Sensor Management Using Relevance Feedback Learning

Chris Kreucher*, Keith Kastella and Alfred O. Hero III, *IEEE Fellow*

***Abstract*— An approach that is common in the machine learning literature, known as relevance feedback learning, is applied to provide a method for managing agile sensors. In the context of a machine learning application such as image retrieval, relevance feedback proceeds as follows. The user has a goal image in mind that is to be retrieved from a database of images (i.e., learned by the system). The system computes an image or set of images to display (the query). Oftentimes, the decision as to which images to display is done using divergence metrics such as the Kullback-Leibler (KL) divergence. The user then indicates the relevance of each image to his goal image and the system updates its estimates (typically a probability mass function on the database of images). The procedure repeats until the desired image is found. Our method for managing agile sensors proceeds in an analogous manner. The goal of the system is to learn the number and states of a group of moving targets occupying a surveillance region. The system computes a sensing action to take (the query), based on a divergence measure called the Rényi divergence. A measurement is made, providing relevance feedback and the system updates its probability density on the number and states of the targets. This procedure repeats at each time where a sensor is available for use. It is shown using simulated measurements on real recorded target trajectories that this method of sensor management yields a ten fold gain in sensor efficiency when compared to periodic scanning.**

EDICS Category: 2-INFO

***Index Terms*— Sensor Management, Machine Learning, Relevance Feedback, Multitarget Tracking, Particle Filtering, Joint Multitarget Probability Density.**

I. Introduction

THE problem of sensor management is to determine the best way to task a sensor or group of sensors when each sensor may have many modes and search patterns. Typically, the sensors are used to gain information about the kinematic state (e.g. position and velocity) and identification of a group of targets. Applications of sensor management are often military in nature [7], but also include things such as wireless networking [12] and robot path planning [10]. There are many objectives that the sensor manager may be tuned to meet, e.g. minimization of track loss, probability of target detection, minimization of track error/covariance, and identification accuracy. Each of these different objectives taken alone may lead to a different sensor allocation strategy [7][8].

Many researchers have approached the sensor scheduling problem with a Markov decision process (MDP) strategy. However, a complete long-term (non-myopic) scheduling solution suffers from combinatorial explosion when solving practical problems of even moderate size. Researchers have thus worked at approximate solution techniques. Krishnamurthy [1][2] uses a multi-arm bandit formulation involving hidden Markov models. In [1], an optimal algorithm is formulated to track multiple targets with an ESA that has a single steerable beam. Since the optimal approach has prohibitive computational complexity, several suboptimal approximate methods are given and some simple numerical examples involving a small number of targets moving among a small number of discrete states are presented. Even with the proposed suboptimal solutions, the problem is still very challenging numerically. In [2], the problem is reversed, and a single target is observed by a single sensor from a collection of sensors. Again, approximate methods are formulated due to the intractability of the globally optimal solution. Bertsekas and Castanon [3][6] formulate heuristics for the solution of a stochastic scheduling problem corresponding to sensor scheduling. They implement a rollout algorithm based on their heuristics to approximate the stochastic dynamic programming algorithm. Additionally, Castanon [4][5] formulates the problem of classifying a large number of stationary objects with a multi-mode sensor based on a combination of stochastic dynamic programming and optimization techniques. Malhotra [13] proposes using reinforcement learning as an approximate approach to dynamic programming.

Recently, others have proposed using divergence measures as an alternative means of sensor management. In the context of Bayesian estimation, a good measure of the quality of a sensing action is the reduction in entropy of the posterior distribution that is induced by the measurement. Therefore, information theoretic methodologies strive to take the sensing action that maximizes the expected gain in information. The possible sensing actions are enumerated, the expected gain for each measurement is calculated, and the action that yields the maximal expected gain is chosen. Hintz et. al. [15][16] focus on using the expected change in Shannon entropy when tracking a single target moving in one dimension with Kalman Filters. A related approach uses discrimination gain based on a measure of relative entropy, the Kullback-Leibler

Chris Kreucher is with the University of Michigan and Veridian, both in Ann Arbor, Michigan (email: Christopher.Kreucher@veridian.com, US mail: Veridian's Ann Arbor Research and Development Center, P. O. Box 134008, Ann Arbor, MI 48113-4008, phone: (734) 994-1200 x2717)

Keith Kastella is with Veridian, Ann Arbor, Michigan (email: Keith.Kastella@veridian.com, US mail: Veridian's Ann Arbor Research and Development Center, P. O. Box 134008, Ann Arbor, MI 48113-4008)

Professor Al Hero is with the University of Michigan, Ann Arbor, Michigan (email: hero@eecs.umich.edu, US mail: Dept. Of Electrical and Computer Engineering, The University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109-2122.)

