

Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density

Chris Kreucher^a, Keith Kastella^a, and Alfred O. Hero III^b

^aVeridian's Ann Arbor Research and Development Center, Ann Arbor, MI, USA

^bThe University of Michigan Department of Electrical Engineering and Computer Science,
Ann Arbor, MI, USA

ABSTRACT

This paper addresses the problem of tracking multiple moving targets by estimating their joint multitarget probability density (JMPD). The JMPD technique is a Bayesian method for tracking multiple targets that allows nonlinear, non-Gaussian target motions and measurement to state coupling. JMPD simultaneously estimates both the target states and the number of targets. In this paper, we give a new grid-free implementation of JMPD based on particle filtering techniques and explore several particle proposal strategies, resampling techniques, and particle diversification methods. We report the effect of these techniques on tracker performance in terms of tracks lost, mean squared error, and computational burden.

Keywords: Nonlinear Filtering, Particle Filtering, Multi-target Tracking

1. INTRODUCTION

The problem of multitarget tracking is to estimate the state of a collection of targets in a surveillance region given a set of noisy measurements, \mathbf{Z} . One way to characterize the states of a collection of targets is to use Bayesian methods to construct the joint multitarget probability density (JMPD) $p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T-1}, \mathbf{x}_T | \mathbf{Z})$, which is the density for T targets having states $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T-1}, \mathbf{x}_T$ conditioned on the set of measurements \mathbf{Z} . In principle, the JMPD can be recursively updated as new measurements come in using standard Bayesian methods, namely solving the prediction and update equations.⁴

In the Kalman Filter case, a set of analytical expressions for filter recursions are available. However, in the more general nonlinear filtering problem where states may evolve nonlinearly, measurements may be nonlinearly coupled to states, and densities may be non-Gaussian, no such analytical solution exists. In the past, grid based techniques^{5,6} have been successfully used to show the potential of nonlinear filtering, but have ultimately proven to be too computationally intensive to extend to more realistic scenarios.

We present here a new grid-free implementation of JMPD based on particle filtering techniques. We find that this method allows us to break the computational logjam involved in grid based solutions, and to tackle more realistic problems. In the sequel to this paper⁷ we show how this particle filter based multitarget tracking technique allows us to address the interesting problem of sensor allocation.

The paper is organized as follows. In Section 2, we present the conceptual details of our multitarget tracking algorithm. Specifically, we give the details of JMPD and examine the numerical difficulties involved in directly implementing JMPD on a grid. In Section 3, we present a particle filter (PF) based implementation of JMPD. We see that this provides for computationally tractable implementation, allowing for applicability in more realistic scenarios. We investigate several particle proposal strategies, resampling techniques, and particle diversification methods and report their effect on tracker performance in terms of tracks lost, mean squared error, and computational burden. We give simulation results on a problem involving ten moving targets in Section 4. Finally, we provide some thoughts on future direction in Section 5.

Please Direct correspondence to Christopher.Kreucher@veridian.com

2. THE JOINT MULTITARGET PROBABILITY DENSITY (JMPD)

In this section, we introduce the details of using the Joint Multitarget Probability Density (JMPD) for target tracking. The concept of JMPD was first discussed by Kastella⁴ where a method of tracking multiple targets that moved between discrete cells on a line was presented. We generalize the discussion here to deal with targets that have N -dimensional continuous valued state vectors and arbitrary kinematics. In the model problems, we are interested in tracking the position (x, y) and velocity (\dot{x}, \dot{y}) of multiple targets and so we describe each target by the four dimensional state vector $[x, \dot{x}, y, \dot{y}]'$.

JMPD provides a means for tracking an unknown number of targets in a Bayesian setting. The statistics model uses the joint multitarget conditional probability density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k | \mathbf{Z}^k)$ as the probability density for exactly T targets with states $\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ at time k based on a set of observations \mathbf{Z}^k . The number of targets T is a variable to be estimated simultaneously with the states of the T targets. The observation set \mathbf{Z}^k refers to the collection of measurements up to and including time k , i.e. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, \dots, \mathbf{z}^k\}$, 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}_i in the density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k | \mathbf{Z}^k)$ is a vector quantity and may (for example) be of the form $[x, \dot{x}, y, \dot{y}]'$. We refer to each of the T target state vectors $\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ as a partition of the state \mathbf{X} . For convenience, the density will be written more compactly in the traditional manner as

$$p(\mathbf{X}^k | \mathbf{Z}^k) \quad (1)$$

Equation (1) implies that the state-vector \mathbf{X} represents a variable number of targets each possessing their own state vector. As an illustration, some examples illustrating the sample space of p are

$$\begin{aligned} p(\emptyset | \mathbf{Z}), & \text{ the posterior probability density for no targets in the surveillance volume} \\ p(\mathbf{x}_1 | \mathbf{Z}), & \text{ the posterior probability density for one target in state } \mathbf{x}_1 \\ p(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{Z}), & \text{ the posterior probability density for two targets in states } \mathbf{x}_1 \text{ and } \mathbf{x}_2 \\ p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 | \mathbf{Z}), & \text{ the posterior probability density for three targets in states } \mathbf{x}_1, \mathbf{x}_2 \text{ and } \mathbf{x}_3 \end{aligned}$$

Since the targets are not distinguishable on the basis of the measurements available to the tracker, this density must be symmetric under permutation of the target indices. When the targets are widely separated in the sensor's measurement space, each target's measurements can be uniquely associated with it, and the joint multitarget conditional density factorizes. In this case, the problem may be treated as a collection of single target trackers. The characterizing feature of multitarget tracking is that in general some of the measurements have ambiguous associations, and therefore the conditional density does not factorize.

