Information Based Sensor Management for Multitarget Tracking

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

[a]Veridian's Ann Arbor Research and Development Center, Ann Arbor, MI, USA
[b]The University of Michigan Department of Electrical Engineering and Computer Science, Ann Arbor, MI, USA

ABSTRACT

We present in this paper an information based method for sensor management that is based on tasking a sensor to make the measurement that maximizes the *expected* gain in information. The method is 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. 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 use the Rényi Information Divergence to 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.

Keywords: Sensor Management, Particle Filtering, Multi-target Tracking

1. INTRODUCTION

The problem of sensor management (SM) 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.[8] A typical application, and one that we focus on in our model problems, is to direct an electronically scanned aperture (ESA) radar.[1] 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.

We propose here a sensor tasking algorithm that is motivated by information theory. Our technique strives to optimize information flow, which 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 utilize a more general information measure called the Rényi Information Divergence[7] (also known as the α-divergence), which reduces to the KL divergence under a certain limit. The Rényi divergence has additional flexibility in that in allows for emphasis to be placed on specific portions of the information.

We apply our sensor management scheme to the problem of tracking a collection of targets moving in a surveillance region. First, we utilize a target tracking algorithm to recursively estimate 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 Geman[2] although our application is quite different. In addition, Zhao[9] considers the sensor management as one of maximizing expected information and examines a variety of information driven criteria, including the Kullback-Leibler distance.

Please Direct correspondence to Christopher.Kreucher@veridian.com

Signal and Data Processing of Small Targets 2003,
edited by Oliver E. Drummond, Proceedings of SPIE Vol. 5204
(SPIE, Bellingham, WA, 2003) · 0277-786X/03/$15 · doi: 10.1117/12.502699

The paper is organized as follows. In Section 2, we review 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 the joint multitarget probability density (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 realistic scenarios to be considered. The particle filter implementation of JMPD is the subject of the companion paper[6] and thus only those details necessary for the sensor management application are given here. Our sensor management scheme, which is based on calculating the expected Rényi Information Divergence, is extensively detailed in Section 4. We furthermore include simulations and comments with respect to the choice of the α parameter in the Rényi Divergence. A comparison of the performance of the tracker using sensor management to the tracker using a non-managed scheme on two model problems of increasing realism is given in Section 5. We conclude with some thoughts on future direction in Section 6.

2. JOINT MULTITARGET PROBABILITY DENSITIES (JMPD)

As described thoroughly in the companion paper,[6] the joint multitarget probability density (JMPD) provides a means for tracking an unknown number of targets in a Bayesian setting. In short, the joint multitarget conditional probability density $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|\mathbf{Z}^k)$ is the probability density that there are exactly T targets with states $\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ in the surveillance region 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. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, ...\mathbf{z}^k\}$, 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 $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|\mathbf{Z}^k)$ 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 $\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ as a partition of the state $\mathbf{X}$. For convenience, the density will be written more compactly in the traditional manner as

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

With the understanding that the state-vector $\mathbf{X}$ represents a variable number of targets each possessing their own state vector.

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

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

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

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

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 by Kastella[5] 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. THE PARTICLE FILTER IMPLEMENTATION OF JMPD

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, $\mathbf{X}_p, (p = 1...N_{part})$:

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

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:

$$\mathbf{X} = [\mathbf{x}_1, \ \mathbf{x}_2, \ ..., \ \mathbf{x}_{T-1}, \ \mathbf{x}_T] \tag{5}$$

Furthermore, the joint multitarget probability $p(\mathbf{X}|\mathbf{Z})$ is defined for $T = 0...\infty$. Each of the particles $\mathbf{X}_p$, $p = 1...N_{part}$ is a sample drawn from $p(\mathbf{X}|\mathbf{Z})$. Therefore, a particle $\mathbf{X}_p$ may have any number of partitions from 0 to ∞, each partition corresponding to a different target. In practice, of course, the maximum number of targets under surveillance is truncated at some finite number T. 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.

It is important to emphasize here that the set of particles $\mathbf{X}_p$, $p = 1...N_{part}$ constitutes an approximation to the density $p(\mathbf{X}|\mathbf{Z})$. Since $p(\mathbf{X}|\mathbf{Z})$ is defined for all possible number of targets, a particle X_p may have 0, 1, ... T partitions and the number of partitions may vary from particle to particle.

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

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

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 technique helps to alleviate the particle starvation issue, ensuring that all targets are represented by the particles.