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

(KL) divergence. Schmaedeke [18] uses the KL divergence to determine optimal sensor-to-target tasking. Kastella uses the KL divergence to track multiple targets moving amongst discrete cells and for the purposes of target identification [9][14]. Zhao [24] compares several approaches, including simple heuristics, entropy, and relative entropy (KL).

Divergence-based adaptivity measures such as the KL divergence are a common learning metric that have been used in the machine learning literature in techniques with the names “active learning” [23], “learning by query” [21], “relevance feedback” [20], and “stepwise uncertainty reduction” [19]. These techniques are iterative procedures in which the system provides a set of items to the user as a query, the user indicates the relevance of the retrieved items, and the system adaptively chooses new queries based on the user feedback. The ultimate goal is to learn something from the user in an interactive manner.

A specific example of the role of divergence measures in machine learning is the interactive search of a database of imagery for a desired image. Cox et. al. [20] associates a probability of being the correct image to each image in the database. The probability mass function (pmf) is initially either uniformly distributed or peaked due to an informational prior. Queries are posed to the user based on entropy measures, and the pmf is updated according to Bayes’ rule. Similarly, Geman [19] studies the situation where a user has a specific image in mind and the system steps through a sequence of two-image comparisons to the user. The pair of images chosen by the system at each time is the query who’s answer may result in the lowest resulting Shannon entropy after the user responds.

Additionally, Zhai and Lafferty [22] use the KL divergence with feedback documents to improve estimation of query models in an application involving retrieval of documents from a text-based query. Freund et. al [21] study the rate that the prediction error decreases under divergence-based learning as a function of the number of queries for some natural learning problems. Finally, Geman and Jedynak [17] use expected entropy reduction as a means of learning the paths of roads in satellite imagery.

In the signal processing context of multitarget tracking, we use divergence-based methods to learn the number of targets present in the surveillance region as well as their kinematic states. We first utilize a target tracking algorithm to recursively estimate the joint multitarget probability density for the set of targets under surveillance. Analogous to the image/document retrieval applications, at each iteration of our algorithm we use a divergence-based metric to decide on the optimal query to pose. 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. In this work, we consider a more general information measure called the Rényi Information Divergence [25] (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 support of the densities to be compared. In our case the query takes the form of making a measurement with the sensor rather than asking the user whether one image or another is closer to the desired image. However in either case, the relevance of the query is fed back into the system and incorporated by Bayes’ rule.

A distinguishing feature of our application is that the goal is not a fixed entity, but rather a dynamic process that evolves over time. In other words, the kinematic states and number of targets change with time as the targets move through the surveillance region. We therefore include models of the evolution of the joint multitarget density into our Bayesian framework. These models are primarily kinematic in nature, but may include models of target birth and death to accommodate changing numbers of targets.

This paper contains two main contributions. First, we give a particle filter (PF) based multitarget tracking algorithm that by design explicitly enforces the multitarget nature of the problem. Each particle is a sample from the joint multitarget density and thus an estimate of the status of the entire system – the number of targets in the surveillance areas as well as their individual states. We find that the PF based multitarget tracker allows for successful tracking in a highly non-linear non-Gaussian filtering scenario. Furthermore, the PF implementation allows both target tracking and sensor management to be done in a computationally tractable manner. We demonstrate this by evaluating the sensor management scheme and tracking algorithm on a surveillance area containing ten targets, with target motion that is taken from real recorded target trajectories from an actual military battle simulation. Second, we detail a reinforcement learning approach to sensor management where the Rényi divergence is used as the method for estimating the utility of taking different actions. The sensor management algorithm uses the estimated density to predict the utility of a measurement before tasking the sensor, thus leading to actions which maximally gain information. We demonstrate that this method of sensor management yields a ten-fold increase in sensor efficiency over periodic scanning in the scenarios considered.