The temporal update of the posterior likelihood on this density proceeds according to the usual rules of Bayesian filtering. Given a model of how the JMPD evolves overtime, $p(\mathbf{X}^k | \mathbf{X}^{k-1})$, we may compute the time-updated or prediction density via

$$p(\mathbf{X}^k | \mathbf{Z}^{k-1}) = \int d\mathbf{X}^{k-1} p(\mathbf{X}^k | \mathbf{X}^{k-1}) p(\mathbf{X}^{k-1} | \mathbf{Z}^{k-1}) \quad (2)$$

The time evolution of the JMPD may simply be a collection of target kinematic models, or as we will discuss later, may involve target birth and death. Bayes rule is used to update the posterior density as new measurements \mathbf{z}^k arrive as

$$p(\mathbf{X}^k | \mathbf{Z}^k) = \frac{p(\mathbf{z}^k | \mathbf{X}^k) p(\mathbf{X}^k | \mathbf{Z}^{k-1})}{p(\mathbf{z}^k | \mathbf{Z}^{k-1})} \quad (3)$$

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 practice, the sample space of \mathbf{X}^k is very large. It contains all possible configurations of state vectors \mathbf{x}_i for all possible values of T . The original formulation of JMPD given by Kastella⁴ approximated the density by discretizing on a grid. It was immediately found that 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, the number grid cells needed grows as $Locations^{Targets}$, where $Locations$ is the number of discrete locations the targets may occupy and $Targets$ is the number of targets.

Thus, we need a method for approximating the JMPD that leads to more tractable computational burden. In the next section, we show that the Monte Carlo methods collectively known as particle filtering break this computational barrier.

3. PARTICLE FILTER IMPLEMENTATION OF JMPD

We find that a particle filter based implementation of JMPD breaks the computational logjam and allows us to investigate more realistic problems. To implement JMPD via a particle filter (PF), we first approximate the joint multitarget probability density $p(\mathbf{X}|\mathbf{Z})$ by a set of N_{part} weighted samples, \mathbf{X}_p , ($p = 1 \dots N_{part}$):

$$p(\mathbf{X}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \quad (4)$$

Here we have suppressed the time superscript k everywhere for notational simplicity. We will do this whenever time is not relevant to the discussion at hand.

Particle filtering is then simply a method of solving the prediction and update equations given in the previous section by simulation. Samples from a density are used to represent the density and to propagate it through time. The prediction equation (eq. 2) is solved by proposing new particles from the existing set of particles using a model of state dynamics and perhaps the measurements. The update equation (eq. 3) is solved by assigning a weight to each of the particles that have been proposed using the measurements and perhaps the model of state dynamics.

The specific details of the PF implementation are as follows. Recall first from Section 2 that our multitarget state vector \mathbf{X} has T partitions, each corresponding to a target written explicitly in equation (5):

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{T-1}, \mathbf{x}_T] \quad (5)$$

Furthermore, the joint multitarget probability $p(\mathbf{X}|\mathbf{Z})$ is defined for $T = 0 \dots \infty$. Each of the particles \mathbf{X}_p , $p = 1 \dots N_{part}$ is a sample drawn from the JMPD $p(\mathbf{X}|\mathbf{Z})$. Therefore, a particle \mathbf{X}_p may have $0, 1, \dots \infty$ partitions, each partition corresponding to a different target. We will denote the number of partitions in particle \mathbf{X}_p by n_p , where n_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.

To make our notation more concrete, assume that a particular particle, \mathbf{X}_p , is tracking n_p targets. Then \mathbf{X}_p has n_p partitions and will be given by

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \dots, \mathbf{x}_{p,n_p}] \quad (6)$$

In the case where each partition (target) is modelled using the state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]'$, the particle will have n_p partitions each of which has 4 components:

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \dots, \mathbf{x}_{p,n_p}] = \begin{pmatrix} x_{p,1} & x_{p,2} & \dots & x_{p,n_p} \\ \dot{x}_{p,1} & \dot{x}_{p,2} & \dots & \dot{x}_{p,n_p} \\ y_{p,1} & y_{p,2} & \dots & y_{p,n_p} \\ \dot{y}_{p,1} & \dot{y}_{p,2} & \dots & \dot{y}_{p,n_p} \end{pmatrix} \quad (7)$$

Where here we expand the notation a bit and use $x_{p,1}$ to denote the x position estimate that particle p has of target 1.

