a method for tracking multiple targets which allows nonlinear target motion and measurement to state coupling as well as non-Gaussian target state densities. The JMPD technique simultaneously estimates both the target states and the number of targets in the surveillance region based on the set of measurements made. We give an implementation of the JMPD method based on particle filtering techniques and provide an adaptive sampling scheme which explicitly models the multitarget nature of the problem. We show that this implementation of the JMPD technique provides a natural way to track a collection of targets, is computationally tractable, and performs well under difficult conditions such as target crossing, convoy movement, and low measurement signal-to-noise ratio (SNR).

Manuscript received December 10, 2003; revised May 3 and July 20, 2004; released for publication March 19, 2005.

IEEE Log No. T-AES/41/4/860802.

Refereeing of this contribution was handled by W. Koch.

This work was supported under the United States Air Force Contract F33615-02-C-1199, AFRL Contract SPO900-96-D-0080, and by ARO-DARPA MURI Grant DAAD19-02-1-0262. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the United States Air Force.

Authors' addresses: C. Kreucher and K. Kastella, General Dynamics, 1200 Joe Hall Dr., Ypsilanti, MI 48197, E-mail: (Christopher.Kreucher@gd-ais.com); A. O. Hero, III, Dept. of Electrical and Computer Engineering, University of Michigan, 1301 Beal Ave., Ann Arbor, MI 48109-2122.

0018-9251/05/$17.00 © 2005 IEEE

I. INTRODUCTION

The problem of tracking a single maneuvering target in a cluttered environment is a very well-studied area [4]. Normally, the objective is to predict the state of an object based on a set of noisy and ambiguous measurements. There are wide range of applications in which the target tracking problem arises, including vehicle collision warning and avoidance [39, 17], mobile robotics [50], human-computer interaction [24], speaker localization [66], animal tracking [65], tracking a person [11], and tracking a military target such as a ship, aircraft, or tank [8].

The single target tracking problem can be formulated and solved in a Bayesian setting by representing the target state probabilistically and incorporating statistical models for the sensing action and the target state transition. Implementationally, the standard tool is the ubiquitous Kalman filter [46], applicable and optimal when the measurement and state dynamics are Gaussian and linear.

In a more general setting where nonlinear target motions, non-Gaussian densities, or nonlinear measurement-to-target couplings are involved, more sophisticated nonlinear filtering techniques are necessary [3]. Standard nonlinear filtering techniques involve modifications to the Kalman filter such as the extended Kalman filter [26], the unscented Kalman filter [27], and Gaussian sum approximations [1], all of which relax some of the linearity assumptions present in the Kalman filter. However, these techniques do not accurately model all of the salient features of the density, which limits their applicability to scenarios where the target state posterior density is well approximated by a multivariate Gaussian density. To address this deficiency, others have studied grid-based approaches [35, 37], which utilize a discrete representation of the entire single target density. In this setup, no assumptions on the form of the density are required, so arbitrarily complicated densities may be accommodated. However, fixed grid approaches are computationally intractable except in the case of very low state space dimensionality [7].

Recently, the interest of the tracking community has turned to the set of Monte Carlo techniques known as particle filtering [19, 59]. A particle filter approximates a probability density on a set of discrete points, where the points are chosen dynamically via importance sampling. Particle filtering techniques have the advantage that they provide computational tractability [51], have provable convergence properties [12], and are applicable under the most general of circumstances, as there is no assumption made on the form of the density [15]. Indeed, particle filter based approaches have been used successfully in areas where grid-based [13] or extended or unscented Kalman filter-based [2, 44] filters have previously been employed.

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

The multitarget tracking problem has been traditionally addressed with techniques such as multiple hypothesis tracking (MHT) and joint probabilistic data association (JPDA) [8, 5, 6]. 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 [61] develops a mathematical theory of multiple target tracking from a Bayesian point of view. Srivistava, Miller [48], and Kastella [32] did early work in this area. For the same reasons as the single target case, fixed grid approaches to multitarget tracking are very computationally challenging.

Recently, some researchers have applied particle filter based strategies to the problem of multitarget tracking. In [22], Hue and Le Cadre use a particle filter based on the probabilistic multiple hypothesis tracker (PMHT) introduced by Streit [62]. 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 [29, 10].

The BraMBLe [25] system, the independent partition particle filter (IPPF) of Orton and Fitzgerald [52] and the work of Maskell [45] 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 focused on a tractable implementation of ideas in [61].

Mahler [41, 18, 49, 42] 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 [67, 57]. To date, these implementations have been limited to small scale problems for computational reasons. While developed independently [31]–[33], the JMPD method can be derived using the mathematics of random sets and expressed in the FISST framework (see [49]). As discussed there, JMPD can be traced back to the event-averaged maximum likelihood estimation (EAMLE) work of Kastella [31, 32] and many earlier works, e.g. [28, 47, 56]. Although this paper is more simply viewed as an application of the JMPD method developed in [31]–[33], it can also be viewed as an application of the FISST method in the setting of [49].

The main contribution of this paper is the development of a multiple target tracker that recursively estimates the entire joint multitarget probability density (JMPD) using particle filtering methods with adaptive sampling schemes. This is an expansion of our preliminary work which was published in [36]. In our formulation, we use one particle per scenario. That is, a particle encodes a hypothesis about the entire multitarget state—which includes the number of targets and the state (position, velocity, etc.) of each target. We demonstrate that the particle filter implementation of JMPD provides a natural way to track a collection of targets, is computationally tractable, and performs well under difficult conditions such as target crossing and convoy movement. The method avoids the need to create detections via thresholding and avoids directly performing measurement-to-track association. The measurement enters into the Bayesian formulation through its likelihood, which allows raw sensor measurements to be utilized. This feature allows the tracker to perform well in the low signal-to-noise ratio (SNR) regime.

These features distinguish the particle filter based JMPD approach from traditional approaches of MHT and JPDA as well as the approaches of Hue [22, 23] and others [29, 55, 16], which require thresholded measurements (detections) and a measurement-to-track association procedure. Further, by estimating the joint multitarget density rather than many single target densities, our method explicitly models target correlations. By utilizing an adaptive sampling scheme that exploits independence when present, our method benefits from computational advantages as in [52].

The rest of this paper is organized in the following manner. In Section II, we introduce the notion of the JMPD 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 III. 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. We furthermore detail the permutation symmetry issue (present in all multitarget tracking algoritms) and its manifestation in our particle filter estimation of the JMPD. In Section IV, we give simulation results detailing the performance of the particle filter based multitarget tracker proposed here. Finally, we conclude in Section V with a brief summary and discussion.

II. JOINT MULTITARGET PROBABILITY DENSITY

In this section, we introduce the details of using the JMPD for target tracking. The concept of JMPD was discussed by Kastella [30] where a method of tracking multiple targets that move between fixed discrete cells on a line was presented. We generalize the discussion to deal with targets

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

that have N-dimensional continuous valued state vectors and arbitrary kinematics. In the tracking problems considered here, we are interested in tracking the position (x, y) and velocity $(\dot{x}, \dot{y})$ of multiple targets. Each target is therefore described by the four-dimensional state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]$. By straightforward extension, the individual state vectors of the targets may be augmented to include things like target identification.

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 $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k \mid \mathbf{Z}^k)$ as the probability density for exactly T targets with states $\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ at time k based on a set of past observations $\mathbf{Z}^k$. In general, 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 time k, i.e., $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^k\}$, 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 density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k \mid \mathbf{Z}^k)$ 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 $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}, \mathbf{x}_T$ as a partition of the multitarget state $\mathbf{X}$. For convenience, the density is written more compactly in the traditional manner as $p(\mathbf{X}^k, T^k \mid \mathbf{Z}^k)$, which implies that the state-vector $\mathbf{X}$ represents a variable number of targets each possessing their own state vector. As an illustration, some examples of the sample space of p are

$p(\emptyset,\ T = 0 \mid \mathbf{Z})$, the posterior probability density for no targets in the surveillance volume,
$p(\mathbf{x}_1,\ T = 1 \mid \mathbf{Z})$, the posterior probability density for one target with state $\mathbf{x}_1$,
$p(\mathbf{x}_1, \mathbf{x}_2,\ T = 2 \mid \mathbf{Z})$, the posterior probability density for two targets with respective states $\mathbf{x}_1$ and $\mathbf{x}_2$,
$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3,\ T = 3 \mid \mathbf{Z})$, the posterior probability density for three targets with respective states $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$.

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 state $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2]$ 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.