The paper is organized in the following manner. In Section II, 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 JMPD on a grid. In Section III, we present our particle filter based implementation of JMPD. We see that this implementation provides for computationally tractable implementation, allowing realistic scenarios to be considered. Our sensor management scheme, which is a learning algorithm that employs the Rényi divergence as a metric, is extensively detailed in Section IV. 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 V. We briefly illustrate the effect of non-myopic (long term) planning in this information theoretic context. We conclude with some thoughts on future direction in Section VI.

II. The Joint Multitarget Probability Density

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 Kastella [11] 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 $[x, \dot{x}, y, \dot{y}]'$. A simple schematic showing three targets (Targets A, B, and C) moving through a surveillance area is given in Figure 1. There are two target crossings, a challenging scenario for multitarget trackers.

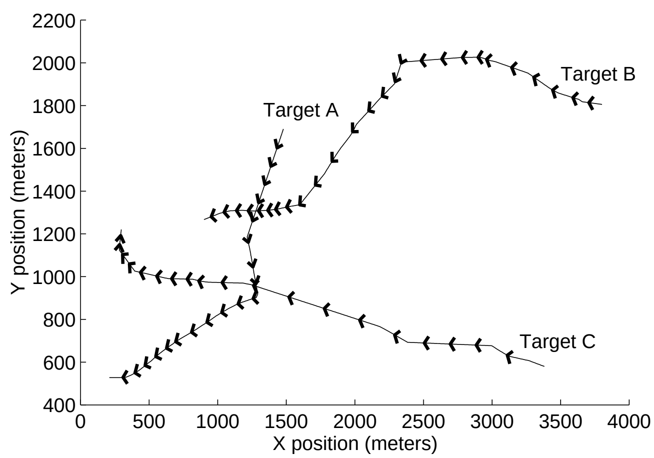

Fig. 1. A simple scenario involving three moving targets. The target paths are indicated by the lines, and direction of travel by the arrows. There are two instances where the target paths cross.

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 $p(\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k|\mathbf{Z}^k)$ as the probability density for exactly T targets with states $\mathbf{x}_1^k, \mathbf{x}_2^k, ...\mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ 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. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, ...\mathbf{z}^k\}$, 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 multitarget state $\mathbf{X}$. For convenience, the density will be written more compactly in the traditional manner as $p(\mathbf{X}^k|\mathbf{Z}^k)$, which implies that the state-vector $\mathbf{X}$ represents a variable number of targets each possessing their own state vector. As an illustration, some examples illustrating the sample space of p are

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

$p(\mathbf{x}_1|\mathbf{Z})$, the posterior probability density for one target with state $\mathbf{x}_1$

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

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

The JMPD is symmetric under permutation of the target indices. If 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

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

The time evolution of the JMPD may simply be a collection of target kinematic models or may involve target birth and death. In the case where target identification is part of the state being estimated, different kinematic models may be used for different target types.

Bayes' rule is used to update the posterior density as new measurements $\mathbf{z}^k$ arrive as

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

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 Kastella [11] approximated the density by discretizing on a grid. It was 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.

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

$$p(\mathbf{X}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \tag{3}$$

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 earlier by simulation [29]. Samples are used to represent the density and to propagate it through time. The prediction equation (eq. 1) 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. 2) is solved by assigning a weight to each of the particles that have been proposed using the measurements and the model of state dynamics.

The specific details of the PF implementation are as follows. Recall first from Section II that our multitarget state vector $\mathbf{X}$ has T partitions, each corresponding to a target, written explicitly in equation (4):

$$\mathbf{X} = [\mathbf{x}_1, \ \mathbf{x}_2, \ ..., \ \mathbf{x}_{T-1}, \ \mathbf{x}_T] \tag{4}$$

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 the JMPD $p(\mathbf{X}|\mathbf{Z})$. Therefore, a particle $\mathbf{X}_p$ may have $0, 1, ...\infty$ 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 $\mathbf{X}_p$. In practice, the maximum number of targets a particle may track in truncated at some large finite number T_{max}. 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

$$\mathbf{X}_p = [\mathbf{x}_{p,1}, \ \mathbf{x}_{p,2}, \ \ldots \ \mathbf{x}_{p,n_p}] \tag{5}$$

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:

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

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. This is particularly critical in the challenging scenario of target crossing. This paradigm is important for estimating the number of targets in the surveillance region.

The permutation symmetry discussed in Section II 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 and during estimation of target positions.