Notice that this method differs from traditional particle filter tracking algorithms where a single particle corresponds to a single target. We find that when each particle is attached to a single target, some targets become particle starved over time. All of the particles tend to attach to the target receiving the best measurements. Our method explicitly enforces the multitarget nature of the problem by encoding in each particle the estimate of the number of targets and the states of those targets. This technique helps to alleviate the particle starvation issue, ensuring that all targets are represented by the particles. It is also valuable in estimating the number of targets in the surveillance region.

We illustrate the particle filter representation of JMPD in Figure 1 by setting $n_p = 3$ for all particles, and showing the prediction and update simulations graphically. Each of the three targets is characterized by the four dimensional state $\mathbf{x} = [x, \dot{x}, y, \dot{y}]'$, but only the x and y components are plotted in Figure 1. Each particle which is used to approximate the JMPD contains a partition corresponding to *each* of the three targets, and hence will have 12 dimensions.

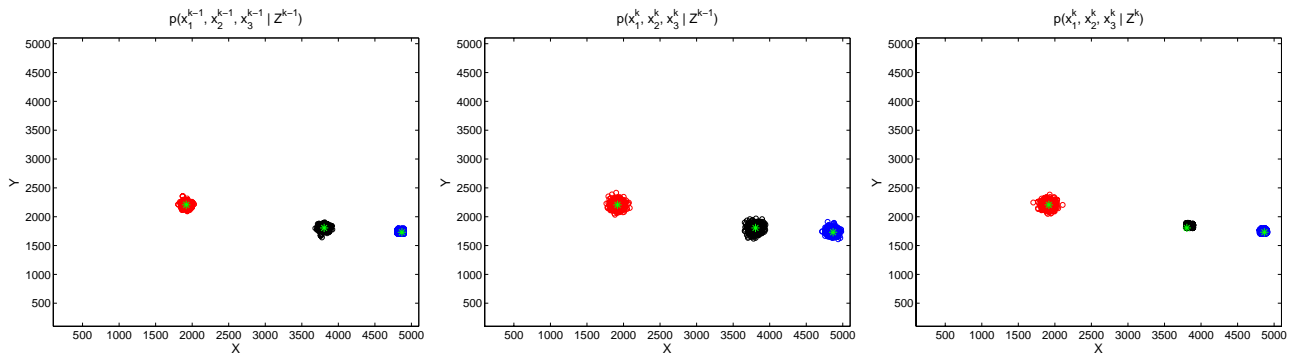


Figure 1. Left: Particles Representing $p(\mathbf{X}^{k-1}|\mathbf{Z}^{k-1})$. Middle: Particles Representing $p(\mathbf{X}^k|\mathbf{Z}^{k-1})$, formed by simulating the prediction equation (eq. 2). Right: Particles Representing $p(\mathbf{X}^k|\mathbf{Z}^k)$, formed by simulating the measurement update equation (eq. 3). Note that each particle contains a sample from each of the three targets

The permutation symmetry discussed in Section 2 is directly inherited by the particle filter representation. 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. This permutation symmetry must be dealt with carefully in the particle proposal process (Section 3.2), and estimation of target positions (Section 3.5).

In the following subsections, we detail the particle filter implementation of JMPD.

3.1. Initialization

As this is primarily a target tracking application, we typically assume that an estimate of the actual ground truth is available at time 0. To this end, we typically initialize a small set of particles (e.g. 10%) to contain the true target states at time 0 and randomly assign values (both target states and number of targets) to the rest of the particles.

Alternatively, we have successfully employed the following detection scheme. All particles are initialized with $n_p = 0$ (i.e. believing there are 0 targets in the surveillance region). For the first t time steps the algorithm is in detection mode and the sensor is scheduled to periodically scan the surveillance area. As targets are detected, particles are mutated to be consistent with the detection. As our method allows for particles to be of different dimension (i.e. different particles may be witness to different numbers of targets in the surveillance area), this is a natural method of detection. Over time, those particles with the correct number of targets will be consistent with measurements and survive to approximate the density.

3.2. Particle Proposal

We have investigated several methods of particle proposal. The standard method used, which we will refer to as sampling from the Kinematic Prior, proposes new particles at time k , \mathbf{X}_p^k , according to the traditional Sampling-Importance Resampling (SIR) method. For each particle at time $k - 1$, \mathbf{X}_p^{k-1} , a new particle \mathbf{X}_p^k is generated by simply sampling from the kinematic prior $p(\mathbf{X}^k|\mathbf{X}^{k-1})$. Notice that the state \mathbf{X}_p represents a collection of targets and therefore we must time update each of the targets.

This method has the benefit that it is simple to implement and is computationally inexpensive. However, one obvious drawback is that it does not 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.

To overcome these deficiencies, we have investigated alternative particle proposal techniques, all of which are developed as a means of biasing the proposal process towards the measurements. These techniques are collectively called multi-partition proposal (MPP) strategies. The MPP strategies propose each partition (target) in a particle separately, and form new particles as the combination of the proposed partitions.

In this manner, particles are herded towards the correct location of state space. Both of these measurement-aided techniques still rely on the kinematic prior for proposing particles and so all proposed particles are consistent with the model of target kinematics. The results of this study are detailed in sections 3.2.1 through 3.2.4.

