

Multi-target Sensor Management Using Alpha-Divergence Measures^{*}

Chris Kreucher¹, Keith Kastella¹, and Alfred O. Hero²

¹ Veridian's Ann Arbor Research and Development Center, Ann Arbor, MI
{Christopher.Kreucher, Keith.Kastella}@Veridian.com

² The University of Michigan
Department of Electrical Engineering and Computer Science, Ann Arbor, MI
hero@eecs.umich.edu

Abstract. This paper presents a sensor management scheme based on maximizing the expected Rényi Information Divergence at each sample, applied to the problem of tracking multiple targets. The underlying tracking methodology is a multiple target tracking scheme based on recursive estimation of a Joint Multitarget Probability Density (JMPD), which is implemented using particle filtering methods. This Bayesian method for tracking multiple targets allows nonlinear, non-Gaussian target motion and measurement-to-state coupling. Our implementation of JMPD eliminates the need for a regular grid as required for finite element-based schemes, yielding several computational advantages. The sensor management scheme is predicated on maximizing the *expected* Rényi Information Divergence between the current JMPD and the JMPD after a measurement has been made. The Rényi Information Divergence, a generalization of the Kullback-Leibler Distance, provides a way to measure the dissimilarity between two densities. We evaluate the expected information gain for each of the possible measurement decisions, and select the measurement that maximizes the expected information gain for each sample.

1 Introduction

The problem of sensor management is to determine the best way to task a sensor where the sensor may have many modes and may be pointed in many directions. This problem has recently enjoyed a great deal of interest [9]. A typical application, and one that we focus on in our model problem, is to direct an electronically scanned aperture (ESA) radar [2]. An ESA provides great flexibility in pointing and mode selection. For example, the beam can be redirected in a few microseconds, enabling targets to be illuminated at will.

^{*} This material is based upon work supported by the United States Air Force under Contract No. F33615-02-C-1199. 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.

One way of designing a sensor management system is by optimizing information flow. This is analogous to designing a communications system to maximize the channel capacity. Past work in this area has been based on maximizing Kullback-Leibler (KL) divergence. In this work, we use a more general information measure called the Rényi Information Divergence (also known as the α -divergence) [8], which reduces to the KL divergence under a certain limit. The Rényi divergence has additional flexibility in that it allows for emphasis to be placed on specific portions of the information.

We propose here an algorithm for sensor tasking that is motivated by information theory. First, we utilize a target tracking algorithm that recursively estimates the joint multitarget probability density for the set of targets under surveillance. We then strive to task the sensor in such a way that the sensing action it makes results in the maximum amount of information gain. To that end, we employ the Rényi information divergence as a measure of distance between two densities. The decision as to how to use a sensor then becomes one of determining which sensing action will maximize the expected information gain between the current joint multitarget probability density and the joint multitarget probability density after a measurement has been made. This methodology is similar in spirit to that of [10], although our application is quite different. In addition, [11] considers the sensor management as one of maximizing expected information and examines a variety of information driven criteria, including the Kullback-Leibler distance.

The paper is organized as follows. In Section 2, we present the target tracking algorithm that is central to our sensor management scheme. Specifically, we give the details of the 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 implementation provides for computationally tractable implementation, allowing realistic simulations to be made. A sensor management scheme, based on calculating the expected Rényi Information Divergence is given in Section 4. A comparison of the performance of the tracker using sensor management to the tracker using a non-managed scheme on a model problem is given in Section 5. We furthermore include some comments and results as the α parameter in the Rényi Divergence is varied. We conclude with some thoughts on future direction in Section 6.