As detailed in another work [26], while following the standard sampling importance resampling method for particle filtering, we employ an adaptive sampling scheme for particle proposal that dramatically reduces the number of particles necessary to effectively track groups of targets. This scheme automatically factorizes the JMPD when targets or groups of targets are acting independently from the others (i.e. there is no measurement to target association issue) while maintaining the couplings when necessary. We also employ an adaptive resampling on the particle set based on the effective number of particles. It is shown therein that it is computationally tractable to track tens of targets moving over a large surveillance region with the particle filter implementation of JMPD.

IV. Relevance Feedback Learning for Sensor Management

In this section, we present the details of our information based method for sensor management. As mentioned earlier, our paradigm for sensor management is analogous to the machine learning methodologies present in relevance feedback techniques.

The goal of the multitarget tracker is to learn the number and states of a set of targets in a surveillance region. At each instance when a sensor is available for use, we use a divergence based method to compute the best sensing action to take (the query). This is done 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/pointing angle, or a combination of the two. Next, the *expected* information gain is calculated for each of the possible actions, and the action that yields the maximum expected information gain is taken. The measurement received is treated as the relevance feedback, analogous to a user selecting an image that is most relevant to his query. This measurement is used to update the JMPD, which is in turn used to determine the next measurement to make.

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:

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

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).

$$\lim_{\alpha\to1} D_\alpha(f_1||f_0) = \int f_0(x) ln \frac{f_0(x)}{f_1(x)} dx \qquad (8)$$

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

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

The function D_α given in (eq. 7) is a measure of the divergence between the 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

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

The integral in equation (7) reduces to a summation since any discrete approximation of $p(\mathbf{X}|\mathbf{Z}^{k-1})$, including our particle filter approximation, only has nonzero probability at a finite number of target states. After some algebra and the incorporation of Bayes' rule (eq. 2), one finds that this quantity can be simplified to

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

Our particle filter approximation of the density (eq. 3) reduces equation (11) to

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

where

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

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 the expected gain in information.

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

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

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

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

Which, using equation (12) results in

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

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. The measurement made is then fed back into the JMPD via Bayes' rule. Notice that this is a greedy scheme, which chooses to make the measurement that optimizes information gain only for the next time step.

A. 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 retrieval, georegistration of imagery, and target detection [27][28]. These studies have provided some guidance as to the optimal choice of the parameters α.

In the georegistration problem [27] 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 [28] 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.

These results give reason to believe that either $\alpha = 0.5$ or $\alpha = 1$ are good choices. We investigate the performance of our scheme under both choices in Section V.

B. Extensions to Non-Myopic Sensor Management

The sensor management algorithm proposed here is myopic in that it does not take into account long-term ramifications of the current sensing action when deciding the optimal action. If the dynamics of the problem change rapidly (either due to fast moving targets, moving sensors or a combination of the two), non-myopic scheduling is important. We propose then as a first step towards non-myopic sensor management a Monte Carlo rollout technique like that given by Castanon [3].

At each time a measurement decision is to be made, we first enumerate all possible measurements and the corresponding expected information gains. For each candidate measurement, we simulate making the measurement based on our estimated JMPD, update the density to the next time step based on the simulated measurement received, and compute the actual information gain received under this simulated measurement. We can then compute the expected gains of all possible measurements at the new time, and the actual gain received plus the maximum expected gain at the new time give the total information gain for making the particular measurement. Running this procedure many times gives a Monte Carlo estimate of the 2-step ramification of making a particular measurement. Extensions to n-step are straightforward, but computationally burdensome.

V. Simulation Results

In this section, we provide simulation results that show the benefit of sensor management in the multitarget tracking scenario. We first present a purely synthetic scenario and then proceed to a more realistic scenario using real recorded target trajectories from a military battle simulation. We conclude with some preliminary results on the benefit of non-myopic sensor scheduling.

A. An Extensive Evaluation of Sensor Management Performance Using Three Simulated Targets

We first gauge the performance of the sensor management scheme by considering the following model problem. There are three targets moving on a 12×12 sensor grid. Each target is modelled 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).

The target kinematics assumed by the filter (eq. 1) 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 equation (2). 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 [31]. 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 the performance of the tracker when the sensor uses a non-managed (periodic) scheme with the performance when the sensor uses the relevance feedback based management scheme presented in Section IV. 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.

Figure 2 presents a single-time tracker snapshot, which graphically illustrates the difference in behavior between the two schemes.