In any of these methods, target birth and death may be accounted for in a straightforward manner by modifying the proposal density to incorporate a probability that the proposed particle \mathbf{X}_p^k has either fewer or more targets than \mathbf{X}_p^{k-1} . In particular, assume a death rate 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 , a target is removed from particle p and the updated number of targets in this particle is set as $n_p^k = n_p^{k-1} - 1$. 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 in this particle is updated to $n_p^k = n_p^{k-1} + 1$.

3.2.1. Independent-Partition (IP) Method

The independent partition (IP) method given by Orton⁹ is a convenient way to propose particles when part or all of the joint multitarget density factorizes. When applicable, we apply the Independent-Partition (IP) method to propose new partitions independently as follows. For a partition i , each particle at time $k - 1$ has its i^{th} partition proposed via the Kinematic prior and weighted by the measurements. From this set of N_{part} weighted estimates of the state of the i^{th} target, we select N_{part} samples with replacement to form the i^{th} partition of the particles at time k .

Recall that in our paradigm a partition corresponds to a target. See equation (7) for a concrete example of a particle and its partitions. A particle is then built by combining the individual partitions selected and reweighting correctly. Notice that when this method is used an additional term is introduced into the weighting process given in Section 3.3. Pseudocode for this algorithm is given in the table below.

Table 1. Independent Partition Algorithm

```

For each partition  $i, (i = 1...T)$ 
    Propose the  $i^{th}$  partition of all  $N_{part}$  particles at time  $k - 1$  to time  $k$  via Kinematic prior
    Weight the proposed partitions via the measurements from time  $k$ 
    Sample  $N_{part}$  target states from this set to form  $i^{th}$  partition of particles at time  $k$ 
end
Weight particles appropriately, to accommodate for the biased sampling

```

It is important to carefully account for the permutation symmetry issue discussed in Section 3 here. The IP method assumes that the i^{th} partition of each particle corresponds to the same target. Therefore the partitions in each particle must be sorted before this method is applied.

In the case of well separated targets, this method allows us to track many targets with the same number of particles as 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.

3.2.2. Coupled Partition (CP) Proposal Method

When targets are close together in sensor space, we say that the corresponding partitions are coupled. In these instances, the IP method is no longer applicable, and another method of particle proposal such as Coupled Partitions (CP) must be used.

We apply the coupled partitions method as follows. To propose partition i of particle p , CP proposes M possible realizations of the future state using the Kinematic prior. The M proposed futures are then given weights according to the current measurements and a representative is selected. This process is repeated for each particle until the i^{th} partition for all particles has been formed. The pseudocode for this algorithm is given in the table below.

Table 2. Coupled Partition Algorithm

```

For each partition  $i, (i = 1...T)$ 
  For each particle  $p, (p = 1...N_{parts}),$ 
    Propose  $M$  possible futures  $\mathbf{x}_m^* (m = 1...M)$  for partition  $i$  of particle  $p$  via Kinematic prior
    Weight each of the  $M$  possible futures via the measurements
    Select a representative  $\mathbf{x}$  from the  $\mathbf{x}_m^*$ 
    Set  $\mathbf{x}_{p,i}^k$ , the  $i^{th}$  partition of the  $p^{th}$  particle at time  $k$ , to  $\mathbf{x}$ 
  end
end
end

```

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

3.2.3. Comparison of IP and CP Methods

In Figure 2, we compare the performance of the two proposal schemes presented here with that of the traditional scheme of sampling from the Kinematic Prior. This model problem is constructed to have five well separated targets moving in four dimensions (position and velocity for each x and y as outlined earlier). 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. In this case we can use the Kalman Filter as a bound. Note that it is not necessary to make these assumptions for the PF. In fact, the strength of the PF (and nonlinear filtering in general) is that no linearity/Gaussian assumptions are needed. However, we have restricted the problem in this manner here in order to provide an asymptotic performance bound and show that the PF implementation indeed reaches the bound.

The CP method is shown with a particular choice of M , $M = 25$. It can be seen that the IP technique reduces the number of particles needed by between two and three orders of magnitude as compared to the traditional technique. Since the work per particle to perform IP is nearly identical to that of sampling from the kinematic prior, IP actually reduces computational burden by two to three orders of magnitude when targets are well separated.

