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 [27], but also include things such as wireless networking [23] and robot path planning [24]. 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 [27][29].

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. For Example, Krishnamurthy [22][21] uses a multi-arm bandit formulation involving hidden Markov models. In [22], 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 [21], 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 [1] 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 [3][4] 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 [25] 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. [14][13] 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

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.

measure of relative entropy, the Kullback-Leibler (KL) divergence. Schmaedeke and Kastella [31] use the KL divergence to determine optimal sensor-to-target tasking. Kastella [17][19] uses KL divergence to manage a sensor between tracking and identification mode in the multitarget scenario. Others use similar information based approaches in the context of active vision [6][32]. Zhao [37] 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” [35], “learning by query” [8], “relevance feedback” [38][5], and “stepwise uncertainty reduction” [10]. 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, also called content based image retrieval (CBIR). Cox et. al. [5] associates a probability of being the correct image to each image in a database. The probability mass function (pmf) is initially either uniformly distributed or peaked due to an informational prior. Psychophysical experiments are used to develop a probabilistic model for how a human judges images to be similar (the ”sensor” model). Quantities such as intensity, color, and edges are found to be important. At each iteration of the algorithm, queries are posed to the user based on entropy measures, the human responds, and the pmf is updated according to Bayes’ rule. Similarly, Geman [10] 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 whose answer may result in the lowest resulting Shannon entropy after the user responds.

Additionally, Zhai and Lafferty [36] 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 [8] 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 [9] 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 states. This is analogous to learning the target image in a CBIR application. We first utilize a target tracking algorithm to recursively estimate the joint multitarget probability density for the set of targets under surveillance. In the CBIR application, the goal (the image) is a fixed entity and therefore no such tracking algorithm is necessary. In our application, the goal (the number and states of the targets) is a dynamic process that evolves over time. The kinematic states and number of targets change as the targets move through the surveillance region. Therefore we include a model of the evolution of the joint multitarget density into our framework.

In a manner similar to the way query images are chosen in the CBIR application, 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 [30] (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. To the best of our knowledge, this is the first time Rényi divergence has been used in this setting. In contrast to CBIR, our query takes the form of making a measurement with a physical sensor rather than asking the user whether one image or another is closer to the desired image. Our physical sensor is able to be modelled quite precisely, while modelling a human ”sensor” is quite difficult [5]. In either the human or the physical sensor case, the relevance of the query is fed back into the system and incorporated by Bayes’ rule.

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 (JMPD) 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, primarily due to our use of an adaptive sampling scheme for particle proposal that automatically factorizes the JMPD when possible. We demonstrate the algorithm 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 illustrate the efficacy of this algorithm in a scenario where processed sensor measurements consist of detections or no-detections, which leads to a computationally efficient algorithm for tasking the sensor. We show that this method of sensor management yields a ten-fold increase in sensor efficiency over periodic scanning in 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 performance analysis of the tracker using sensor management on two model problems of increasing realism is given in Section V. We include comparisons to a non-managed (periodic) scheme and two other sensor management techniques. 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. Others have studied Bayesian methods for tracking multiple targets [33][26]. The concept of JMPD was discussed by Kastella [17] where a method of tracking multiple targets that moved between discrete cells on a line was presented. The JMPD is a continuous-discrete hybrid system. 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.

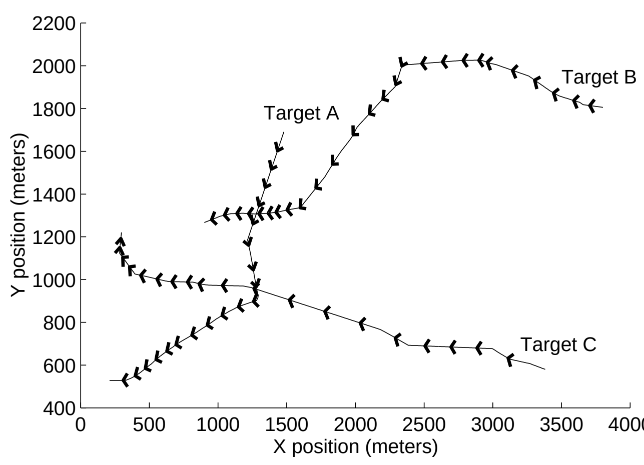

Fig. 1. A simple scenario involving three moving targets. The target paths are indicated by the lines, and direction of travel by the arrows. There are two instances where the target paths cross.