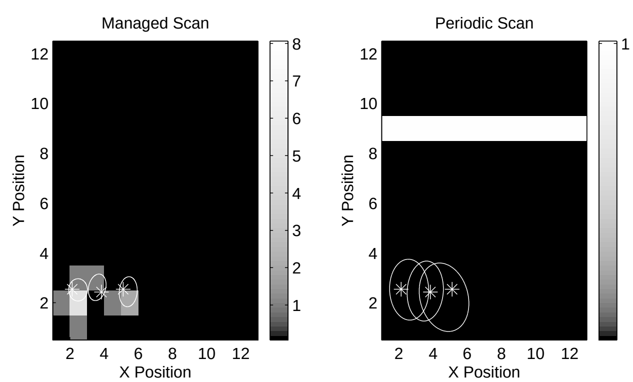

Fig. 2. A comparison of non-managed and managed tracking. (L) Using sensor management, and (R) Using a periodic scheme. Targets are marked with an asterisk, the covariance of the filter estimate is given by the ellipse, and grey scale is used to indicate the number of times each cell has been measured at this time step (the same total number of looks is used in each scenario). With sensor management, measurements 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 examination is provided in the Monte Carlo simulation results of Figure 3. We refer to each cell that is measured as a "Look", and are interested in empirically determining how many looks the non-managed algorithm requires to achieve the same performance as the managed algorithm at a fixed number of looks. The sensor management algorithm was run with 24 looks (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 $\alpha = 0.99999$ (approximately the KL divergence) in equation (8). It is found that the non-managed 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.

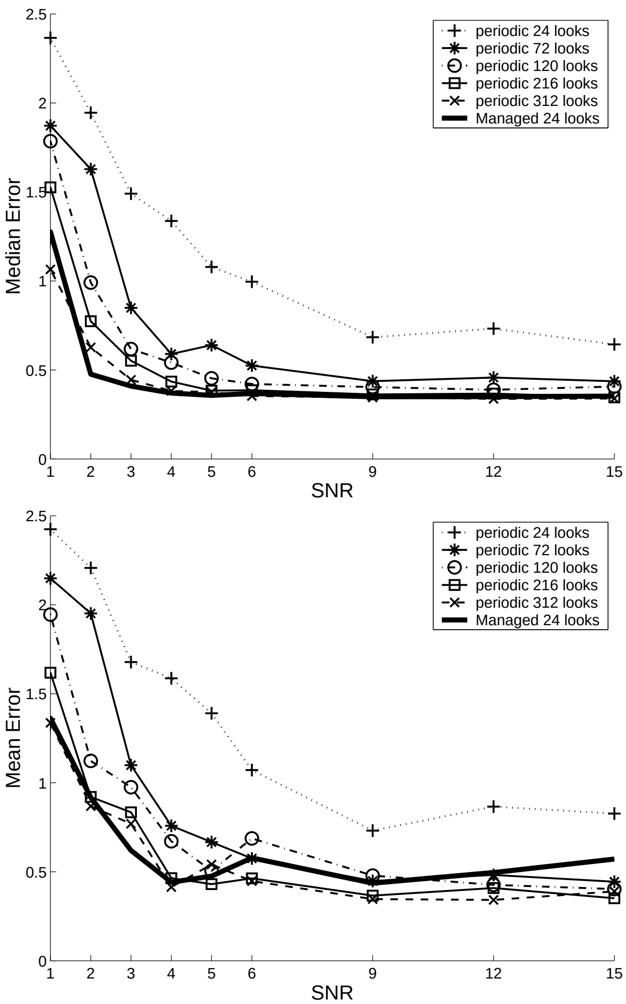

Fig. 3. Median and mean error versus signal to noise ratio (SNR). Managed performance with 24 looks is similar to non-managed with 312 looks.

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 4 shows that in this case, where the actual target motion is very well modelled 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.

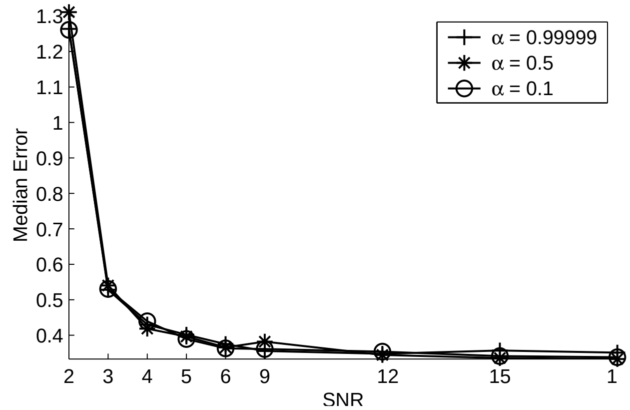

Fig. 4. The performance of the sensor management algorithm with different values of α. We find that in the case where the filter dynamics match the actual target dynamics, the algorithm is insensitive to the choice of α.