We adopt the view that likelihoods such as $p(\mathbf{z} \mid \mathbf{X}, T)$ and the JMPD $p(\mathbf{X}, T \mid \mathbf{Z})$ are conventional Bayesian objects to be manipulated by the usual rules of probability and statistics. Thus, a multitarget system has state $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_T)$ with probability distribution $p(\mathbf{x}_1, \ldots, \mathbf{x}_T, T \mid \mathbf{Z})$ where T is variable. This can be viewed as a hybrid stochastic system where the discrete random variable T governs the dimensionality of $\mathbf{X}$. The probability that there are exactly T targets present in the system is given by the marginal distribution

$$p(T \mid \mathbf{Z}) = \int d\mathbf{x}_1 \cdots d\mathbf{x}_T\, p(\mathbf{x}_1, \ldots, \mathbf{x}_T, T \mid \mathbf{Z}). \tag{1}$$

As a marginal distribution, this is normalized such that

$$\sum_{T=0}^{\infty} p(T \mid \mathbf{Z}) = 1. \tag{2}$$

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 $p(\mathbf{X}^k, T^k \mid \mathbf{X}^{k-1}, T^{k-1})$ and is referred to as the kinematic prior (KP). The KP describes probabilistically how the state of the system evolves over time. It includes models of target motion, target birth and death, and any additional prior information that may exist such as terrain and roadway maps. The time-updated prediction density is computed via the model update equation:

$$p(\mathbf{X}^k, T^k \mid \mathbf{Z}^{k-1}) = \sum_{T^{k-1}=0}^{\infty} \int d\mathbf{X}^{k-1} p(\mathbf{X}^k, T^k \mid \mathbf{X}^{k-1}, T^{k-1}) p(\mathbf{X}^{k-1}, T^{k-1} \mid \mathbf{Z}^{k-1}). \tag{3}$$

This is a direct application of the Chapman-Kolmogorov equation for a hybrid stochastic system. This is equivalent to the set integral used in the FISST formulation [18]. The time evolution of the JMPD may simply be a collection of target kinematic models or may involve target birth and death. In the case where target identification is part of the state being estimated, different kinematic models may be used for different target types.

The measurement update equation uses Bayes' rule to update the posterior density with a new

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

measurement $\mathbf{z}^k$:

$$p(\mathbf{X}^k, T^k \mid \mathbf{Z}^k) = \frac{p(\mathbf{z}^k \mid \mathbf{X}^k, T^k) p(\mathbf{X}^k, T^k \mid \mathbf{Z}^{k-1})}{p(\mathbf{z}^k \mid \mathbf{Z}^{k-1})}. \quad (4)$$

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 $p(\mathbf{z} \mid \mathbf{X}, T)$ for each measurement $\mathbf{z}$ received.

A. Motion Modeling

In the simulation studies of Section IV, we assume that the number of targets is fixed but unknown and model the target motion as linear and independent for each target. The target motion of the simulation was taken from recorded vehicle data. We found that a nearly constant velocity model was adequate for these simulation studies, with the state for target t denoted $\mathbf{x}_t = (x_t, \dot{x}_t, y_t, \dot{y}_t)$ with discrete time transitions with period τ and indexed by k as

$$\mathbf{x}_t^k = \mathbf{F}\mathbf{x}_t^{k-1} + \mathbf{w}_t^k \quad (5)$$

where

$$\mathbf{F} = \begin{pmatrix} 1 & \tau & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \tau \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

$\mathbf{w}_t^k$ is 0-mean Gaussian noise with covariance $\mathbf{Q} = \mathrm{diag}(20,\ 0.2, 20,\ 0.2)$, which was selected based on an empirical fit to the data.

B. Sensor Modeling

In order to implement Bayes' formula (4), we must compute the measurement likelihood $p(\mathbf{z} \mid \mathbf{X}, T)$ (the time index is suppressed here to lighten notation). 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 nonlinearly) 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 the random process $z_m \sim H_t(\mathbf{x}_t, w_t)$.

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 subproblems: 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. [61] and in the grid-based sensor management work of [30]. 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, sidelobe 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.

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

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) [60]. The output of successive CPIs may also be averaged noncoherently. 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 $p(\mathbf{z} \mid \mathbf{X},T)$, 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 nonexceedances. 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

$$p(\mathbf{z} \mid \mathbf{X},T) = \prod_i p(z_i \mid \mathbf{X},T). \tag{7}$$

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 $p(\mathbf{z} \mid \mathbf{X},T)$.

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 sidelobe responses. The indicator function for the joint multitarget state is constructed as the logical disjunction

$$\chi_i(\mathbf{X}) = \bigvee_{t=1}^{T} \chi_i(\mathbf{x}_t). \tag{8}$$

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

$$i_{\mathbf{X}} = \{i \mid \chi_i(\mathbf{X}) = 1\}. \tag{9}$$

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, (7) becomes

$$p(\mathbf{z} \mid \mathbf{X},T) = \prod_{i\in i_{\mathbf{X}}} p(z_i \mid \mathbf{X},T) \prod_{i\notin i_{\mathbf{X}}} p_0(z_i) \propto \prod_{i\in i_{\mathbf{X}}} \frac{p(z_i \mid \mathbf{X},T)}{p_0(z_i)}. \tag{10}$$

Equation (10) allows for fairly general modeling of a pixelized sensor response to a collection of targets including nonlinear effects due to multiple targets contributing to a single pixel. One limitation is that 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 SNR, assumed constant across all targets, is denoted λ. We assume that when multiple targets lie within the same pixel their amplitudes add noncoherently (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 (10) becomes

$$p(\mathbf{z} \mid \mathbf{X},T) \propto \prod_{i\in i_{\mathbf{X}}} \frac{p_{n_i(\mathbf{X})}(z_i)}{p_0(z_i)}. \tag{11}$$

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 [9, 64] and to model clutter and target returns in turbulent

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

environments [20]. Rayleigh models are also often used for diffuse fading channels. In the unthresholded case, this is

$$p_n(z) = \frac{z}{1+n\lambda}\exp\left(-\frac{z^2}{2(1+n\lambda)}\right). \tag{12}$$

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

$$P_{d,n} = P_f^{1/(1+n\lambda)}. \tag{13}$$

This extends the usual relation $P_d = P_f^{1/(1+\lambda)}$ for thresholded Rayleigh random variables at SNR λ [5].

III. PARTICLE FILTER IMPLEMENTATION OF JMPD

We now turn to the development of a particle filter approximation to the JMPD. Even for modest problems, the sample space of $\mathbf{X}^k$ is large since it contains all possible configurations of state vectors $\mathbf{x}_i$ for all possible values of T. Earlier implementations of JMPD given by Kastella [30] 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 of 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 based implementation of JMPD breaks the computational logjam and allows us to investigate more realistic problems.

A. Single Target Particle Filter

Before detailing the particle filter implementation of JMPD, we first briefly review standard single target particle filtering. Particle filtering is a method of approximately solving the prediction and update equations by simulation [3, 19], 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, $p(\mathbf{x} \mid \mathbf{Z})$, is approximated by a set of N_{part} weighted samples (particles):

$$p(\mathbf{x} \mid \mathbf{Z}) \approx \sum_{p=1}^{N_{\text{part}}} w_p \delta_D(\mathbf{x} - \mathbf{x}_p) \tag{14}$$

where δ_D represents the usual Dirac delta function.

TABLE I
SIR Single Target Particle Filter

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

The model update (3) and the measurement update (4) are simulated by the following three-step recursion, summarized in Table I. 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 $q(\mathbf{x}^k \mid \mathbf{x}^{k-1}, \mathbf{z}^k)$. The design of the importance density is a well-studied area [15], 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 $p(\mathbf{x}^k \mid \mathbf{x}^{k-1}, \mathbf{z}^k)$, but this density is typically prohibitively difficult to sample from. In practice, oftentimes the importance density is chosen just to be the KP $p(\mathbf{x}^k \mid \mathbf{x}^{k-1})$. 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 KP) 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 [3]

$$w_p^k = w_p^{k-1} \frac{p(\mathbf{z}^k \mid \mathbf{x}_p^k) p(\mathbf{x}_p^k \mid \mathbf{x}_p^{k-1})}{q(\mathbf{x}_p^k \mid \mathbf{x}_p^{k-1}, \mathbf{z}^k)}. \tag{15}$$

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

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 [14]. Resampling may be done on a fixed schedule or based on the weight variance.