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 $\mathbf{x}_3$

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.

The JMPD is symmetric under permutation of the target indices. This symmetry is a fundamental property of the JMPD and not related to any assumptions on the indistinguishability of targets. The multitarget state $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2]$ and $\mathbf{X} = [\mathbf{x}_2, \mathbf{x}_1]$ refer to the same event, namely there are two targets – one with state $\mathbf{x}_1$ and one with state $\mathbf{x}_2$. This is true regardless of the makeup of the single target state vector. For example, the single target state vector may include target ID or even a target serial number and the permutation symmetry remains. Therefore, all algorithms designed to implement the JMPD (and algorithms that implement the relevance feedback learning based sensor management) are permutation invariant. Proper treatment of this permutation symmetry has a significant impact on how to implement particle sampling schemes, as described in the Appendix.

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 into a product of single target densities.

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 over time $p(\mathbf{X}^k|\mathbf{X}^{k-1})$, we may compute the time-updated or prediction density via

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

$p(\mathbf{X}^k|\mathbf{Z}^{k-1})$ is referred to as the prior or prediction density at time k, as it is the density at time k conditioned on measurements up to and including time $k-1$. The time evolution of the JMPD may be a collection of target kinematic models or may involve target birth and death. In the case where target identity is part of the state being estimated, different kinematic models may be used for different target types.

Given a model of the sensor, $p(\mathbf{z}^k|\mathbf{X}^k)$, and assuming conditional independence of the measurements given the state, Bayes' rule is used to update the posterior density as new measurements $\mathbf{z}^k$ arrive via

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

$p(\mathbf{X}^k|\mathbf{Z}^k)$ is referred to as the posterior or the updated density at time k as it is the density at time k conditioned on all measurements up to and including time k.

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 implementation of JMPD given by Kastella [18] 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. Other authors [16][34] have investigated using particle filter algorithms to approximate a multi-object density in the context of computer vision. The algorithm that we present here introduces an adaptive sampling scheme that substantially increases the efficiency of particles so as to allow tracking of large numbers of objects with a relatively few number of particles.

To implement JMPD via a particle filter (PF), we approximate the joint multitarget probability density $p(\mathbf{X}|\mathbf{Z})$ by a set of N_{part} weighted samples (particles). First, let the multitarget state vector be written

$$\mathbf{X} = [\mathbf{x}_1,\ \mathbf{x}_2,\ ...,\ \mathbf{x}_{T-1},\ \mathbf{x}_T] \quad (3)$$

and be defined for all T, $T = 1...\infty$. Next, let the particle state vector be written

$$\mathbf{X}_p = [\mathbf{x}_{p,1},\ \mathbf{x}_{p,2},\ \ldots\ \mathbf{x}_{p,T_p}] \quad (4)$$

where T_p is the estimate particle p has for the number of targets in the surveillance region. Letting δ_D denote the usual Dirac delta where it is understood that it is defined on the domain of its argument (i.e. finite dimensional real or complex vector), we define

$$\delta(\mathbf{X} - \mathbf{X}_p) = \begin{cases} 0 & T \neq T_p \\ \delta_D(\mathbf{X} - \mathbf{X}_p) & \text{otherwise} \end{cases} \quad (5)$$

Then the particle filter approximation to the JMPD is given by

$$p(\mathbf{X}|\mathbf{Z}) \approx \sum_{p=1}^{N_{part}} w_p \delta(\mathbf{X} - \mathbf{X}_p) \quad (6)$$

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

Particle filtering is a method of approximately solving the prediction and update equations by simulation [7]. Samples are used to represent the density and to propagate it through time. The prediction equation (eq. 1) is implemented by proposing new particles from the existing set of particles using a model of state dynamics and the measurements. The update equation (eq. 2) is implemented by assigning a weight to each of the particles that have been proposed using the measurements and the model of state dynamics.