4. RÉNYI INFORMATION DIVERGENCE FOR SENSOR MANAGEMENT

Our information based method for tasking the sensor is to choose the sensing action that maximizes the expected information gain. To that end, our algorithm proceeds by first enumerating all possible sensing actions. A sensing action may consist of choosing a particular mode (i.e. SAR mode versus GMTI mode), a particular dwell point, or a combination of the two. We next calculate the *expected* information gain in making each of the possible sensing actions, and select to take the action that yields the maximum expected information gain.

The calculation of information gain between two densities f_1 and f_0 is done using the Rényi information divergence (7), also known as the α-divergence:

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha - 1} ln \int f_1^\alpha(x) f_0^{1-\alpha}(x) dx \tag{7}$$

The particle filter based multitarget tracker yields a particularly convenient form for calculation of equation (7). This is detailed in Section 4.1.

The α parameter in equation (7) may be used to adjust how heavily one emphasizes the tails of the two distributions f_1 and f_0. In the limiting case of $\alpha \to 1$ the Rényi divergence becomes the more commonly utilized Kullback-Leibler (KL) discrimination (8).

$$\lim_{\alpha \to 1} D_\alpha(f_1||f_0) = \int f_0(x) ln \frac{f_0(x)}{f_1(x)} dx \tag{8}$$

In the case that for $\alpha = 0.5$, the Rényi information divergence becomes the log Hellinger distance squared, where the Hellinger distance is defined by

$$d_H(f_1, f_0) = \frac{1}{2} \int \left(\sqrt{f_1(x)} - \sqrt{f_0(x)} \right)^2 dx \tag{9}$$

We find that the flexibility introduced by the α parameter allows the algorithm to be tuned to perform well in challenging situations, such as filter-target model mismatch. We detail our analysis of the choice of α in Section (4.2).

4.1. Derivation of Expected Rényi Information Divergence for the PF Implementation of JMPD

The function D_α given in equation (7) is a measure of the divergence 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, $p(\mathbf{X}|\mathbf{Z}^k)$. Therefore, we write

$$D_\alpha \left(p(\mathbf{X}|\mathbf{Z}^k) || p(\mathbf{X}|\mathbf{Z}^{k-1}) \right) = \frac{1}{\alpha - 1} ln \sum_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}^k)^\alpha p(\mathbf{X}|\mathbf{Z}^{k-1})^{1-\alpha} \tag{10}$$

The integral in equation (7) 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 (eq. 3), one finds that this quantity can be simplified to

$$D_\alpha \left(p(\mathbf{X}|\mathbf{Z}^k) || p(\mathbf{X}|\mathbf{Z}^{k-1}) \right) = \frac{1}{\alpha - 1} ln \frac{1}{p(\mathbf{z}|\mathbf{Z}^{k-1})^\alpha} \sum_{\mathbf{X}} p(\mathbf{X}|\mathbf{Z}^{k-1}) p(\mathbf{z}|\mathbf{X})^\alpha \tag{11}$$

Our particle filter approximation of the density (eq. 4) reduces equation (11) to

$$D_\alpha \left(p(\mathbf{X}|\mathbf{Z}^k) || p(\mathbf{X}|\mathbf{Z}^{k-1}) \right) = \frac{1}{\alpha - 1} ln \frac{1}{p(\mathbf{z})^\alpha} \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X_p})^\alpha \tag{12}$$

where

$$p(\mathbf{z}) = \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X}_p) \tag{13}$$

We note in passing here that the sensor model $p(\mathbf{z}|\mathbf{X}_p)$ is used to incorporate everything known about the sensor, including SNR, detection probabilities, and even whether the locations represented by $\mathbf{X}_p$ are visible to the sensor.

We would like to choose to perform the measurement that makes 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, $p(\mathbf{X}|\mathbf{Z}^{k-1})$.

We propose, then, as a method of sensor management calculating the expected value of equation (12) for each of the $m(m = 1...M)$ 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 selection and sensor beam positioning. In this manner, we say that we are making the measurement that maximizes expected information gain.