2 The Joint Multitarget Probability Density (JMPD)

In this section, we provide the details of using the Joint Multitarget Probability Density (JMPD) for target tracking. The concept of JMPD was first discussed by Kastella in [1], where an association free method of tracking multiple targets that moved between discrete cells on a line based on a set of sensor measurements was presented. We generalize that discussion here to deal with targets that have N -dimensional continuous valued state vectors. In the model problem considered herein, we are interested in tracking the position (x, y) and velocity (\dot{x}, \dot{y}) of

multiple targets and so we describe targets 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

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

With the understanding 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

$p(\emptyset | \mathbf{Z})$, the posterior likelihood for no targets in the surveillance volume

$p(\mathbf{x}_1 | \mathbf{Z})$, the posterior likelihood for one target in state \mathbf{x}_1

$p(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{Z})$, the posterior likelihood for two targets in states \mathbf{x}_1 and \mathbf{x}_2

$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3 | \mathbf{Z})$, the posterior likelihood for three targets in states $\mathbf{x}_1, \mathbf{x}_2$ and \mathbf{x}_3

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

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

Bayes rule enables us to update the posterior density as new measurements \mathbf{z}^k arrive as

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

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 in [1] 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 logjam.

3 Particle Filter Implementation of JMPD

We expect that a particle filter based implementation of JMPD will break the computational logjam and allow 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.

Recall from Section 2 that our multitarget state vector \mathbf{X} has T partitions, each corresponding to a target:

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

Furthermore, the joint multitarget probability density $p(\mathbf{X}|\mathbf{Z})$ is defined for $T = 0 \dots \infty$. Each of the particles $\mathbf{X}_p, p = 1 \dots N_{part}$ is a sample drawn from $p(\mathbf{X}|\mathbf{Z})$. 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 each target is modeled using the state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]'$. Then a particular \mathbf{X}_p , which is tracking n_p targets, will be given as

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \dots, \mathbf{x}_{p,n_p}] = \begin{pmatrix} x_{p,1} & x_{p,2} & \dots & x_{p,n_p} \\ \dot{x}_{p,1} & \dot{x}_{p,2} & \dots & \dot{x}_{p,n_p} \\ y_{p,1} & y_{p,2} & \dots & y_{p,n_p} \\ \dot{y}_{p,1} & \dot{y}_{p,2} & \dots & \dot{y}_{p,n_p} \end{pmatrix} \quad (6)$$

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 helps to alleviate the particle starvation issue.

Note there is a permutation symmetry inherent in JMPD, i.e. $p(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{Z}) = p(\mathbf{x}_2, \mathbf{x}_1 | \mathbf{Z})$. This is particularly relevant when targets are near each other and particle partitions begin to swap allegiances. We will have more to say about this issue in 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 to believing there are 0 targets. 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.

3.2 Particle Proposal

Several methods of particle proposal have been investigated. The standard method used, which will be referred 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 using the kinematic prior $p(\mathbf{X}^k | \mathbf{X}^{k-1})$. In the case where the targets are indistinguishable and move independently, each target in \mathbf{X}_p^{k-1} behaves according to the same motion model and is proposed independently of the other targets.

In addition, we have investigated three alternate particle proposal techniques, all of which are developed as a means of biasing the proposal process towards the measurements.

First, the multi-particle proposal method proposes a set of M distinct particles, $\mathbf{X}_p^k(m)$, $m = 1 \dots M$, for each particle at time $k-1$. The proposed particles $\mathbf{X}_p^k(m)$ are then given weights according to the likelihood and a single representative is selected as \mathbf{X}_p^k based on the weights.

Second, the multi-partition proposal method proposes M possible realizations for each partition of a particle $\mathbf{X}_{p,j}^k(m)$. In this notation $\mathbf{X}_{p,j}^k(m)$ refers to the m^{th} proposal for the j^{th} partition of the p^{th} particle at time k . See (6) for a concrete example of a particle and its partitions. The proposed partitions are then given weights according to the likelihood and a new particle \mathbf{X}_p^k is chosen by selecting a representative from each of the proposed partition sets.

Third, the independent-partition method proposed by [7] proposes new partitions and weights each partition independently. Particles at time k , \mathbf{X}_p^k , are formed by selecting partitions from the set of weighted proposed partitions from the particles at time $k - 1$. This method assumes that the targets states are independent, which is not the case when targets cross.