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

B. Multitarget Particle Filtering

To implement the JMPD recursions via a particle filter, we similarly approximate the joint multitarget probability density $p(\mathbf{X}, T \mid \mathbf{Z})$ by a set of N_{part} weighted samples. The multitarget state vector for T

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

TABLE II
SIR Multitarget Particle Filter

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

targets is written as

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}, \mathbf{x}_T]. \tag{16}$$

The particle state vector for T_p targets is

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \ldots \mathbf{x}_{p,T_p}] \tag{17}$$

where T_p can be any nonnegative integer. We refer to $\mathbf{x}_{p,j}$ as partition j of particle p. With δ_D denoting the Dirac delta, we define

$$\delta(\mathbf{X} - \mathbf{X}_p) = \begin{cases} 0 & T \neq T_p \\ \delta_D(\mathbf{X} - \mathbf{X}_p) & \text{otherwise} \end{cases}. \tag{18}$$

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

$$p(\mathbf{X}, T \mid \mathbf{Z}) \approx \sum_{p=1}^{N_{\text{part}}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \tag{19}$$

where $\sum w_p = 1$.

Different particles in the approximation may correspond to different estimates of the number T_p of targets in the surveillance region. In practice, the maximum number of targets a particle may track is truncated at some large finite number $T_{\max}$.

The JMPD $p(\mathbf{X}, T \mid \mathbf{Z})$ is defined for all possible numbers of targets, $T = 0, 1, 2, \ldots$. As each of the particles $\mathbf{X}_p$, $p = 1 \ldots N_{\text{part}}$ is a sample drawn from the JMPD $p(\mathbf{X}, T \mid \mathbf{Z})$, a particle $\mathbf{X}_p$ may have $0, 1, 2, \ldots$ partitions, each partition corresponding to a different target. Note that it is possible to have two or more targets in the same state. We have denoted the number of partitions in the pth particle $\mathbf{X}_p$ by T_p, where T_p may be different for different $\mathbf{X}_p$. 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 II. This simply proposes new particles at time k using the particles at time $k-1$ and the target kinematics model (5) while (11) is used in the weight update.

Again, since the model of target kinematics $p(\mathbf{X}^k \mid \mathbf{X}^{k-1})$ is used to propose particles, the weight equation (15) simplifies to become the measurement likelihood $p(\mathbf{z} \mid \mathbf{X}_p)$.

Targets entering or leaving the surveillance region can be accounted for by modifying the proposal density to incorporate a probability that the proposed particle $\mathbf{X}_p^k$ has either fewer targets or more targets then $\mathbf{X}_p^{k-1}$. For example, assume a per target death rate of α, which may be spatially varying to account for the fact that targets exit along the boundaries of the surveillance region. Then when proposing new particles, with probability α, partitions are removed from proposed particle p and the updated number of targets in this particle T_p^k is correspondingly reduced from T_p^{k-1}. In Table II, step 1a is modified so that each partition is proposed forward with probability $1 - \alpha$, rather than with probability 1.

Further, assume a birth rate β. Then when proposing new particles, with probability β, a new target is added to particle p. The location of the new target may be random, or more realistically chosen along the perimeter of the surveillance area. In this case, the number of targets represented by this particle is updated to $T_p^k = T_p^{k-1} + 1$. Again, this modifies step 1a in Table II, in that one extra partition is proposed to exist in particle p at time k with probability β. This target birth/death model is similar to other models, e.g. see [43] and [58], that have been proposed for multiple target tracking.

More complicated models of target birth and death (e.g. multiple targets arriving simultaneously, or targets arriving when existing targets are in certain strategically important places, etc.) are permissible under this framework but are not considered here.

C. Multitarget Particle Proposal

Using the KP as the importance density has the benefit that it is simple to implement and is computationally inexpensive on a per particle basis. As we will see later, this computational efficiency is erased by the fact that a very large number of particles are required using this importance density. One obvious drawback is that the KP does not explicitly take advantage of the fact that the state vector in fact represents many targets. Targets that are far apart in measurement space behave independently and should be treated as such. A second drawback, common to many particle filtering applications, is that 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 employed alternative particle proposal techniques which bias the proposal process towards the measurements and allow for factorization of the target state when

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

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 [52] 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.

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.

1) *IP Method*: The IP method given by Orton [52] is a convenient way to propose particles when part or all of the joint multitarget density factors. The IP method proposes a new partition independently as follows. For a partition t, each particle at time $k-1$ has it's tth partition proposed via the KP and weighted by the measurements. From this set of N_{part} weighted estimates of the state of the tth target, we select N_{part} samples with replacement to form the tth partition of the particles at time k.

Note that the importance density q is no longer simply the model of target kinematics $p(\mathbf{X}^k, T^k \mid \mathbf{X}^{k-1}, T^{k-1})$ as in the SIR multitarget particle filter. Therefore, the weight given by the weight (15) does not simply become the likelihood $p(\mathbf{z}^k \mid \mathbf{X}^k, T^k)$. There is a bias added 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 $p(\mathbf{z}^k \mid \mathbf{X}^k, T^k)$ when computing particle weights. This is summarized in Table III.

It is important to carefully account for the permutation symmetry issue discussed in Section III 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.

TABLE III
Independent Partition Particle Filter

1) For each partition, $t = 1 \cdots T_{\max}$,
   a) Propose partition t via Independent Partition Subroutine
2) Compute $w_p^k = w_p^{k-1} * \frac{p(\mathbf{z}|\mathbf{X}_p)}{\prod_{t=1}^{T_p} b_{p,t}}$

Independent Partition Subroutine for Target t:

1) For each particle $p = 1, \ldots, N_{\text{part}}$,
   a) Sample $\mathbf{X}_{p,t}^* \sim p(\mathbf{X}_{p,t}^k \mid \mathbf{X}_{p,t}^{k-1})$
   b) Compute $w_p = p(\mathbf{z} \mid \mathbf{X}_{p,t}^*)$
2) Normalize w_p to sum to 1, $w_p \leftarrow w_p / \sum_{i=1}^{N_{\text{parts}}} w_i$.
3) For each particle $p = 1, \ldots, N_{\text{part}}$,
   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, $b_{p,t} = w_j$

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.

2) *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 CP must be used. An alternative method would be to use the IP strategy on groups of partitions as is suggested in [52].

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 KP. 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 tth partition for all particles has been formed. This can interpreted as an auxiliary particle filter [53] 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 IV.

This algorithm is a modified version of the traditional SIR technique that operates on partitions

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

TABLE IV
Coupled Partition Particle Filter

1) For each partition, $t = 1 \cdots T_{\max}$
   a) Propose partition t via Coupled Partition Subroutine
2) Compute $w_p^k = w_p^{k-1} * \frac{p(\mathbf{z}|\mathbf{X}_p)}{\prod_{t=1}^{T_p} b_{p,t}}$

Coupled Partition Subroutine for Target t

1) For each particle $p = 1,\ldots,N_{\text{part}}$,
   a) For each proposal $r = 1,\ldots,R$
      i) Sample $\mathbf{X}_{p,t}^*(r) \sim p(\mathbf{X}_{p,t}^k \mid \mathbf{X}_{p,t}^{k-1})$
      ii) Compute $w_r = p(\mathbf{z} \mid \mathbf{X}_{p,t}^*(r))$
   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, $b_{p,t} = w_j$

individually. It improves tracking performance over SIR at the expense of additional computations.

Again, target birth and death may be accounted for in the CP/IP algorithms by choosing some particles to have less or more partitions when proposing. As discussed after Table II, a model of how targets arrive and depart in the particular setting would be used to choose birth and death rates.

3) *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 AP method. The AP 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.

To determine when targets are sufficiently separated, we threshold based on distance in sensor space between the estimated state of the ith partition and the jth partition. Denote by $\hat{\mathbf{x}}_i$ the estimated x and y positions of the ith partition (30). Notice that 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 (20), where $\hat{\Sigma}_j$ is the covariance associated with the estimate of the jth partition (31)

$$r^2 = (\hat{x}_i - \hat{x}_j)' \hat{\Sigma}_j^{-1} (\hat{x}_i - \hat{x}_j). \tag{20}$$

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 than 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 V.