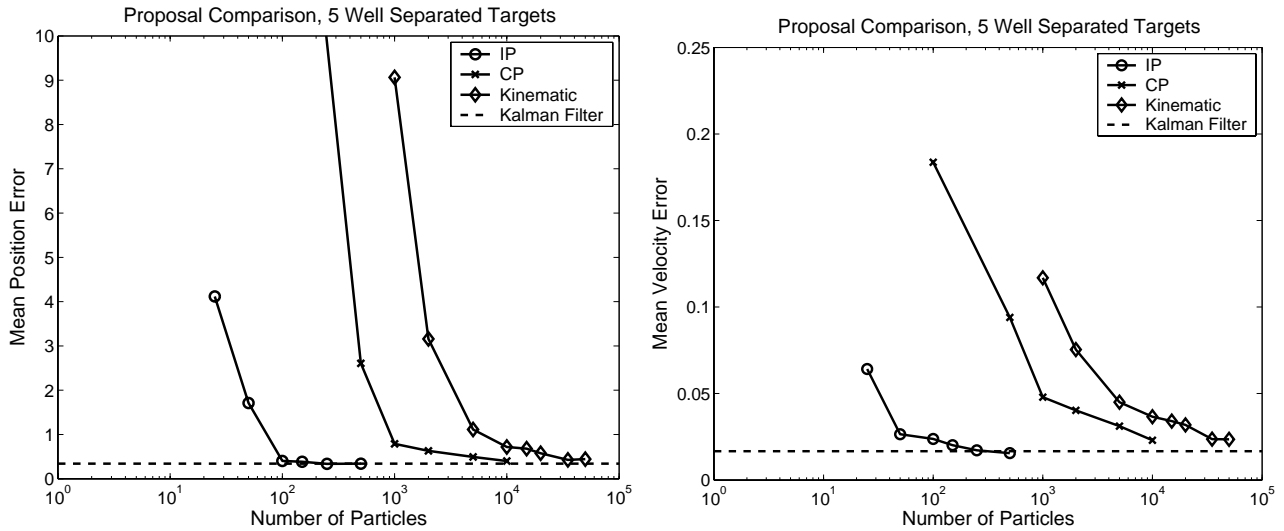


Figure 2. The Performance of the Two Proposal Schemes Used Here, In Comparison to Sampling from the Kinematic Prior. In this simple example, the Kalman Filter is optimal and is shown as a performance bound.

3.2.4. Adaptive Particle Proposal Method

In order to mitigate the problem of additional computational cost of the CP method, and the problems with the IP method when targets are close together, we propose a hybrid solution, called the Adaptive-Partition (AP) method. The adaptive-partition method again considers each partition separately. Those partitions that are sufficiently well separated 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.

The issue of determining when targets are sufficiently separated is settled by computing distances between the estimated centers of the i^{th} partition and the j^{th} partition. We have computed this distance using a simple distance between the estimated centers, and the Mahalanobis metric (eq. 8), where $\hat{\Sigma}_j$ is the covariance associated with the estimate of the j^{th} partition (see equation 21).

$$r^2 = (\hat{x}_i - \hat{x}_j)' \hat{\Sigma}_j^{-1} (\hat{x}_i - \hat{x}_j) \quad (8)$$

In practice, it is found that simply using the distance between estimated centers is sufficient and less computationally burdensome.

Figure 3 gives a comparison of the adaptive-partition methods with the IP, CP, and Kinematic proposal methods in terms of mean-squared track error and percentage of tracks lost. The model problem used to generate these results is truly a nonlinear filtering problem. The problem involves three simulated targets which cross paths several times during the simulation, and remain close in sensor space for about 50% of the time. The measurements couple to the states of the targets in a nonlinear manner.

As expected, since the IP method is inappropriate during target crossings, the performance of the tracker using IP exclusively is poor. The CP method makes no assumption about the independence of the targets and therefore performs very well, with significant computational cost. Most importantly, the adaptive method, which uses IP when appropriate and CP otherwise, performs nearly as well as the CP method itself, at about a 50% reduction in flops as compared to the CP method. This indicates that during the 50% of the simulation where targets were well separated IP was used, and during the time where targets were close, CP was used.

3.3. Measurement Update

Regardless of the particle proposal scheme employed, the proposed multitarget particles must be weighted. Each proposed particle is given a weight according to its agreement with the measurements, the kinematic model,