The expected value of equation (12) may be written as an integral over all possible outcomes z_m when performing sensing action m:

$$< D_\alpha >_m = \int dz_m p(z_m|\mathbf{Z}^{k-1}) D_\alpha \left(p(\mathbf{X}|\mathbf{Z}^k) || p(\mathbf{X}|\mathbf{Z}^{k-1}) \right) \tag{14}$$

In the special case where measurements are thresholded and are therefore either detections or no-detections (i.e. $z = 0$ or $z = 1$), this integral reduces to

$$< D_\alpha >_m = p(z = 0|\mathbf{Z}^{k-1}) D_\alpha|_{m,z=0} + p(z = 1|\mathbf{Z}^{k-1}) D_\alpha|_{m,z=1} \tag{15}$$

Which, using equation (12) results in

$$< D_\alpha >_m = \frac{1}{\alpha - 1} \sum_{z=0}^{1} p(z) ln \frac{1}{p(z)^\alpha} \sum_{p=1}^{N_{part}} w_p p(z|\mathbf{X}_p)^\alpha \tag{16}$$

Implementationally, the value of equation (16) can be calculated for a host of possible actions using only a single loop through the particles. This results in a computationally efficient method for making sensor tasking decisions.

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 equation (16) for each possible sensing action m. We then select and make the sensing action that gives maximal expected information gain. Notice that this is a greedy scheme, which chooses to make the measurement that optimizes information gain only for the next time step.

4.2. On the Value of α in the Rényi Divergence

The Rényi divergence has been used in the past in many diverse applications, including content-based image retreival, georegistration of imagery, and target detection[3,4] . These studies have provided some guidance as to the optimal choice of the parameters α.

In the geogregistration problem[3] it was empirically determined that the value of α leading to highest resolution clusters around either $\alpha = 1$ or $\alpha = 0.5$ corresponding to the KL divergence (eq. 8) and the Hellinger affinity (eq. 9) respectively. The determining factor appears to be the degree of differentiation between the two densities under consideration. If the densities are very similar, i.e. difficult to discriminate, then the indexing performance of the Hellinger affinity distance ($\alpha = 0.5$) was observed to be better that the KL divergence ($\alpha = 1$). In fact, an asymptotic analysis[4] has shown that $\alpha = .5$ results in the maximum distance between two densities that are very similar. We say, then, that this value of α stresses the tails, i.e. the minor differences, between two densities.

Therefore, we have reason to believe that either $\alpha = 0.5$ or $\alpha = 1$ are good choices. We investigate the performance of our sensor management scheme under both choices in Section (5).

5. SIMULATION RESULTS

In this section, we provide some simulation results that show the benefit of sensor management in the multitarget tracking scenario.

5.1. An Extensive Evaluation of SM Performance Using Three Simulated Targets

We first test the performance of the sensor management scheme by considering the following model problem. There are three targets moving on a $12x12$ 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 (i.e. come within sensor resolution of each other), to make the problem challenging.

The target kinematics assumed by the filter (equation 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. The filter is initialized with 10% of the particles in the correct state (both number of targets and kinematic state). The rest of the particles are uniformly distributed in both the number of targets and kinematic state.

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 $1...L$ are measured. At time 2, cells $L+1...2L$ 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. This often results in the same cell being measured several times at one time step.

In Fig. 1, we present a single-time snapshot from the tracker, which graphically illustrates the difference between the two schemes.

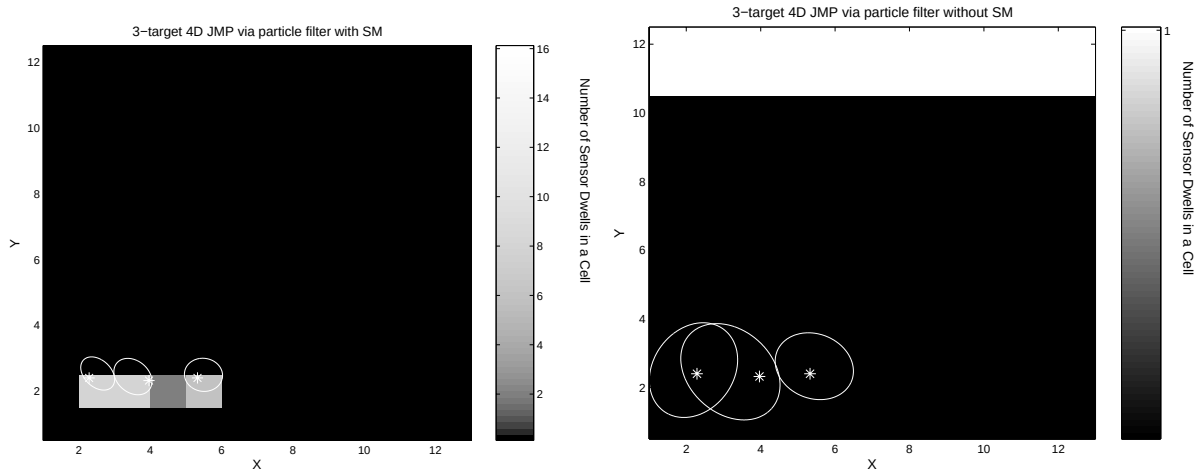

Figure 1. A Comparison of Non-Managed and Managed Tracking. (L) Using Sensor Management, and (R) Using a Periodic Scheme. With Sensor Management, Dwells are Only Used in Areas that Contain Targets and the Covariance Ellipses are Much Tighter.