TABLE V
Adaptive Proposal Method

1) For each partition $t = 1 : T_{\max}$
   a) $d(t) = \min_{j \neq t} \|\hat{\mathbf{x}}_t - \hat{\mathbf{x}}_j\|$
   b) if $d(t) > \tau$
      Propose partition t using IP method
   c) else
      Propose partition t using CP method
2) For each particle $p = 1,\ldots,N_{\text{part}}$, $w_p^k = w_p^{k-1} * \frac{p(\mathbf{z}|\mathbf{X}_p)}{\prod_{t=1}^{T_p} b_{p,t}}$

D. Permutation Symmetry and Partition Sorting

As discussed throughout the preceding sections, the permutation symmetry associated with the JMPD noted in Section II 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 [21] algorithm to cluster the partitions of each particle, where the optimization is done across permutations of the particles. In practice, this is a very light computational burden. First, those partitions that are not coupled are already consistently ordered and need not be involved in the clustering procedure. Second, since this 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 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

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \ldots, \mathbf{x}_{p,T_p}]. \tag{21}$$

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

TABLE VI
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 (23)
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 $\pi = (\pi_1, \ldots, \pi_p, \ldots, \pi_{N_{\text{part}}})$ and go to 2

Under the permutation π_p is reordered to

$$\mathbf{X}_p = [\mathbf{x}_{p,\pi_p(1)}, \mathbf{x}_{p,\pi_p(2)}, \ldots, \mathbf{x}_{p,\pi_p(T_p)}]. \tag{22}$$

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

$$\bar{\mathbf{X}}_t(\pi) = \sum_{p=1}^{N_{\text{parts}}} w_p \mathbf{X}_{p,\pi_p(t)} \tag{23}$$

where it is understood that the summation is taken over only those particles that have partition t. Further, define the χ^2 statistic

$$\chi^2(\pi) = \sum_{p=1}^{N_{\text{parts}}} \sum_{t=1}^{T_p} w_p (\mathbf{X}_{p,\pi_p(t)} - \bar{\mathbf{X}}_t(\pi_p))^2. \tag{24}$$

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

$$\hat{\pi} = \min_{\pi} \chi^2(\pi). \tag{25}$$

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. The algorithm is given in Table VI.

E. Estimation

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

Equation (1) gives the expression for computing the probability that 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 $p = 1 \cdots N_{\text{parts}}$,

$$I_p(T) = \begin{cases} 1 & \text{if} \quad T_p = T \\ 0 & \text{otherwise} \end{cases}. \tag{26}$$

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

$$p(T \mid \mathbf{Z}) \approx \sum_{p=1}^{N_{\text{part}}} I_p(T) w_p. \tag{27}$$

The estimate of the probability that there are T targets in the surveillance volume is 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):

$$\tilde{I}_p(i) = \begin{cases} 1 & \text{if partition } i \text{ exists in particle } p \\ 0 & \text{otherwise} \end{cases}. \tag{28}$$

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

$$\hat{w}_p = \frac{w_p \tilde{I}_p(i)}{\sum_{l=1}^{N_{\text{part}}} \tilde{I}_l(i) w_l}. \tag{29}$$

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

$$\hat{\mathbf{X}}(i) = E[\mathbf{X}(i)] = \sum_{p=1}^{N_{\text{part}}} \hat{w}_p \mathbf{X}_{p,i} \tag{30}$$

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

$$\hat{\mathbf{\Lambda}}(i) = \sum_{p=1}^{N_{\text{part}}} \hat{w}_p (\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i))(\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i))'. \tag{31}$$

The indicator function $\tilde{I}_p(i)$ ensures that the summations in (30) and (31) 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 (30) or (31), the clustering procedure discussed in Section IIID is performed.

F. Resampling

In the traditional method of resampling, after each measurement update, N_{part} particles are selected with replacement from $\mathbf{X}_p$ based upon the particle weights w_p. The result is a set of N_{part} particles that have uniform weight which approximate the multitarget density $p(\mathbf{X}, T \mid \mathbf{Z})$. Particles that do not correspond to measurements are eliminated; in particular, particles

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

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 [3]. 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 [15]. Of these methods, we have found that adaptively choosing at which time steps to resample [40] based on the number of effective particles leads to improved performance while reducing compute time. All results presented herein use the method of [40] to determine which times to resample and use systematic resampling [3] to perform resampling. We have also found that Markov chain Monte Carlo (McMC) moves using a Metropolis-Hastings scheme [15] leads to slightly improved performance in our application.

IV. SIMULATION RESULTS

A. Introduction

We illustrate the performance of our multitarget tracking scheme by considering the following model scenario. An example 3-target episode is given in Fig. 1.

The ground targets used in this simulation move in a 5000 m $\times$ 5000 m 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 (5)) was adequate to model the behavior of the vehicles for these simulation studies and is therefore used in all experimental results presented herein. In another study [38], we have found that a multiple model particle filter with modes corresponding to nearly constant velocity, rapid acceleration, and stationarity provides more efficient filtering.

We utilize the idealized sensor described in Section IIB. The sensor scans a fixed rectangular region of 50 $\times$ 50 pixels, where each pixel represents a 100 m $\times$ 100 m area on the ground plane. The sensor returns Rayleigh-distributed measurements in each pixel, depending on the number of targets that occupy the pixel. Unthresholded measurements return energy according to (12) while thresholded measurements behave according to (13).

We present the results of 5 simulation studies here. First, in Section IVB, we illustrate the benefit of the adaptive proposal scheme detailed in Section IIIC. We contrast the performance of the CP, IP, and AP methods in two scenarios, one where targets are always well separated and one more realistic scenario where targets frequently interact. Second, in Section IVC, we quantitatively evaluate the performance difference when using unthresholded measurements versus thresholded measurements. Third, in Section IVD, we evaluate the ability of the tracker to determine the number of targets when the number is initially unknown. Fourth, in Section IVE, we investigate the computational burden of the algorithm and how it scales with target number. Fifth, in Section IVF, we illustrate partition swapping when two targets cross. The scenario is contrasted with and without partition sorting as described in Section IIID.

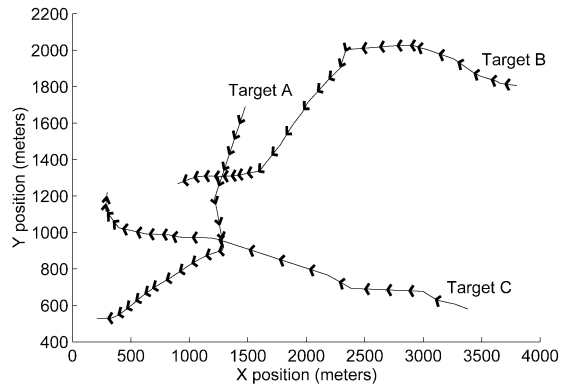


Fig. 1. Schematic showing motion of three of the ten targets in the simulation scenario. Target paths are indicated by lines, direction of travel by arrows. There are two instances where target paths cross (i.e., are at same position at same time).

B. Adaptive Proposal Results

In Fig. 2, we compare the performance of the IP (Table III), CP (Table IV), and AP (Table V) proposal schemes presented here with that of the traditional scheme of sampling from the KP (Table II), in terms of RMS tracking error. In this example we use 3 targets with motion taken from real recorded trajectories. The targets remain close in sensor space for about 50% of the time. As this simulation is designed to exhibit tracking performance, the filter is initialized with the correct number of targets and the states distributed in a Gaussian manner about the true state. Thresholded measurements with $P_d = 0.5$ are used and the SNR parameter λ is varied from 1 to 21.

Due to partition swapping, the IP method is inappropriate during target crossings and hence the tracker-only IP has poor performance. The CP method makes no assumption about the independence of the targets and therefore performs very well, although at significantly higher computational cost. Most importantly, the adaptive method, which uses IP on partitions that are independent and CP otherwise,

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

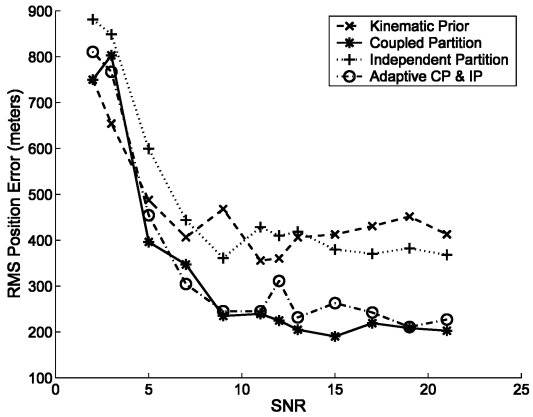


Fig. 2. Performance of CP, IP, and AP schemes in comparison to simply using KP. Performance is measured in terms of RMS position error in meters. For this simulation, we have extracted three targets from our large database of real recorded target trajectories. Targets were chosen so that they spent approximately one-half of the simulation in close proximity. IP algorithm used alone is inappropriate during target crossings and so performs poorly here. CP algorithm is always appropriate, but computationally demanding. AP algorithm adaptively switches between IP and CP resulting in good performance at reduced computation.

TABLE VII
FLOPS for KP, CP, IP, AP Methods

Method	Flops
Coupled Partition	1.25e + 8
Independent Partition	6.74e + 6
Adaptive Partition	5.48e + 7
Kinematic Prior	6.32e + 6

performs nearly as well as the CP method itself. AP achieves approximately a 50% reduction in computational burden (measured by floating point operations (FLOPS)) as compared with the CP method alone (see Table VII). For this simulation, we have extracted 3 targets from our large database of real recorded target trajectories. The targets were chosen so that they spent approximately one-half of the simulation in close proximity. The AP algorithm correctly chooses to use IP during the half of the simulation where targets are well separated and CP during the other half, which results in the stated reduction in computation.