B. 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 $5000m \times 5000m$ 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 the Army's National Training Center. Targets routinely come within sensor cell resolution (i.e. crossing trajectories). Therefore, there is often measurement to track ambiguity, which is handled automatically by JMPD since there is no measurement to track assignment necessary. Target positions are recorded at 1 second intervals, and the simulation duration is 1000 time steps. An image showing the initial target positions and the road network on which the targets travel is given in Figure 5.

The filter again assumes constant velocity motion with a large diffusive component as the model of target kinematics. However, in this case, the 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 which 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 $100\text{m}\times100\text{m}$ 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

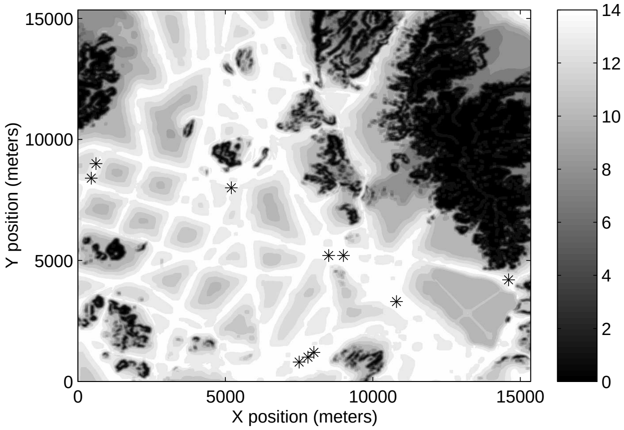

Fig. 5. An image showing the ten-target case under consideration, where the initial positions of the targets are each marked with an asterisk. The backdrop is the hospitability – which is a military derived product that indicates the drivability of the terrain. Road networks are visible (high drivability, hence a white color) in the hospitability.

of measurements is returned, corresponding to the depth of the beam. In this simulation, 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).

As in the previous simulation, 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. In this illustration, we take $P_d = 0.5$, $SNR = 10dB$, and $P_f = P_d^{(1+SNR)}$.

We compare the performance of the managed and non-managed scenarios in Figure 6. Our method of comparison again is to determine empirically the number of looks needed in the non-managed scenario to achieve the same performance as the managed algorithm with $L = 50$ looks.

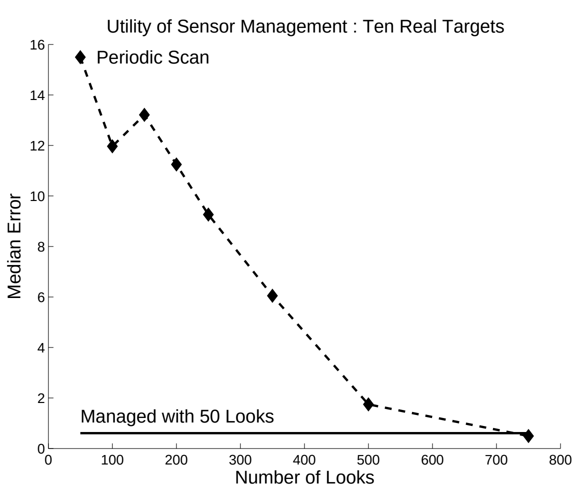

Fig. 6. Median error versus number of looks. managed performance with 50 looks performs similarly to non-managed with 700 looks.

Figure 6 shows that the non-managed scenario needs approximately 700 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.

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 V-A for several reasons (e.g. number of targets, number of target crossing events, and model mismatch). 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 (see Section IV-A) leads us to believe that $\alpha = 0.5$ is the right choice in this scenario.

In Figure 7, 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 I. 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 $\alpha \approx 1$ and $\alpha = 0.1$ case lose track of targets on several occasions.