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 gray scale to indicate the number of times each cell has been measured at this time step. 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 examination 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. Here we take $\alpha \approx 1$ (KL) in equation (8). 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. This efficiency implies that in an operational scenario target tracking could be done with an order of magnitude fewer sensor dwells. Alternatively put, more targets could be tracked with the same number of total resources when this sensor management strategy is employed.

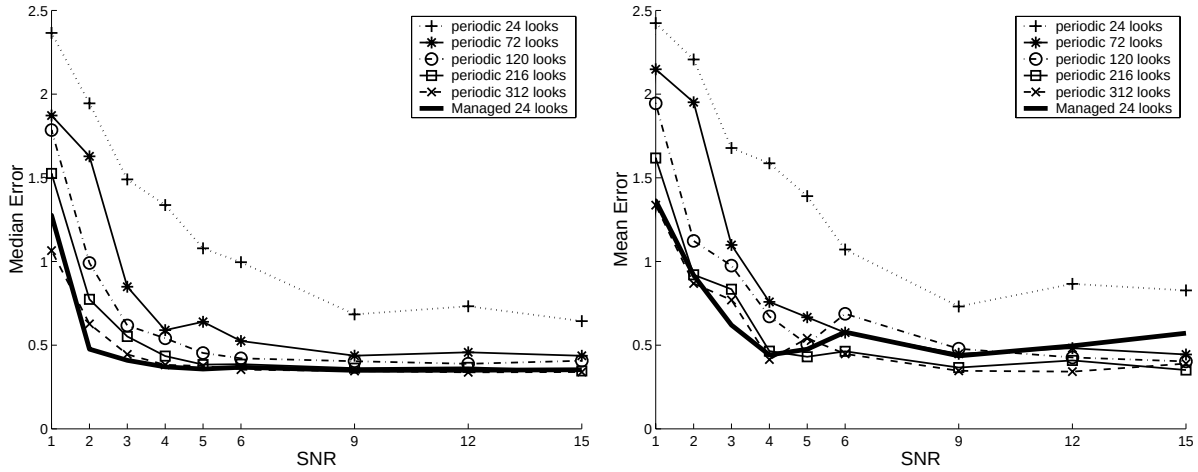

Figure 2. Median and Mean Error vs. Signal To Noise Ratio (SNR). Managed Performance With 24 Looks is Similar to Unmanaged Performance With 312 Looks.

To determine the sensitivity of the sensor management algorithm to the choice of α, we test the performance with $\alpha = .1$, $\alpha = .5$, and $\alpha \approx 1$.

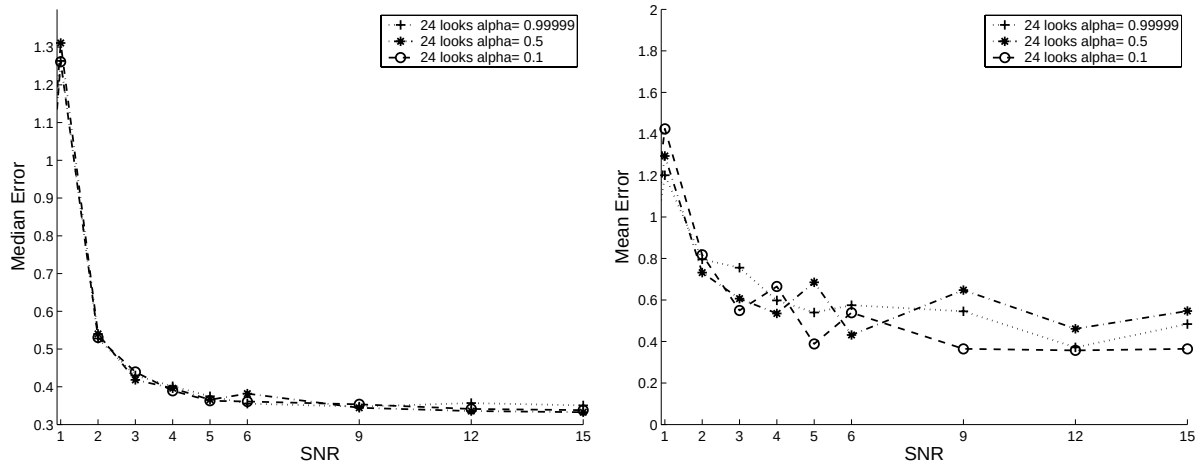

Figure 3. The Performance of the Sensor Management Algorithm with Different Values of α. We Find that in the Case Where the Filter Dynamics Match the Actual Target Dynamics, the Algorithm is Insensitive to the Choice of α.