Finally, in any of these methods, target birth and death may be accounted for 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} . For example, with some birth rate probability α we propose a particle with $n_p + 1$ targets at time $k + 1$ starting from a particle with only n_p targets at time k . Similarly, we may propose a particle with fewer targets according to some target death rate. In practice, targets enter and leave only along the boundaries of the surveillance area and so this must be taken into account as well.

3.3 Measurement Update

Each proposed particle is given a weight according to its agreement with the measurements, the kinematic model, and the importance density [4]. Since we are proposing particles based on $p(\mathbf{X}^k|\mathbf{X}^{k-1})$, it can be shown that the proper weighting is given by

$$w_p \propto p(\mathbf{z}|\mathbf{X}_p^k) \quad (7)$$

where the density p comes from the sensor model and incorporates both target detection and false alarm rates.

Recall that 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 (e.g. measurements taken on a XY grid) where the measurement in each cell is independent of the measurements in the other cells, we compute the weight as

$$w_p \propto \prod_i p(z_i|\mathbf{X}_p) \quad (8)$$

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):

$$i_x = i_1, i_2, \dots, i_{n_p} \quad (9)$$

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

$$w_p \propto \prod_{i \notin i_x} p_0(z_i) \prod_{i \in i_x} p(z_i|\mathbf{X}_p) \quad (10)$$

$$w_p \propto \prod_i p_0(z_i) \prod_{i \in i_x} \frac{p(z_i|\mathbf{X}_p)}{p_0(z_i)} \quad (11)$$

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

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

$$w_p \propto \prod_{i \in i_x} \frac{p(z_i | O_{i,p})}{p_0(z_i)} \quad (13)$$

Notice that there is no association of measurement to tracks as is done in conventional multi-target trackers. Each particle \mathbf{X}_p is an estimate of the system state (both number of targets and their states) and has its weight w_p computed based on agreement with the measurements.

3.4 Resampling

As noted by [4], over time the variance of the weights w_p can only increase. If left unchecked, eventually all of the particles except one have near zero weight. To remedy this situation a resampling step is added. When the variance of the weights becomes too high, a new set of N_{part} particles is selected from the existing set with replacement based on the weights w_p . We then have a collection of N_{part} particles with uniform weight that approximate the density $p(\mathbf{X}|\mathbf{Z})$. At this step, particles that do not correspond to measurements are not retained – in particular, particles that have an n_p that is unsupported by measurements are not retained.

The particular resampling that we have implemented is called systematic resampling [4]. We like this scheme because it is easy to implement, runs in $O(N)$, is unbiased, and minimizes Monte Carlo variance. In addition, we favor resampling only when necessary as advocated by [6]. This saves time as well as reduces the variance of the estimate. Many other resampling schemes and modifications are presented in the literature [5]. Of these methods, we have found that adaptive resampling [6] and Markov Chain Monte Carlo (MCMC) moves using a Metropolis-Hasting scheme [5] lead to improved performance over straightforward resampling 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

$$I_p = \begin{cases} 1 & \text{if } n_p = n \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

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

$$p(n|\mathbf{Z}) = \sum_{p=1}^{N_{part}} I_p w_p \tag{15}$$

So to estimate the probability of n targets in the surveillance volume, we sum up the weights of the particles that have n partitions.

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 :

$$\tilde{I}_p = \begin{cases} 1 & \text{if } n_p \geq n \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Furthermore, we define the normalized weights to be

$$\hat{w}_p = \frac{w_p \tilde{I}_p}{\sum_{l=1}^{N_{part}} \tilde{I}_l w_l} \tag{17}$$

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

$$\hat{\mathbf{X}}(i) = E[\mathbf{X}(i)] = \sum_{p=1}^{N_{part}} \tilde{I}_p \hat{w}_p \mathbf{X}_{p,i} \tag{18}$$

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

$$\hat{\mathbf{\Lambda}}(i) = \sum_{p=1}^{N_{part}} \tilde{I}_p \hat{w}_p (\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i)) (\mathbf{X}_{p,i} - \hat{\mathbf{X}}(i))' \tag{19}$$

The indicator function \tilde{I}_p causes the summations in (18) and (19) to be taken over only those particles that are tracking target i . The permutation symmetry issue mentioned in Section 3 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 (18) or (19) can be made, we must ensure that partition i , ($i = 1...T$), corresponds to the same target in each particle. In our work, this is accomplished by taking care to retain the ordering during the particle proposal process.