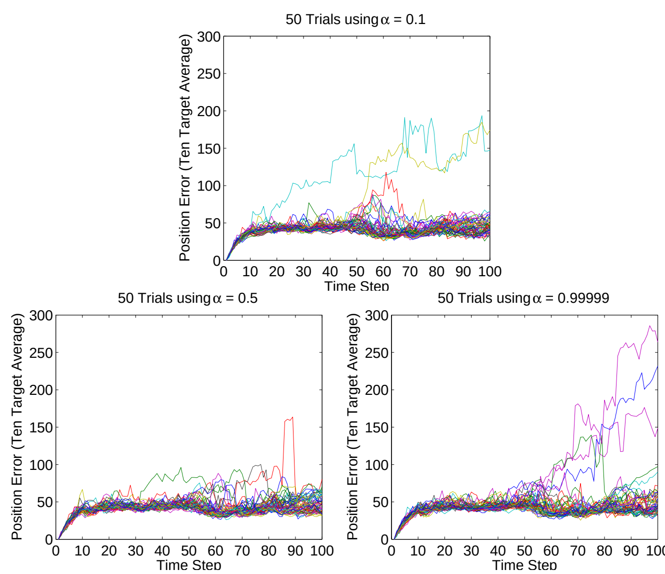

Fig. 7. A comparison of sensor management performance under different values of the Rényi divergence parameter, α.

TABLE I

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

C. *The Effect of Non-Myopic Scheduling*

Finally, we give some preliminary results on the ramifications of non-myopic (long-term) sensor management on algorithm performance. We inspect a challenging scenario in which the sensor is prevented from seeing half of the region every other time step. At even time steps, all of the targets are visible; at odd time steps only half of the targets are visible. For the purposes of exposition, we assume that this pattern is fixed and known ahead of time by the sensor manager.

The myopic (greedy) management scheme simply measures the targets whose expected information gain is highest at the current time. This implies that at odd time steps it will only measure targets that are visible to the sensor, but at even time steps will have no preference as to which targets to measure. Intuitively, we would like the manager to measure targets that are about to become obscured from the sensor preferentially, since the system must wait two time steps to have an opportunity to revisit.

The non-myopic sensor management technique discussed in IV-B takes the dynamics of the scene into account. When making a measurement at even time steps it prefers to measure those targets that will be invisible at the next time step, because it rolls out the ramifications of its action and determines the best action to take is to measure targets that are about to become obscured since this will result in the maximum total (2-step) information gain.

We show in Figure 8 the results of tracking in this challenging scenario. It turns out that it is only modestly important to be non-myopic. Myopic sensor scheduling results in loss of track approximately 22% of the time, while non-myopic scheduling only loses track 11% of the time. It is especially important to be non-myopic around time step 150, where the dynamics of the problem accelerate due to the speed up of some of the targets.

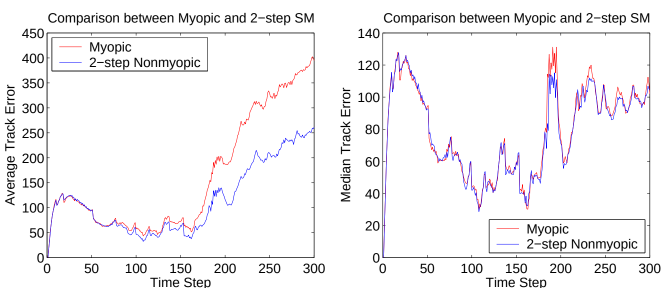

Fig. 8. A comparison of sensor management performance in the myopic (greedy) case and in the 2-step non-myopic case.

VI. Discussion

We have applied an approach that is common in the machine learning literature, known as relevance feedback learning, to provide a method for managing agile sensors. 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 via particle filtering. 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.

References

[1] V. Krishnamurthy and D. Evans, "Hidden Markov Model Multiarm Bandits: A Methodology for Beam Scheduling in Multitarget Tracking", *IEEE Transactions on Signal Processing*, Vol. 49, no. 12, pp. 2893-2908, December 2001.

[2] V. Krishnamurthy, "Algorithms for Optimal Scheduling and Management of Hidden Markov Model Sensors", *IEEE Transactions on Signal Processing*, Vol. 50, no. 6, pp. 1382-1397, June 2002.

[3] D. P. Bertsekas and D. Castanon, "Rollout Algorithms for Stochastic Scheduling Problems", *Journal of Hueristics*, Vol. 5, no. 1, pp. 89-108, 1999.

[4] D. Castanon, "Approximate Dynamic Programming for Sensor Management", *Proceedings of the 1997 Conference on Decision and Control*, 1997.