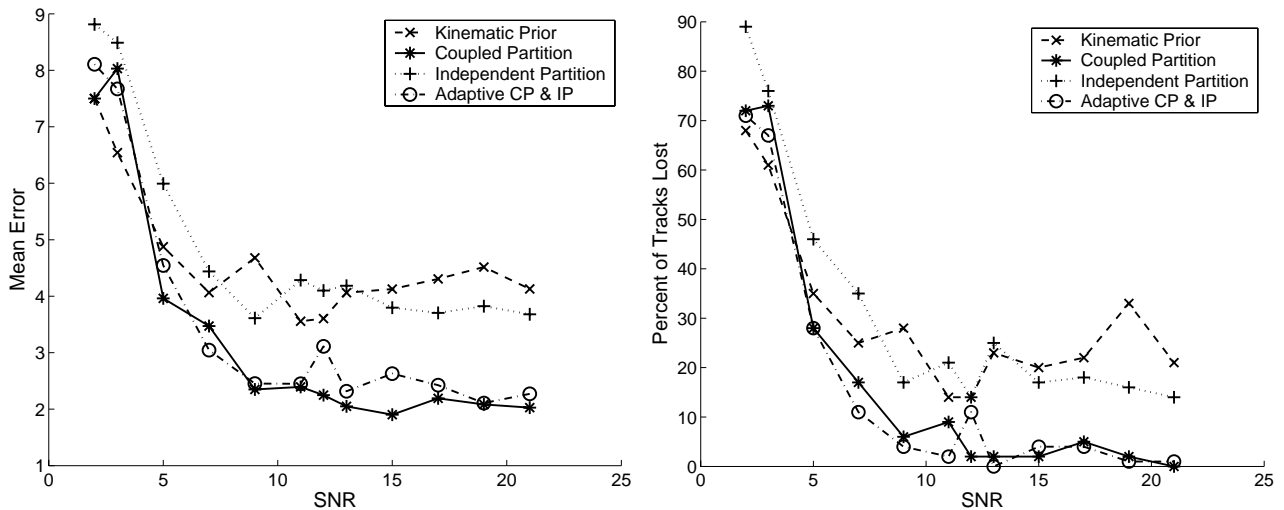


Figure 3. The Performance of the Two Proposal Schemes Used Here, In Comparison to Simply Using the Kinematic Prior.

and the importance density.¹ When proposing particles based on the Kinematic prior $p(\mathbf{X}^k|\mathbf{X}^{k-1})$, the proper weighting is given by

$$w_p \propto p(\mathbf{z}|\mathbf{X}_p^k) \quad (9)$$

Each particle \mathbf{X}_p^k simultaneously postulates that a specific number of targets exist in the surveillance region (n_p) and that the target states are given by $[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_p-1}, \mathbf{x}_{n_p}]$. In the case where the measurement set is made up of a scan i cells (say for example on a XY grid) where the measurement in each cell is independent of the measurements in the other cells, the weight becomes

$$w_p \propto \prod_i p(z_i|\mathbf{X}_p) \quad (10)$$

where in this notation z_i refers to the measurement made in cell i . A particular particle \mathbf{X}_p will postulate that there are targets in some cells i_x (not necessarily distinct):

$$i_x = i_1, i_2, \dots, i_{n_p} \quad (11)$$

We denote the measurement density when there are 0 targets present as p_0 , and simplify the weight equation as

$$w_p \propto \prod_{i \notin i_x} p_0(z_i) \prod_{i \in i_x} p(z_i|\mathbf{X}_p) \quad (12)$$

$$w_p \propto \prod_i p_0(z_i) \prod_{i \in i_x} \frac{p(z_i|\mathbf{X}_p)}{p_0(z_i)} \quad (13)$$

$$w_p \propto \prod_{i \in i_x} \frac{p(z_i|\mathbf{X}_p)}{p_0(z_i)} \quad (14)$$

If we let $O_{i,p}$ (the occupation number) denote the number of targets that a particle p postulates exist in cell i , then we write the weight as

$$w_p \propto \prod_{i \in i_x} \frac{p(z_i | O_{i,p})}{p_0(z_i)} \quad (15)$$

3.4. 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 weights w_p . What results is a set of N_{part} particles having uniform weight that approximate the density $p(\mathbf{X}|\mathbf{Z})$. At this step, particles that do not correspond to measurements are eliminated – in particular, particles that have an n_p that is not supported by measurements are not selected.

The particular resampling that we have implemented is called systematic resampling.¹ We like this scheme because it is easy to implement, runs in $O(N)$, is unbiased, and minimizes Monte Carlo variance. Many other resampling schemes and modifications are presented in the literature.³ Of these methods, we have found that adaptive resampling,⁸ also known as resample-when-needed, leads to improved performance while reducing compute time. All results presented in this work use adaptive resampling. We have also found that Markov Chain Monte Carlo (MCMC) moves using a Metropolis-Hastings scheme³ leads to slightly improved performance in our application.

3.5. Estimation

Estimates of various interesting quantities may be easily made using the particles. Estimation is best performed before resampling, as resampling has been shown to only increase the variance of the estimate.

To compute the probability that there are exactly n targets in the surveillance volume, first define the indicator variable

$$I_p = \begin{cases} 1 & \text{if } n_p = n \\ 0 & \text{otherwise} \end{cases} \quad (16)$$

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

$$p(n|\mathbf{Z}) = \sum_{p=1}^{N_{part}} I_p w_p \quad (17)$$

So to estimate the probability of n targets in the surveillance volume, we sum up the weights of the particles that have n partitions. Note that the weights are normalized to sum to 1 for equations in this section.

To compute the estimated state and covariance of target i , we first define a second indicator variable \tilde{I}_p that indicates if particle p has a partition corresponding to target i :

$$\tilde{I}_p = \begin{cases} 1 & \text{if } n_p \geq n \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Furthermore, we define the normalized weights to be

$$\hat{w}_p = \frac{w_p \tilde{I}_p}{\sum_{l=1}^{N_{part}} \tilde{I}_l w_l} \quad (19)$$

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

$$\hat{\mathbf{X}}(i) = E[\mathbf{X}(i)] = \sum_{p=1}^{N_{part}} \hat{w}_p \mathbf{X}_{p,i} \quad (20)$$

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

$$\hat{\mathbf{\Lambda}}(i) = \sum_{p=1}^{N_{part}} \hat{w}_p (\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i)) (\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i))' \quad (21)$$

The indicator function \tilde{I}_p ensures that the summations in (20) and (21) are taken over only those particles that are tracking target i . The permutation symmetry issue mentioned in Section 3 earlier comes to the forefront here. Notice that 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 equation (20) or equation (21) can be made, a sorting of the partitions in each particle must be accomplished.