4 Rényi Information Divergence for Sensor Management

Our paradigm for tasking the sensor is to choose the sensing action (i.e. sensing modality or sensor pointing direction) that maximizes the expected information gain. To that end, we introduce the Rényi information divergence (20), also known as the α -divergence, between two densities f_1 and f_0 :

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha - 1} \ln \int f_1^\alpha(x) f_0^{1-\alpha}(x) dx \tag{20}$$

The function D_α gives a measure of the distance between the two densities f_0 and f_1 . In our application, we are interested in computing the divergence between the predicted density $p(\mathbf{X}|\mathbf{Z}^{k-1})$ and the updated density after a measurement is made, $p(\mathbf{X}|\mathbf{Z}^k)$.

$$D_\alpha(p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) = \frac{1}{\alpha - 1} \ln \sum_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}^k)^\alpha p(\mathbf{X}|\mathbf{Z}^{k-1})^{1-\alpha} \quad (21)$$

The integral in (20) reduces to a summation since any discrete approximation of $p(\mathbf{X}|\mathbf{Z}^{k-1})$ only has nonzero probability at a finite number of target states. After some algebra and the incorporation of Bayes rule (3), one finds that this quantity can be simplified to

$$D_\alpha(p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) = \frac{1}{\alpha - 1} \ln \frac{1}{p(\mathbf{z}|\mathbf{Z}^{k-1})^\alpha} \sum_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}^{k-1}) p(\mathbf{z}|\mathbf{X})^\alpha \quad (22)$$

Our particle filter approximation of the density reduces (22) to

$$D_\alpha(p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) = \frac{1}{\alpha - 1} \ln \frac{1}{p(\mathbf{z})^\alpha} \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X}_p)^\alpha \quad (23)$$

where

$$p(\mathbf{z}) = \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X}_p) \quad (24)$$

We would like to make the divergence between the current density and the density after a new measurement has been made as large as possible. This indicates that the sensing action has maximally increased the information content of the measurement updated density, $p(\mathbf{X}|\mathbf{Z}^k)$, with respect to the density before a measurement was made, $p(\mathbf{X}|\mathbf{Z}^{k-1})$.

Of course, we cannot calculate (23) exactly until after the measurement at time k has been made. However, we can calculate the expected value of this quantity for different sensing actions. We propose as a method of sensor management, then, calculating the expected value of (23) for each of the m , ($m = 1 \dots M$) possible sensing actions and choosing the action that maximizes the expectation. In this notation m refers to any possible sensing action under consideration, including but not limited to sensor mode and sensor beam positioning. In this manner, we say that we are making the measurement that maximizes expected information gain. Notice that this is a greedy scheme, which chooses to make the measurement that optimizes information gain for the next time step.

The expected value of (23) may be written as an integral over all possible outcomes z_m when performing sensing action m :

$$\langle D_\alpha \rangle_m = \int dz_m p(z_m|\mathbf{Z}^{k-1}) D_\alpha(p(\mathbf{X}|\mathbf{Z}^k)||p(\mathbf{X}|\mathbf{Z}^{k-1})) \quad (25)$$

In the special case where measurements are thresholded and are therefore either detections or no-detections, this integral reduces to

$$\langle D_\alpha \rangle_m = p(z = 0 | \mathbf{Z}^{k-1}) D_\alpha |_{m, z=0} + p(z = 1 | \mathbf{Z}^{k-1}) D_\alpha |_{m, z=1} \quad (26)$$

Which, using (23) results in

$$\langle D_\alpha \rangle_m = \frac{1}{\alpha - 1} \sum_{z=0}^1 p(z) \ln \frac{1}{p(z)^\alpha} \sum_{p=1}^{N_{part}} w_p p(z | \mathbf{X}_p)^\alpha \quad (27)$$

Implementationally, the value of equation (27) can be calculated for a host of possible actions using only a single loop through the particles.