Figure 3 shows that in this case, where the actual target motion is very well modeled by the filter dynamics, that the performance of the sensor management algorithm is insensitive to the choice of α. We generally find this to be the case when the filter model is closely matched to the actual target kinematics.

5.2. A Comparison Using Ten Real Targets

We test the sensor management algorithm again using a modified version of the above simulation, which is intended to demonstrate the technique in a scenario of increased realism. Here we have ten targets moving in a $5000m$ x $5000m$ surveillance area. Each target is modeled 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 with in sensor cell resolution (i.e. cross). Therefore, there is often a measurement to track ambiguity, which is handled automatically by JMPD because there is no measurement to track assignment that must be done. Target positions are recorded at 1 second intervals, and the simulation duration is 1000 time steps.

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. This model mismatch adds another level of difficulty to this scenario that was not present in the previous case. We use 500 particles, each of is tracking the states of all ten targets, and therefore each particle has 40 dimensions.

At each time step, an imager is able to measure cells in the surveillance area by making measurements on a grid with 100m x 100m detection cell resolution. The sensor simulates a moving target indicator (MTI) system in that it may lay a beam down on the ground that is one resolution cell wide and many resolution cells deep. Each time a beam is formed, a set of measurements is returned, corresponding to the depth of the beam. We refer to each beam that is laid down as a "Look". We judge the performance of a tracker in terms of the number of looks needed to perform the task (e.g. keep targets in track, or track with a certain mean squared error).

The sensor is at a fixed location above the targets and all cells are always visible to the sensor. When making a measurement, 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.

We compare the performance of the managed and unmanaged scenarios in Figure 4. Our method of comparison here is to determine empirically the number of Looks needed in the unmanaged scenario to achieve the same performance as the managed algorithm with $L = 50$ looks.

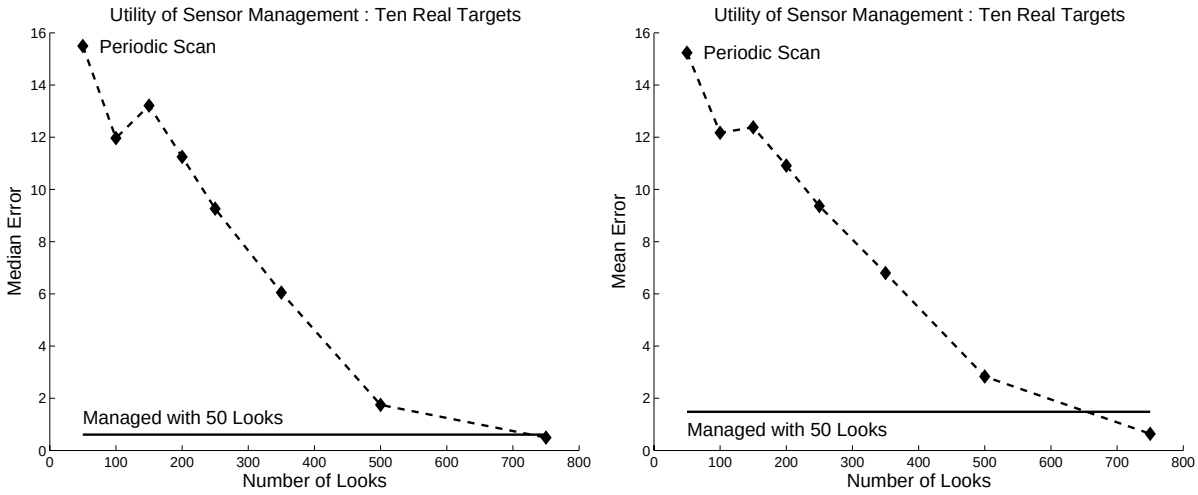

Figure 4. Median and Mean Error vs. Number of Looks. Managed Performance With 50 Looks Performs Similarly to Unmanaged with $600 - 700$ Looks.