4. SIMULATION RESULTS

We illustrate the performance of our multitarget tracking scheme by considering the following model problem. There are ten targets moving in a $5000m \times 5000m$ surveillance area. Each target is modelled using the four-dimensional state vector $[x, \dot{x}, y, \dot{y}]'$. Target trajectories for the simulation come directly from a set of recorded data based on GPS measurements of vehicle positions over time collected as part of a battle training exercise at NTC. Targets routinely come within sensor resolution (i.e. cross). Therefore, there is a track to measurement ambiguity, which is handled automatically by JMPD because here is no measurement to track assignment done in JMPD.

The filter assumes constant velocity motion with a large diffusive component as the model of target kinematics. This model is severely at odds with the actual target behavior which contains sudden accelerations and move-stop-move behavior. We use 500 particles, each of which is tracking the states of all ten targets, and therefore each particle has 40 dimensions. The adaptive particle proposal scheme described earlier in Section 3.2.4 is used.

At each time step, an imager is able to measure the entire surveillance area by making measurements on a grid with $100m \times 100m$ detection cell resolution. The sensor is at a fixed location above the targets and all cells are always visible to the sensor. When measuring a cell, the imager returns either a 0 (no detection) or a 1 (detection) governed by P_d , P_f , and SNR . This model is known by the filter and used to evaluate equation (3). In this illustration, we take $P_d = 0.5$, $SNR = 10dB$, and $P_f = P_d^{(1+SNR)}$, which is a standard model for thresholded detection of Rayleigh returns.

Shown in Figures 4 through 7 are 12 snapshots of a 1000 time step run of the above described simulation, where each time step corresponds to one second of actual time. The actual target positions are indicated with an asterisk. Covariance ellipses derived from the partitions of the particles are given as well. This simulation runs in about one hour in MatLab on a 2.4GhZ Linux Box, which is about three times real time.

We find that even in this challenging scenario where targets exhibit move-stop-move behavior and routinely cross (come within sensor cell resolution of each other), that the filter is consistently able to keep the targets in track.

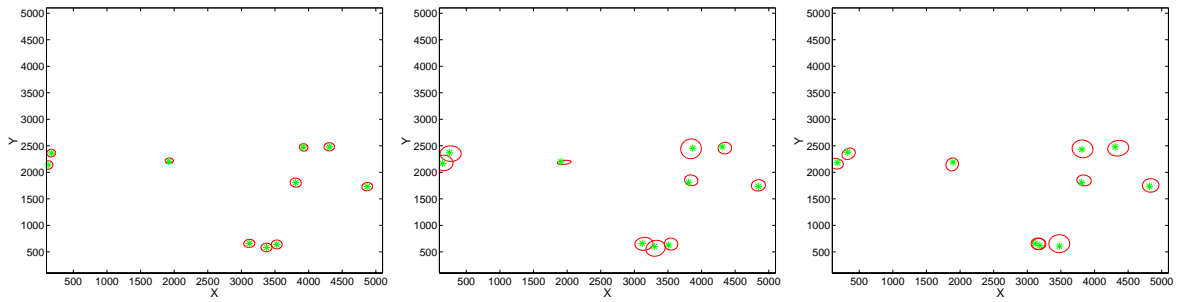


Figure 4. Time Steps 1, 10, and 20. Two Targets Near the Bottom Approach Each Other and then Cross. At Time Step 10, 5 Partitions are Proposed using IP and 5 using CP.

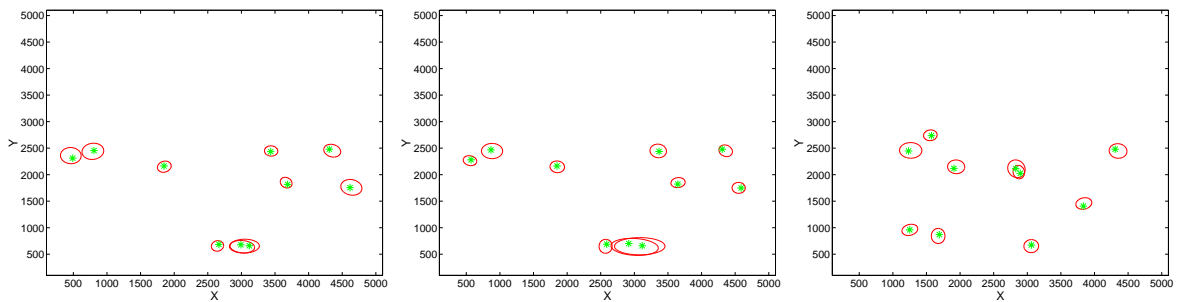


Figure 5. Time Steps 80, 90, 227. A second Target Crossing Occurs in the Same Area as Figure 4. Ambiguity Between the Two Targets Remains for Some Time After the Crossing, but is Resolved in the by Time Step 227.