In summary, our sensor management algorithm is a recursive algorithm that proceeds as follows. At each occasion where a sensing action is to be made, we evaluate the expected information gain as given by (27) for each possible sensing action m . We then select and make the sensing action that gives maximal expected information gain.

We note here that the α parameter may be used to adjust how heavily one emphasizes the tails of the two distributions. In the limiting case of $\alpha \rightarrow 1$ the Rényi divergence becomes the more commonly known Kullback-Leibler (KL) discrimination (28).

$$\lim_{\alpha \rightarrow 1} D_\alpha(f_1 || f_0) = \int f_0(x) \ln \frac{f_0(x)}{f_1(x)} dx \quad (28)$$

5 Simulation Results

We test the performance of the sensor management scheme by considering the following model problem. We have three targets moving on a 12×12 sensor grid. Each target is modeled using the four-dimensional state vector $[x, \dot{x}, y, \dot{y}]'$. Target motion is simulated using a constant-velocity (CV) model with a (relatively) large diffusive component. The trajectories have been shifted and time delayed so that there are two times during the simulation where targets cross paths.

The target kinematics assumed by the filter (2) are CV as in the simulation. At each time step, a set of L (not necessarily distinct) cells are measured. 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 (3). In this illustration, we take $P_d = 0.5$, and $P_f = P_d^{(1+SNR)}$, which is a standard model for thresholded detection of Rayleigh returns.

We contrast in this section the performance of the tracker when the sensor uses a non-managed (periodic) scheme versus the performance when the sensor uses the management scheme presented in Section 4. The periodic scheme measures each cell in sequence. At time 1, cells $1 \dots L$ are measured. At time 2, cells

$L + 1 \dots 2L$ are measured. This sequence continues until all cells have been measured, at which time the scheme resets. The managed scheme uses the expected information divergence to calculate the best L cells to measure at each time.

For the simulations that follow, we have taken α in (27) near 1. However, we still use the Rényi formulation of (20) rather than the KL formulation of (28) because the Rényi formulation provides some computational advantages.

In Fig. 1, we give a single-time snapshot, which graphically illustrates the difference between the two schemes. On the left, we show the managed scheme and on the right the periodic scheme. In both panes, the three targets are marked with an asterisk, the covariance ellipses of the estimated target position are shown, and we use grayscale to indicate the number of times each cell has been measured at this time step.

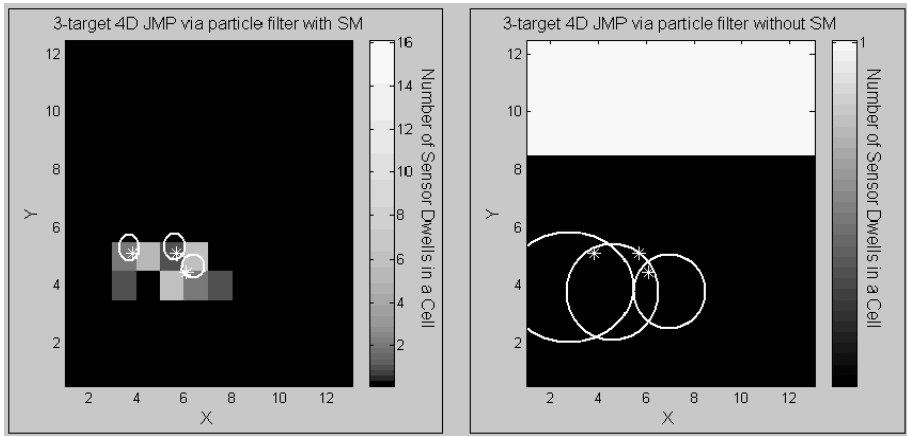


Fig. 1. A Comparison of Non-Managed and Managed Tracking. (L) Using Sensor Management, and (R) Using a Periodic Scheme. With Sensor Management, Dwells are Only Used in Areas that Contain Targets and the Covariance Ellipses are Much Tighter.