As a means of directly comparing the IP and CP methods with the KP, we construct an alternate model problem. We consider five well-separated targets, and look at the performance in Fig. 3. For the purposes of this model problem, we restrict target motion to be linear, measurement-to-state coupling to be linear, and the noise processes to be Gaussian. We use the motion model given by (5) both for the simulation of target motion and in the filter. In this case we can use the Kalman filter as a bound. It is of course not necessary to make these assumptions for the particle filter. In fact, the strength of the particle

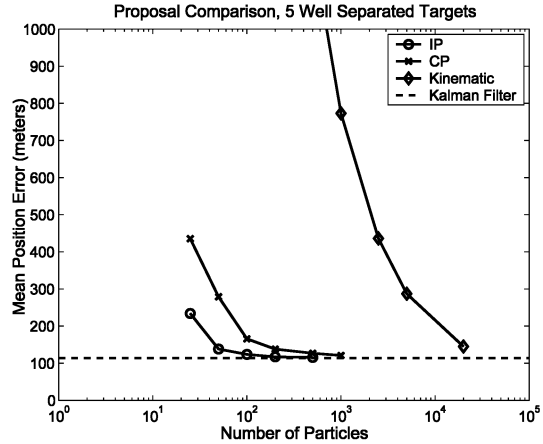


Fig. 3. Performance of CP and IP proposal schemes, in comparison to sampling from the KP. For purposes of this example, we consider five well-separated targets with linear motion and linear state-to-measurement coupling. Therefore, for the purposes of this simple example, the Kalman filter is optimal and is shown as a performance bound.

filter (and nonlinear filtering in general) approach is that no linearity/Gaussianity assumptions are needed. However, we have restricted the problem in this manner here so that we can compute an asymptotic performance bound and show that the particle filter implementation indeed reaches the bound.

More general performance bounds, which apply to the discrete time nonlinear filtering problems such as that of Fig. 2 are available in the literature [63, 54]. However, bounds require knowledge of the true target kinematics $p(\mathbf{X}^k, T^k \mid \mathbf{X}^{k-1}, T^{k-1})$ which we do not have. As stated earlier, the simulations in this paper involve real recorded target motion and hence we do not have the kinematic model precisely.

The CP method is shown with a particular choice of R, $R = 10$. We see that the CP method reduces the number of particles required (as compared with KP) by a factor of approximately 100. It was seen earlier in Table VII that the computational increase is approximately a factor of 20. This tradeoff makes CP a more efficient strategy than simply increasing the number of particles. It can be seen that the IP technique reduces the number of particles needed by between two and three orders of magnitude as compared with the traditional technique (KP). Since the work per particle to perform IP is nearly identical to that of sampling from the KP, IP actually reduces computational burden by two to three orders of magnitude when targets are well separated.

These simulations are the result of particular choices of the plant noise and measurement noise parameters. The number of particles required to reach the Kalman filter bound is sensitive to these choices. Specifically, as the ratio of plant noise to measurement noise increases, the number of particles to reach the bound increases. However, the relative performance of the IP, CP, and KP algorithms remains consistent as

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

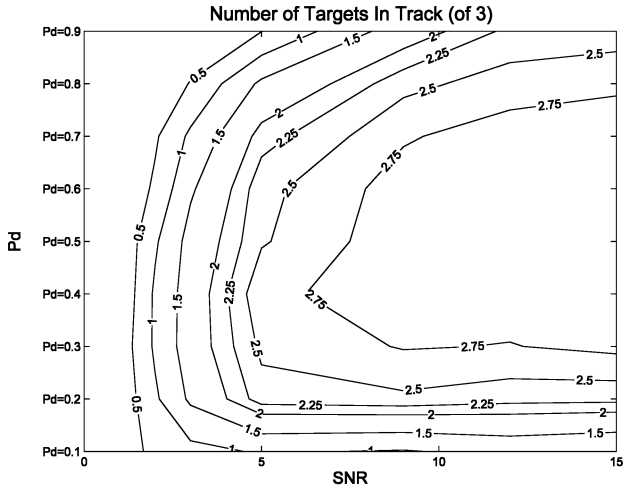


Fig. 4. Contour plot showing number of targets successfully tracked in this three-target experiment versus P_d and SNR when using thresholded measurements.

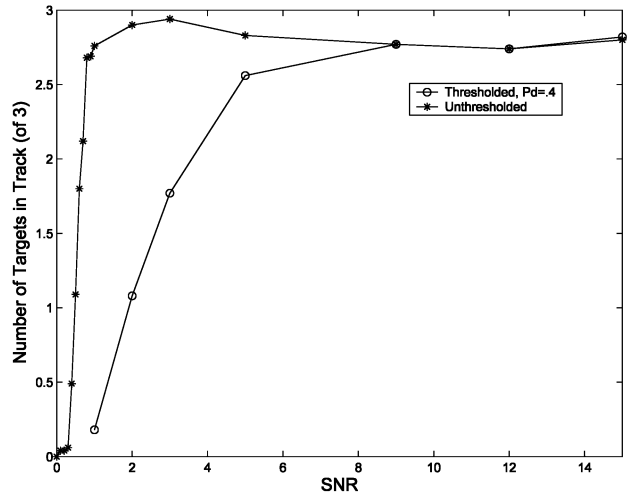


Fig. 5. Plot of the number of targets successfully tracked in the three-target experiment for thresholded measurements and unthresholded measurements as function of SNR.

the plant and measurement noise parameters change.

C. The Value of Not Thresholding

We investigate here the gain from using nonthresholded measurements in the multitarget tracking scenario. One of the strengths of our association-free method is the ability to use nonthresholded measurements.

In this simulation, we study three real targets and benchmark the performance of the tracker versus SNR (λ) for thresholded measurements with $P_d = 0.1 \cdots 0.9$. Again, the filter is initialized with the true target count and initial states distributed about the true initial target states with a small Gaussian perturbation. At a constant SNR, as the P_d is reduced, so is the P_f according to the relation given in (13). The performance of the algorithm versus SNR and P_d is given in Fig. 4.

We contrast the performance of the algorithm using thresholded measurements with the performance when using nonthresholded measurements at the same set of SNR values. Fig. 5 is a plot showing the performance of the algorithm using thresholded measurements at $P_d = 0.4$ (the best performance from Fig. 4) and the algorithm using nonthresholded measurements in terms of the number of targets successfully tracked. We see that nonthresholded measurements provide similar tracking performance at an SNR of 1 as the thresholded measurements provide at an SNR of 5, for a gain of about 7 dB from not thresholding the measurements.

D. Unknown Number of Targets

The ability of the algorithm to determine the number of targets is illustrated in Figs. 6 and 7, which

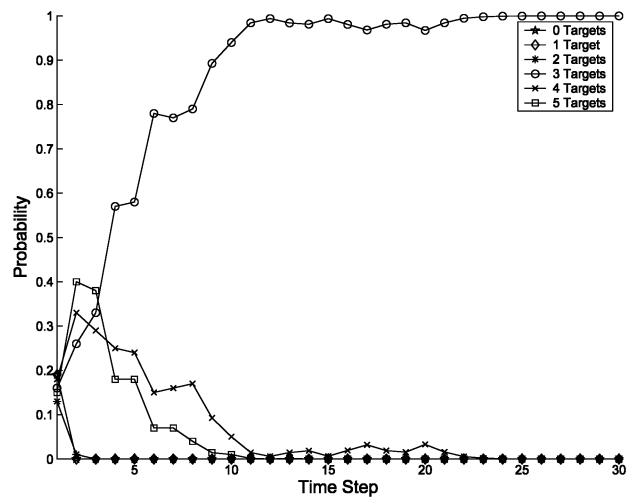


Fig. 6. Estimate of number of targets in surveillance region versus time with SNR = 4. Filter is initialized with probability uniform for $0, 1, \ldots, 5$ targets. Measurements taken over time allow filter to properly estimate number of targets in surveillance area.

show $p(T \mid \mathbf{Z})$ as a function of time for SNR = 4 and SNR = 12, respectively. There are three real targets in this simulation. We initialized the filter uniformly in target number space, allocating one-sixth the probability to $0, 1, \ldots, 5$ targets. In target state space, the filter is initialized by selecting randomly from a set of candidate positions that include the true target states and adding in a small Gaussian perturbation. Over time, the filter is able to accurately estimate the number of targets in the surveillance region. As the SNR improves, the time until correctly determining the number of targets decreases.