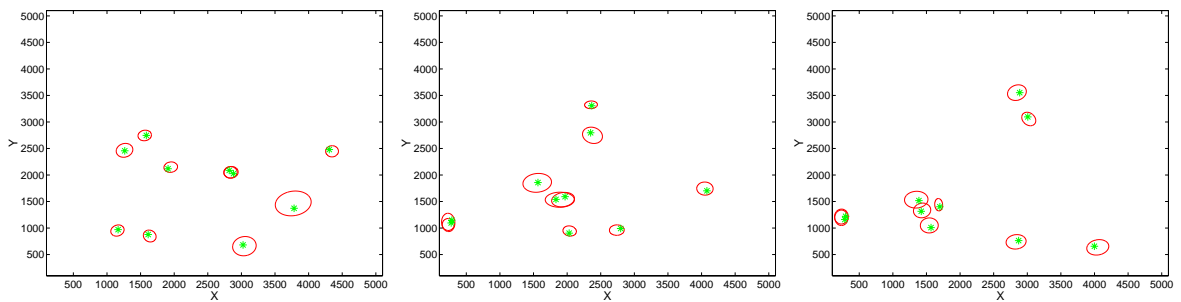


Figure 6. Time Steps 235, 450, 550. Two Targets Near the Center of the Image Come Together and Move in Tandem for Approximately 5 minutes. Meanwhile, the Two Targets Near the Left of the Image Enter a Staging Area.

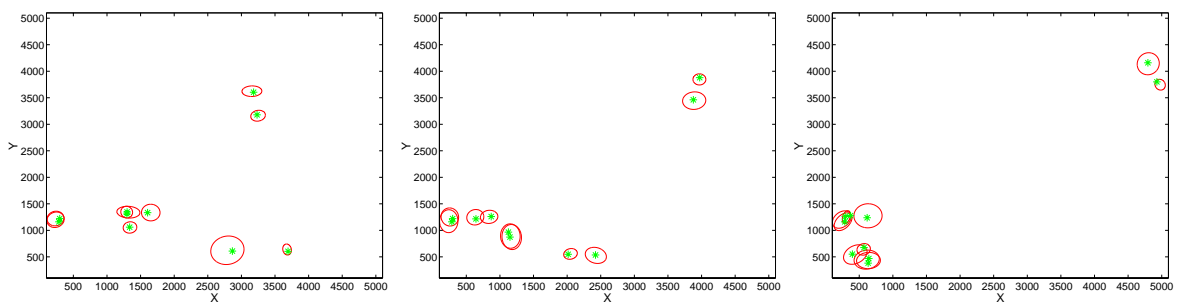


Figure 7. Time Steps 600, 750, and 950. By the End of the Vignette, 8 of the 10 Targets Have Entered a Staging Area, While the Other Two Have Moved to Another Area.

5. DISCUSSION

This paper has presented a new grid-free implementation of a Bayesian method for tracking multiple targets called JMPD. We have developed 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 problems when appropriate. We have demonstrated that this method reduces the computational burden to a reasonable level for realistic problems. In simulations with real target motion, we have shown the ability to track ten targets with complicated kinematic behavior, using thresholded measurements on a grid.

In the sequel to this paper,⁷ we show how this framework can be used to address the sensor management problem. We directly exploit the fact that the JMPD provides a representation that captures all of the information about the states of the targets. This representation can be exploited using information theoretic methods to determine the optimal area of sensor space to direct a measurement.

ACKNOWLEDGMENTS

This work was supported under the United States Air Force contract F33615-02-C-1199, Air Force Research Laboratory 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 author(s) and do not necessarily reflect the views of the United States Air Force.

REFERENCES

1. Arulampalam, M., Maskell, S., Gordon, N. and Clapp, T. "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking", IEEE Transactions on Signal Processing, February 2002.
2. S.S. Blackman, Multiple-Target Tracking with Radar Applications. Norwood, MA. Artech House, 1986.
3. Doucet, A. de Freitas, N., and Gordon, N. "Sequential Monte Carlo Methods in Practice", Springer Publishing, New York, 2001.
4. K. Kastella, "Joint multitarget probabilities for detection and tracking", SPIE Proceedings, Acquisition, Tracking and Pointing XI, 21-25 April, 1997, Orlando, FL.
5. Kastella, K., Pagels, M., and Kreucher, C., "Nonlinear Filtering for Ground Target Applications," SPIE 14th International Aerosense Symposium, Orlando, Florida, April 2000
6. Kreucher, C. and Kastella, K., "Ground Target Tracking Using A Multiple Model Nonlinear Filter", SPIE 15th International Aerosense Symposium, Orlando, Florida, April 2001.
7. Kreucher, C., Kastella, K., and Hero III, Alfred O., "Information-based sensor management for multitarget tracking", SPIE Signal and Data Processing of Small Targets Conference, San Diego, CA, August 2003.
8. Liu, J. and Chen, R. "Sequential Monte Carlo Methods for Dynamic Systems", Journal of the American Statistical Association, September 1998.
9. Matthew Orton and William Fitzgerald, "A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters", IEEE Transactions on Signal Processing, vol. 50, no.2, February 2002, pp. 216-223.