Qualitatively, in the managed scenario the measurements are focused in or near the cells that the targets are in. Furthermore, the covariance ellipses, which reflect the current state of knowledge of the tracker conditioned on all previous measurements, are much tighter. In fact, the non-managed scenario has confusion about which tracks correspond to which target as the covariance ellipses overlap.

A more detailed is provided in the Monte Carlo simulation results of Figure 2. The sensor management algorithm was run with $L = 24$ (i.e. was able to scan 24 cells at each time step) and is compared to the non-managed scheme with 24 to 312 looks. The unmanaged scenario needs approximately 312 looks to equal the performance of the managed algorithm in terms of RMSE error. We say that the sensor manager is approximately 13 times as efficient as allocating the sensors without management.

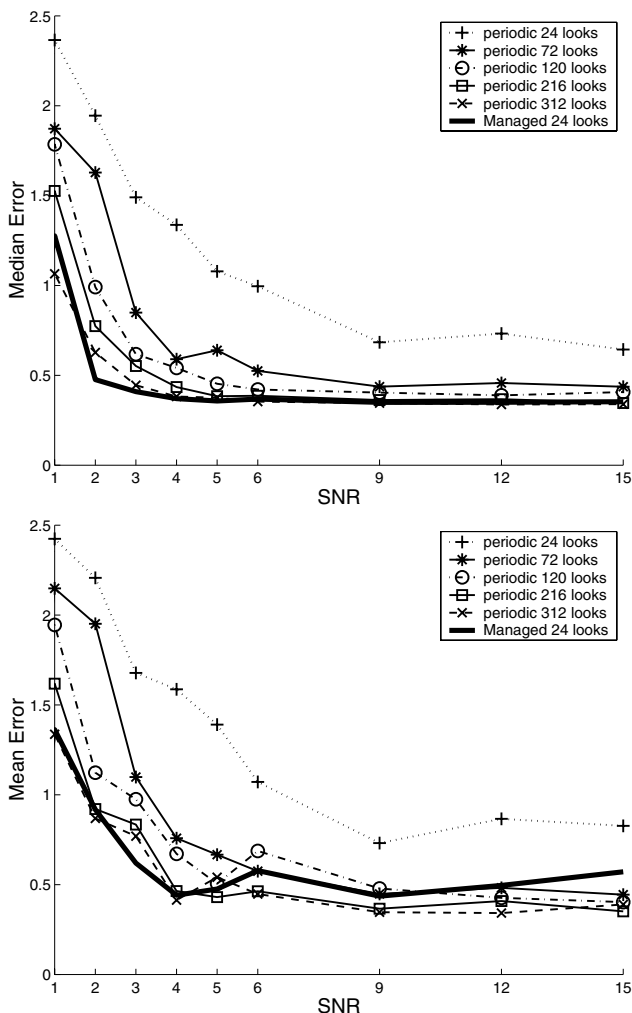


Fig. 2. Median and Mean Error vs. Signal To Noise Ratio (SNR). Managed Performance With 24 Looks is Similar to Unmanaged Performance With 312 Looks.

In addition, we have investigated the performance of the sensor management algorithm with different values of α in (27) under the same set of simulation parameters. As shown in 3, it appears that in the case under consideration that the technique is rather insensitive to the choice of α . We anticipate that values of α that deviate from unity may be useful in the case of model mismatch.