E. Computational Considerations

Using MatLab on an off-the-shelf 3 GHz Linux box, we find that the AP method is able to track

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

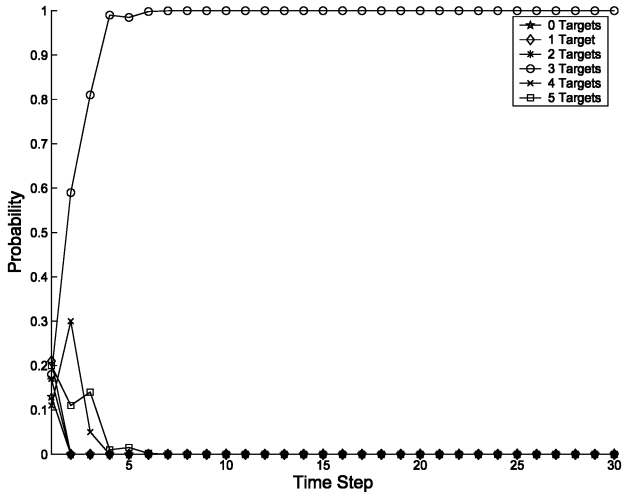


Fig. 7. Estimate of number of targets in surveillance region versus time with SNR = 12. Filter is initialized with probability uniform for $0, 1, \ldots, 5$ targets. Measurements taken over time allow filter to properly estimate number of targets in surveillance area.

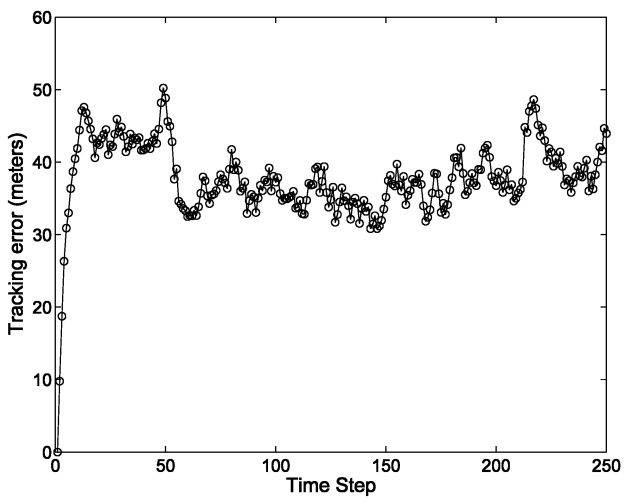


Fig. 8. Performance of particle filter implementation of JMPD when tracking ten real targets. This set of targets contains two convoys (targets following each other closely throughout the simulation), one of four targets and one of three targets. For each simulation, at each time step tracking error is measured as mean track error for the ten targets. Plot shows median tracking error across all 50 simulations. Filter is initialized with true target locations and so initial tracking error is 0. Steady state tracking error is on the order of 40 m. As mentioned earlier, the sensor measures 100 m $\times$ 100 m resolution cells on the ground. Particle filter implementation of JMPD uses 250 particles which allows near real time tracking.

10 real targets with scans of the surveillance area coming in once per second in near real time (for typical situations, the algorithm takes approximately 1.5 s to process each 1 s of data). A more low-level implementation is anticipated to easily allow real time tracking. Fig. 8 shows the tracking performance when using the particle filter implementation of JMPD on ten real targets. The plot is averaged of 50 trials, where in each trial a random set of 10 targets is chosen from our large database of real targets.

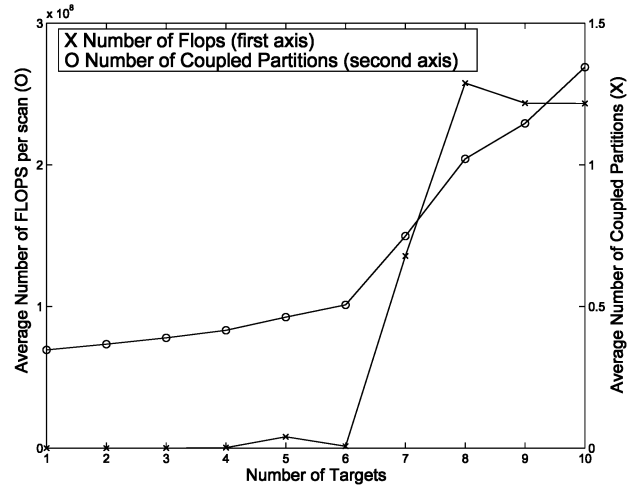


Fig. 9. FLOPS (as measured by MatLab) versus number of targets. One factor that effects computations required is number of closely spaced targets, as coupling must be modeled explicitly and CP algorithm becomes necessary. We include for reference here the average number of coupled targets over all simulations.

One factor that effects computation is the number of coupled targets. This effect can have a greater impact on computational complexity then the number of targets. When targets move close together, their coupling must be explicitly modeled and the CP algorithm becomes necessary. This algorithm is significantly more computationally demanding then the IP method.

In Fig. 9, we show the timing results of simulations where $1 \cdots 10$ targets are tracked. We include for reference the average number of coupled targets during the simulations. For each trial, we select t targets at random from our collection of real recorded data. Depending on which targets are selected, they may or may not be coupled, resulting in a different level computational complexity. The plot in Fig. 9 is averaged over 50 trials.

F. Partition Swapping

We illustrate in this section the issue of partition swapping as discussed in Section IIID. When targets are close together, measurement-to-target ambiguity may result in partitions of individual particles being reordered. In Fig. 10, we give a 9 time-step vignette which includes a target crossing. Initially, the targets are well separated and identically ordered (e.g. Time = 44) and the IP method is used for particle proposal. When the targets cross (Time = 60), partition swapping occurs and the CP method must be used. Without partition sorting using the K-means algorithm of Section IIID, this swapping persists even after the targets separated and the CP method must be used even at Time = 84. This results in an inefficient algorithm, as the CP method is more computationally demanding.

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

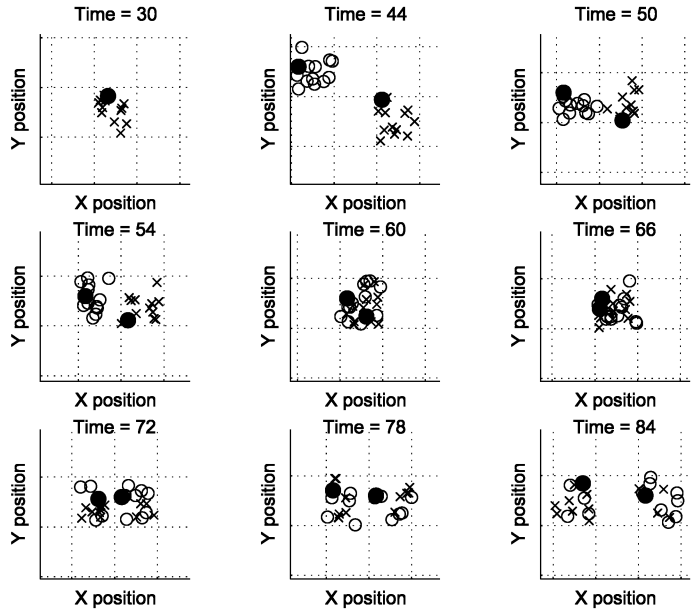


Fig. 10. This figure illustrates the phenomenon of partition swapping that occurs in direct particle filter implementation of SIR filter for JMPD. True target locations are indicated by a solid circle. At time 30 only one target is visible in plot window. At 44 s, both targets can be seen and the two partitions for each particle, plotted with x and o, are well separated. From time 60 to 66, they occupy the same detection cell. At time 84, some partition swapping has occurred, indicated by the fact that there are mixtures of x and o corresponding to each target location.

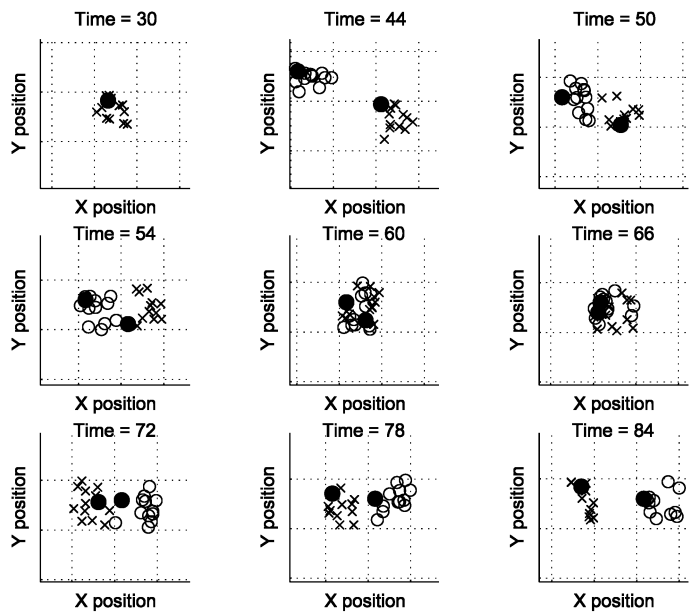


Fig. 11. This figure illustrates the same multitarget tracking scenario as in Fig. 10, except here we perform partition sorting at each time step. True target locations are indicated by a solid circle. At time 30 only one target is visible in plot window. At 44 s, both targets can be seen and the two partitions for each particle, plotted with x and o, are well separated. From time 60 to 66, they occupy the same detection cell. Targets move apart starting at time 72. Notice that the partition swapping visible in Fig. 10 at times 72–84 is avoided here because of partition sorting.