Figure 4 shows that the unmanaged scenario needs approximately 600 to 700 looks to equal the performance of the managed algorithm in terms of RMSE error. We say that the sensor manager is approximately 12 to 14 times as efficient as allocating the sensors without management.

We compare next the performance of the sensor management algorithm under different values of α in equation (7). This problem is more challenging then the simulation of Section 5 for several reasons. Of particular interest is the fact that the filter motion model and actual target kinematics do not match very well. The asymptotic analysis performed previously leads us to believe that $\alpha = 0.5$ is the right choice in this scenario.

In Figure 5, we show the results of 50 Monte Carlo trials using our sensor management technique with $\alpha = 0.1$, $\alpha = 0.5$, and $\alpha = 0.99999$. The statistics are summarized in Table 1. We find that indeed the sensor management algorithm with $\alpha = 0.5$ performs best here as it does not loose track on any of the 10 targets during any of the 50 simulation runs. Both the $\alpha = 1$ and $\alpha = 0.1$ case lose track of targets on several occasions.

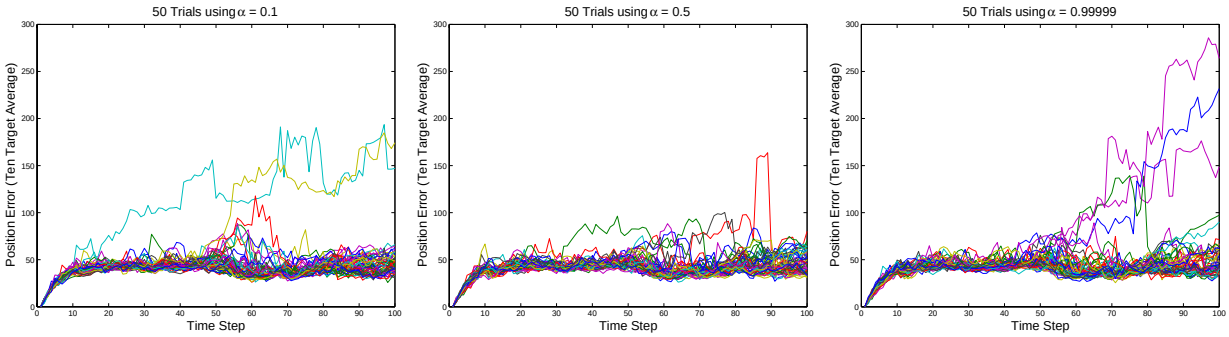

Figure 5. A Comparison of SM Performance Under Different Values of the Rényi Divergence Parameter, α.

Table 1. Sensor Management Performance With Different Values of α.

α	Mean Position Error(m)	Position Error Variance (m)
0.1	49.57	614.01
0.5	47.28	140.25
0.99999	57.44	1955.54

6. DISCUSSION

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 uses the posterior density $p(\mathbf{X}|\mathbf{Z})$ approximated by the multitarget tracker. 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 with more than a ten-fold improvement in sensor efficiency.

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.

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. S.S. Blackman, Mulitple-Target Tracking with Radar Applications. Norwood, MA. Artech House, 1986.
2. 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.
3. A. O. Hero, B. Ma, O. Michel and J. Gorman, "Applications of entropic spanning graphs," IEEE Signal Processing Magazine (Special Issue on Mathematics in Imaging), Vol 19, No. 5, pp 85-95, Sept. 2002.
4. A. O. Hero, B. Ma, O. Michel, and J. D. Gorman, "Alpha divergence for classification, indexing and retrieval, Technical Report 328, Comm. and Sig. Proc. Lab. (CSPL), Dept. EECS, University of Michigan, Ann Arbor, May, 2001.
5. K. Kastella, "Joint multitarget probabilities for detection and tracking", SPIE Proceedings, Acquisition, Tracking and Pointing XI, 21-25 April, 1997, Orlando, FL.
6. Kreucher, C., Kastella, K., and Hero III, Alfred O., "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density", SPIE Signal and Data Processing of Small Targets Conference, San Diego, CA, August 2003.
7. A. Rényi, "On measures of entropy and information", Proc. 4th Berkeley Symp. Math. Stat. and Prob., volume 1, pp. 547-561, 1961.
8. 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.
9. F. Zhao, J. Shin, and J. Reich, "Information-Driven Dynamic Sensor Collaboration", IEEE Signal Processing Magazine, March 2002, pp. 61-72.