6 Conclusion

In this paper, we have proposed an algorithm for multiple target tracking and sensor management. The central element of interest in both the target track-

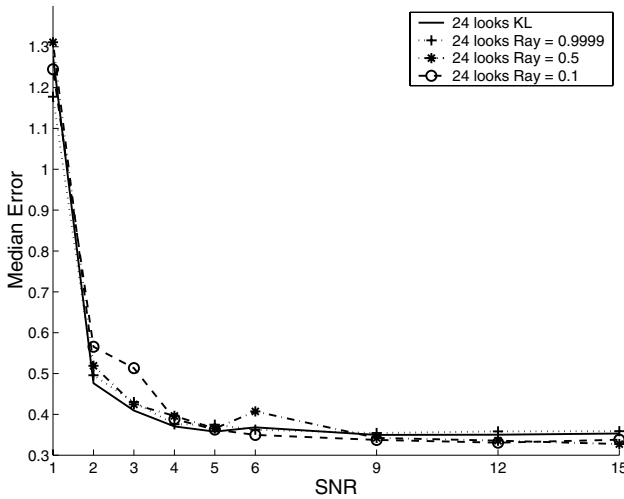


Fig. 3. Performance of the Sensor Management Technique for Different α .

ing and sensor management schemes is the posterior density $p(\mathbf{X}|\mathbf{Z})$, which is approximated using particle filtering methods.

The particle filter implementation we have proposed has three main benefits. First, by nature of being a particle filter implementation, it allows for non-linear measurement to state coupling, non-linear state evolution (target motion), and non-Gaussian densities. Second, the formulation explicitly enforces the multi-target nature of the problem. Each particle simultaneously postulates both the number of targets and their states, both of which are unknown. Finally, the particle filter implementation breaks the computational logjam that grid-based techniques have presented in the past. This makes this technique applicable to larger scale problems.

The information-based sensor management scheme presented in this paper is based on computing the expected information gain for each sensor tasking under consideration. The sensor management algorithm is integrated with the target tracking algorithm in that it too uses the posterior density $p(\mathbf{X}|\mathbf{Z})$. In this case, the posterior is used in conjunction with target kinematic models and sensor models to predict which measurements will provide the most information gain. In simulated scenarios, we find that the tracker with sensor management gives similar performance to the tracker without sensor management while using a factor of 13 fewer sensor dwells.

There are two interesting directions in which we see this work evolving. First, this method is amenable to incorporating auxiliary information such as ground elevation maps and sensor trajectories. For example, if the appropriate auxiliary information were incorporated, this method would clearly never choose to make a measurement in a region that was not visible to the sensor due to hill regions between the sensor and the desired look location. Second, the current algorithm

is a greedy algorithm, choosing to make the measurement that is best at the current time step. It would be beneficial to extend the methodology to plan several time instances in the future.

References

1. K. Kastella, "Joint multitarget probabilities for detection and tracking", SPIE Proceedings, Acquisition, Tracking and Pointing XI, 21-25 April, 1997, Orlando, FL.
2. S.S. Blackman, Multiple-Target Tracking with Radar Applications. Norwood, MA. Artech House, 1986.
3. Alfred O. Hero III, Bing Ma, Olivier J.J. Michel, and John Gorman, "Applications of Entropic Spanning Graphs", IEEE Signal Processing Magazine, September 2002, pp. 85-95.
4. 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.
5. Doucet, A. de Freitas, N., and Gordon, N. "Sequential Monte Carlo Methods in Practice", Springer Publishing, New York, 2001.
6. Liu, J. and Chen, R. "Sequential Monte Carlo Methods for Dynamic Systems", Journal of the American Statistical Association, September 1998.
7. 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.
8. A. Rényi, "On measures of entropy and information", Proc. 4th Berkeley Symp. Math. Stat. and Prob., volume 1, pp. 547-561, 1961.
9. D. Sinno and D. Kreithen, "A Constrained Joint Optimization Approach to Dynamic Sensor Configuration", Thirty Six Asilomar Conference on Signals, Systems, and Computers, November 2002.
10. D. Geman and B. Jedynak, "An active testing model for tracking roads from satellite images", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 1, January 1996, pp. 1-14.
11. F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration", IEEE Signal Processing Magazine, March 2002, pp. 61-72.