In Fig. 11 we show the same vignette as in Fig. 10, but this time we use the partition sorting algorithm outlined in Section IIID at each time step. While the CP method must still be used when the targets are occupying the same detection cell, when they move out (Time= 72) the IP method may be used again. The partition sorting allows for the more computationally efficient IP method to be used for proposal by reordering the particles appropriately.

V. CONCLUSION

This paper has presented a new grid-free implementation of a Bayesian method for tracking

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

multiple targets based on recursively estimating the JMPD. We have detailed an adaptive particle proposal scheme that explicitly takes into account the multitarget nature of the problem and automatically factors it into a series of smaller independent problems when appropriate, while properly treating the permutation symmetry and correlations that arise when targets are close together. This implementation reduces the computational burden to a reasonable level and allows implementation for realistic problems. In simulations with real target motion, we have shown the ability to track ten targets with complicated kinematic behavior, using pixelated measurements on a grid. This technique has the benefit that raw sensor measurements may be directly incorporated through the use of a likelihood function. This algorithm can process unthresholded data, obtaining a 3–6 dB effective gain compared with thresholded data. Furthermore, no measurement-to-target association is explicitly required.

REFERENCES

[1] Alspach, D. L., and Sorensen, H. W.
Nonlinear Bayesian estimation using Gaussian sum approximations.
IEEE Transactions on Automatic Control, **17**, 4 (Aug. 1972), 439–448.

[2] Arulampalam, M. S., and Ristic, B.
Comparison of the particle filter with range arameterized and modified polar EKFs for angle-only tracking.
Proceedings of SPIE, **4048** (2000), 288–299.

[3] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T.
A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking.
IEEE Transactions on Signal Processing, **50**, 2 (Feb. 2002), 174–188.

[4] Bar-Shalom, Y., and Li, X. R.
Estimation and Tracking: Principles, Techniques, and Software.
Norwood, MA: Artech House, 1993.

[5] Bar-Shalom, Y.
Multitarget Multisensor Tracking: Advanced Applications.
Norwood, MA: Artech House, 1990.

[6] Bar-Shalom, Y., and Blair, W. D.
Multitarget-Multisensor Tracking: Applications and Advances, Vol. III.
Norwood, MA: Artech House, 2000.

[7] Bergman, N.
Recursive Bayesian estimation: Navigation and tracking applications.
Ph.D. dissertation, Linköping University, Linköping, Sweden, 1999.

[8] Blackman, S. S.
Mulitple-Target Tracking with Radar Applications.
Norwood, MA: Artech House, 1986.

[9] Blair, W. D., and Brandt-Pearce, M.
Unresolved Rayleigh target detection using monopulse measurements.
IEEE Transactions on Aerospace and Electronic Systems, **34**, 2 (1998), 543–552.

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

[11] Chood, K., and Fleet, D. J.
People tracking using hybrid Monte Carlo filtering.
In *Proceedings of The International Conference on Computer Vision*, vol. 2, 2001, 321–328.

[12] Crisan, D., and Doucet, A.
A survey of convergence results on particle filtering methods for practicioners.
IEEE Transactions on Signal Processing, **50**, 3 (2002), 736–746.

[13] Dellaert, F., Fox, D., Burgard, W., and Thrun, S.
Monte Carlo localization for mobile robots.
Presented at IEEE International Conference on Robotics and Automation, May 1999.

[14] Doucet, A.
On sequential Monte Carlo methods for Bayesian filtering.
Technical report, Department of Engineering, University of Cambridge, UK, 1998.

[15] Doucet, A., de Freitas, N., and Gordon, N.
Sequential Monte Carlo Methods in Practice.
New York: Springer, 2001.

[16] Doucet, A., Vo, B-N., Andrieu, C., and Davy, M.
Particle filtering for multi-target tracking and sensor management.
Presented at The Fifth International Conference on Information Fusion, Maryland, 2002.

[17] Gonzalez-Banos, H. H., Lee, C-Y., and Latombe, J-C.
Real-time combinatorial tracking of a target moving unpredictably among obstacles.
In *Proceedings of the IEEE Conference on Robotics and Automation*, Washington, D.C., May 2002.

[18] Goodman, I., Mahler, R., and Nguyen, H.
Mathematics of Data Fusion.
Boston: Kluwer Academic Publishers, 1997.

[19] Gordon, N. J., Salmond, D. J., and Smith, A. F. M.
A novel approach to non-linear and non-Gaussian Bayesian state estimation.
IEE Proceedings on Radar and Signal Processing, **140** (1993), 107–113.

[20] Gowda, C. H., and Viswanatha, R.
Performance of distributed CFAR test under various clutter amplitudes.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 4 (1999), 1410–1419.

[21] Hastie, T., Tibshirani, R., and Friedman, J.
The Elements of Statistical Learning (Springer Series in Statistics).
New York: Springer-Verlag, 2001.

[22] Hue, C., Le Cadre, J-P., and Perez, P.
Tracking multiple objects with particle filtering.
IEEE Transactions on Aerospace and Electronic Systems, **38**, 3 (2002), 791–812.

[23] Hue, C., Le Cadre, J-P., and Perez, P.
Sequential Monte Carlo methods for multiple target tracking and data fusion.
IEEE Transactions on Signal Processing, **50**, 2 (2002), 309–325.

[24] Isard, M., and Blake, A.
Visual tracking by stochastic propagation of conditional density.
In *Proceedings of the 4th European Conference on Computer Vision*, Cambridge, England, 1996, 343–356.

[25] Isard, M., and MacCormick, J.
BraMBLe: A Bayesian multiple-blob tracker.
In *Proceedings of the 8th International Conference on Computer Vision*, 2001.

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

[26] Jazwinski, A. H.
Stochastic Processes and Filtering Theory.
New York: Academic Press, 1970.

[27] Julier, S. J., and Uhlman, J. K.
A new extension of the Kalman filter to nonlinear systems.
In *Proceedings of Aerosense: The Eleventh International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, vol. 3068, 1997, 182–193.

[28] Kamen, E. W.
Multiple target tracking based on symmetric measurement functions
IEEE Transactions on Automatic Control, **37**, 3 (1992), 371–374.

[29] Karlsson, R., and Gustafsson, F.
Monte Carlo data association for multiple target tracking.
In *Proceedings of the IEE Workshop on Target Tracking: Algorithms and Applications*, The Netherlands, 2001.

[30] Kastella, K.
Discrimination gain for sensor management in multitarget detection and tracking.
In *IEEE-SMC and IMACS Multiconference CESA '96*, vol. 1, Lille, France, July 9–12, 1996, 167–172.

[31] Kastella, K.
A maximum likelihood estimator for report-to-track association.
SPIE Proceedings, **1954** (1993), 386–393.

[32] Kastella, K.
Event-averaged maximum liklihood estimation and mean-field theory in multitarget tracking.
IEEE Transactions on Automatic Control, **40**, 6 (1995), 1070–1074.

[33] Kastella, K.
Event averaged maximum likelihood estimation and mean-field theory in multitarget tracking.
IEEE Transactions on Automatic Control, **50**, 6 (June 1995), 1070–1073.

[34] Kirubarajan, T., and Bar-Shalom, Y.
Probabilistic data association techniques for target tracking in clutter.
Proceedings of the IEEE, **92**, 3 (Mar. 2004), 536–557.

[35] Kitagawa, G.
Non-Gaussian state-space modelling of non-stationary time series.
Journal of the American Statistical Association, **82** (1987), 1032–1063.

[36] Kreucher, C., Kastella, K., and Hero, A. O.
Tracking multiple targets using a particle filter representation of the joint multitarget probability density.
Presented at the SPIE International Symposium on Optical Science and Technology, San Diego, CA, Aug. 2003.