[5] D. Castanon, "Optimal Search Strategies for Dynamic Hypothesis Testing", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 25, no. 7, pp. 1130-1138, 1995.

[6] D. P. Bertsekas and D. A. Castanon, "Rollout Alogorithms for Stochastic Scheduling Problems", *Proceedings of the 37th IEEE Conference on Decision and Control*, pp. 2143-2148, Tampa, FL, 1998.

[7] S. Musick and R. Malhotra, "Chasing the Elusive Sensor Manager", *Proceedings of NAECON*, Dayton, OH, May 1994, pp. 606-613.

[8] R. Popoli, "The Sensor Management Imperative", in *Multitarget-Multisensor Tracking: Advanced Applications*, vol. II, Y. Bar-Shalom, editor, Boston, MA: Artech House, 1992.

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

[10] V. J. Lumelsky, S. Mukhopadhyay, and K. Sun, "Dynamic Path Planning in Sensor-Based Terrain Acquisition, IEEE Transactions on Robotics and Automation, vol. 6, no. 4, pp. 462-472, August 1990.

[11] K. Kastella, "Joint multitarget probabilities for detection and tracking", SPIE Proceedings, Acquisition, Tracking and Pointing XI, 21-25 April, 1997, Orlando, FL.

[12] J. Liu, P. Cheung, L. Guibas, and F. Zhao, "A Dual-Space Approach to Tracking and Sensor Management in Wireless Sensor Networks", ACM International Workshop on Wireless Sensor Networks and Applications Workshop, Atlanta, September 2002.

[13] R. Malhotra, "Temporal Considerations in Sensor Management", Proceedings of the IEEE 1995 National Aerospace and Electronics Conference, NAECON 1995, vol. 1, Dayton, OH, May 22-26 1995, pp. 86-93.

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

[15] K. J. Hintz and E. S. McVey, "Multi-Process Constrained Estimation", *IEEE Transactions on Man, Systems, and Cybernetics*, Vol. 21, no. 1, pp. 434-442, January/February 1991.

[16] K. J. Hintz, "A Measure of the Information Gain Attributable to Cueing", IEEE Transaction on Systems, Man, and Cybernetics, vol. 21, no. 2, pp. 237-244, March/April 1991.

[17] D. Geman and B. Jedynak, "An Active Testing Model for Tracking Roads in Sattelite Images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, no. 1, pp. 1-13, January, 1995.

[18] W. Schmaedeke, "Information Based Sensor Management", Signal Processing, Sensor Fusion, and Target Recognition II. Proceedings of the SPIE - The International Society for Optical Engineering, vol. 1955, Orlando, FL, April 12-14 1993, pp. 156-64.

[19] D. Geman and R. Moquet, "Q and A Models for Interactive Search", Technical Report, University of Massachusetts at Amherst, December 2000.

[20] I. J. Cox, M. L. Miller, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments", *IEEE Transactions on Image Processing*, Vol. 9, no. 1, pp. 20-37, January 2000.

[21] Y. Freund, H. S. Seung, E. Shamir and N. Tishby, "Selective Sampling Using the Query by Committee Algorithm", *Machine Learning*, Vol. 28, pp. 138-168, 1997.

[22] C. Zhai and J. Lafferty, "Model-based feedback in the KL-divergence retrieval model", *Proceedings of The Tenth International Conference on Information and Knowledge Management*, pp. 403410, 2001.

[23] S. Tong and D. Koller, "Active Learning for Parameter Estimation in Bayesian Networks", *Proceedings of the Thirteenth Annual Conference on Neural Information Processing Systems*, pp. 647-653, Denver, CO, December 2000.

[24] F. Zhao, J. Shin, and J. Reich, “Information-Driven Dynamic Sensor Collaboration”, IEEE Signal Processing Magazine, March 2002, pp. 61-72.

[25] A. Rényi, “On measures of entropy and information”, Proc. 4th Berkeley Symp. Math. Stat. and Prob., volume 1, pp. 547-561, 1961.

[26] 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.

[27] 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.

[28] 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.

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

[30] B. Horling, and R. Vincent, R. Mailler, J. Shen, R. Becker, K. Rawlins and V. Lesser, “Distributed Sensor Network for Real Time Tracking”, *Proceedings of the 5th International Conference on Autonomous Agents*, pp. 417-424, June, 2001.

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