[37] Kreucher, C., and Kastella, K.
Multiple-model nonlinear filtering for low-signal ground target applications.
In *Proceedings of The Fifteenth International Aerosense Symposium*, vol. 4380, 2001, 1–12.

[38] Kreucher, C., Hero, A. O., and Kastella, K.
Multiple model particle filtering for multi-target tracking.
In *Proceedings of the Twelfth Annual Workshop on Adaptive Sensor Array Processing*, Mar. 2004.

[39] Lee, M-S., and Kim, Y-H.
An efficient multitarget tracking algorithm for car applications.
IEEE Transactions on Industrial Electronics, **50**, 2 (2003), 397–399.

[40] Liu, J., and Chen, R.
Sequential Monte Carlo methods for dynamic systems.
Journal of the American Statistical Association, (Sept. 1998).

[41] Mahler, R. P. S.
A unified foundation for data fusion.
In *The Seventh Joint Service Data Fusion Symposium*, Laurel, MD, Oct. 25–28, 1994, 154–174.

[42] Mahler, R.
Global optimal sensor allocation.
In *Proceedings of the Ninth National Symposium on Sensor Fusion*, vol. I, Mar. 12–14, 1996, 167–172.

[43] Mahler, R.
An extended first-order Bayes filter for force aggregation.
In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, vol. 4729, 2002.

[44] Mallick, M.
Comparison of EKF, UKF, and PF for UGS and GMTI sensors.
Presneted at The 2003 Defense Science and Technology Workshop, Adelaide, Australia, July 2003.

[45] Maskell, S., Rollason, M., Gordon, N., and Salmond, D.
Efficient particle filtering for multiple target tracking with application to tracking in structured images.
In *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets, 2002*, Orlando, FL, Apr. 1–5, 2002.

[46] Maybeck, P. S.
Stochastic Models, Estimation, and Control, Vol. 1.
New York: Academic Press, 1982.

[47] Mori, S., Shong, C. Y., Tse, E., and Wishner, R. P.
Tracking and classifying multiple targets without a prior identification.
IEEE Transactions on Automatic Control, **AC-31** (May 1986).

[48] Miller, M. I., Srivastava, A., and Grenander, U.
Conditional mean estimation via jump-diffusion processes in multiple target tracking/recognition.
IEEE Transactions on Signal Processing, **43**, 11 (1995), 2678–2690.

[49] Musick, S., Kastella, K., and Mahler, R.
A practical implementation of joint multitarget probabilities.
SPIE Proceedings, **3374** (1998), 26–37.

[50] Montemerlo, M., Thrun, S., and Whittaker, W.
Conditional particle filter for simultaneous mobile robot localization and people tracking.
In *Procceedings of the IEEE Conference on Robotics and Automation*, vol. 1, 2002, 695–701.

[51] Musick, S., Greenewald, J., Kreucher, C., and Kastella, K.
Comparison of particle method and finite difference nonlinear filters for low SNR target tracking.
In *Proceedings of The Fourth Annual Conference on Information Fusion*, Montreal, Canada, 2001.

[52] Orton, M., and Fitzgerald, W.
A Bayesian approach to tracking multiple targets using sensor arrays and particle filters.
IEEE Transactions on Signal Processing, **50**, 2 (2002), 216–223.

[53] Pitt, M. K., and Shephard, N.
Filtering via simulation: Auxiliary particle filters.
Journal of the American Statistical Association, **94** (June 1999), 446, 590–599.

[54] Ristic, B., and Arulampalam, M. S.
Tracking a manoeuvring target using angle-only measurements: Algorithms and performance.
Signal Processing, **83** (2003), 1223–1238.

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

[55] Schulz, D., Fox, D., and Hightower, J.
People tracking with anonymous and ID-sensors using Rao-Blackwellised particle filter.
In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence*, 2003.

[56] Shertukde, H. M., and Bar-Shalom, Y.
Tracking of crossing targets with imaging sensors.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 4 (1991), 582–592.

[57] Sidenbladh, H., and Wirkander, S-L.
Particle filtering for random sets.
IEEE Transactions on Aerospace and Electronic Systems, to be published.

[58] Sidenbladh, H., and Wirkander, S-L.
Tracking random sets of vehicles in terrain.
Presented at the IEEE Workshop on Multi-Object Tracking, Madison, WI, 2003.

[59] (Special issue on particle filtering methods).
IEEE Transactions on Signal Processing, **50**, 2 (2002).

[60] Stimson, G. W.
Introduction to Airborne Radar (2nd ed.).
Rayleigh, NC: SciTech Publishing, 1998.

[61] Stone, L. D., Barlow, C. A., and Corwin, T. L.
Bayesian Multiple Target Tracking.
Norwood, MA: Artech House, 1999.

[62] Streit, R. L., and Luginbuhl, T. E.
Maximum likelihood methods for probabilistic multi-hypothesis tracking.
Proceedings of SPIE, **2235** (Apr. 5–7, 1994).

[63] Tichavsky, P., Muravchik, C., and Nehorai, A.
Posterior Cramer-Rao bounds for discrete-time nonlinear filtering.
IEEE Transactions on Signal Processing, **46**, 5 (May 1998), 1386–1396.

[64] Tullsson, B. E.
Monopulse tracking of Rayleigh targets: A simple approach.
IEEE Transactions on Aerospace and Electronic Systems, **27**, 3 (1991), 520–531.

[65] Tweed D., and Calway, A.
Tracking multiple animals in wildlife footage.
In *Proceedings of the Conference on Pattern Recognition*, vol. 2, 2002, 24–27.

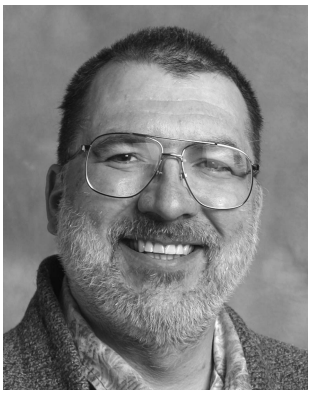
[66] Vermaak, J., and Blake, A.
Nonlinear filtering for speaker tracking in noisy and reverberant environments.
In *Proceedings of International Conference on Acoustics Speech and Signal Processing*, vol. V, 2001, 3021–3024.

[67] Vo, B-N., Singh, S., and Doucet, A.
Sequential Monte Carlo implementatoin of the PHD filter for multi-target tracking.
In *Proceedings of the Sixth International Conference on Information Fusion*, July 2993, 792–799.

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.

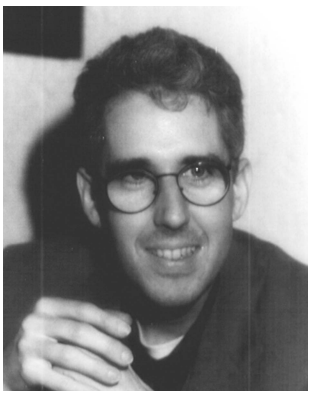
Chris Kreucher received the B.S. and M.S. degrees from the University of Michigan–Dearborn (1997 and 1998, respectively) and the Ph.D. degree from the University of Michigan–Ann Arbor, in 2005, all in electrical engineering.

Since 1997, he has been a research scientist at General Dynamics Advanced Information Systems' Michigan Research & Development Facility (formerly known as ERIM). His current research interests include Bayesian filtering (specifically multitarget tracking) and sensor management.



Keith Kastella received a B.A. in physics in 1980 from Reed College, Portland OR, and the M.S. and Ph.D. degrees from the State University of New York at Stony Brook (SUNY-SB) in 1985 and 1988, respectively.

He is a chief scientist at General Dynamics Advanced Information Systems in Ypsilanti, MI. His current research interests are in data fusion, signal processing and novel applications of quantum entanglement in sensing and signal processing.



Alfred O. Hero, III (S'79—M'80—SM'96—F'98) received the B.S. (summa cum laude) from Boston University (1980) and the Ph.D from Princeton University (1984), both in electrical engineering.

Since 1984 he has been with the University of Michigan, Ann Arbor, where he is a professor in the Department of Electrical Engineering and Computer Science, and by courtesy, in the Department of Biomedical Engineering and the Department of Statistics. His recent research interests have been in areas including: inference on sensor networks, bioinformatics, and statistical signal and image processing.

Dr. Hero has received an IEEE Signal Processing Society Meritorious Service Award (1998), an IEEE Signal Processing Society Best Paper Award (1998), and the IEEE Third Millenium Medal (2000). He is currently president-elect of the IEEE Signal Processing Society (2004–2005) and associate editor of the *IEEE Transactions on Computational Biology and Bioinformatics* (2004–2006).

Authorized licensed use limited to: University of Michigan Library. Downloaded on May 5, 2009 at 16:01 from IEEE Xplore. Restrictions apply.