To make our notation more concrete, assume that a particular particle, $\mathbf{X}_p$, is tracking T_p targets. In the case where each target is modelled using the state vector $\mathbf{x} = [x, \dot{x}, y, \dot{y}]'$, the particle will have T_p partitions each of which has 4 components:

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

Notice that this method differs from traditional particle filter tracking algorithms where a single particle corresponds to a single target. The single target/single particle model is inappropriate in the multitarget scenario. If each particle is attached to a single target, some targets will become particle starved over time. All of the particles tend to attach to the target receiving the best measurements. This 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 important in the challenging scenario of target crossing, and 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 are 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. Algorithms for particle proposal, sensor management and estimation of target parameters must all be permutation invariant.

Representing the full joint multitarget density rather than merely using a factorized representation provides the advantage that correlations between targets are explicitly modelled. However, due to the dramatic increase in dimensionality, a simplistic implementation leads to greatly increased computational burden. The key to computational tractability of the particle filter algorithm presented here is an adaptive sampling scheme for particle proposal that automatically factorizes the JMPD when targets or groups of targets are acting independently from the others (i.e. when there is no measurement to target association ambiguity), while maintaining the couplings when necessary. Our multi-partition proposal scheme is outlined in the Appendix and more thoroughly in [20].

Estimating the multitarget states from the particle filter representation of JMPD must be done in a way that is invariant to permutations of the particles. Before estimating target states, we permute the particles so that each of the particles has the targets in the same order. We use the K-means algorithm to cluster the partitions of each particle, where the optimization is done across permutations of the particles. In practice, this is a very light computational burden. First, those partitions that are not coupled (see the Appendix) are already correctly ordered and are not involved in the clustering procedure. Second, since this ordering occurs at each time step, those partitions that are coupled are nearly ordered already, and so one iteration of the K-means algorithm is enough to find the best permutation.

IV. Relevance Feedback Learning for Sensor Management

The goal of the multitarget tracker is to learn the number and states of a set of targets in a surveillance region. This goal is to be obtained as quickly and accurately as possible by using the sensor in the best manner possible. A good measure of the quality of each sensing action is the reduction in entropy of the posterior distribution that is expected to be induced by the measurement. Therefore, at each instance when a sensor is available, 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 (e.g. SAR mode or 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. This measurement is used to update the JMPD, which is in turn used to determine the next measurement to make.

As mentioned earlier, our paradigm for sensor management is analogous to the machine learning methodologies present in relevance feedback techniques such as content based image retrieval (CBIR). The central element in both is an estimate of a density (in our case a multitarget density). This estimate is used to determine the query to perform based on maximizing a divergence measure. The response to the query (measurement) is then fed back into the system to further refine the estimate of the multitarget density.

The calculation of information gain between two densities f_1 and f_0 is done using the Rényi information divergence [30][12], also known as the α-divergence:

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha-1} ln \int f_1^\alpha(x) f_0^{1-\alpha}(x)dx \qquad (8)$$

The α parameter in equation (8) 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 (9).

$$\lim_{\alpha\to1} D_\alpha(f_1||f_0) = \int f_0(x) ln \frac{f_0(x)}{f_1(x)}dx \qquad (9)$$

In the case that $\alpha = 0.5$, the Rényi information divergence is related to the Hellinger-Battacharya distance squared [11]

$$d_H(f_1, f_0) = \frac{1}{2}\int \left(\sqrt{f_1(x)} - \sqrt{f_0(x)}\right)^2 dx \qquad (10)$$

The function D_α given in (eq. 8) 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 $p(\mathbf{X}^k|\mathbf{Z}^{k-1})$ and the updated density after a measurement is made, $p(\mathbf{X}^k|\mathbf{Z}^k)$. Therefore, we write

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

The symbol $\sum_{\mathbf{X}} f(\mathbf{X})$ is intended to denote the integral over the domain. This can be precisely written as

$$\int d\mathbf{X} f(\mathbf{X}) \doteq \sum_{T=0}^{\infty} \int d\mathbf{x}_1...\mathbf{x}_T f(\mathbf{X}) \qquad (12)$$

After some algebra and the incorporation of Bayes' rule (eq. 2), equation (11) can be simplified to

$$D_\alpha\left(p(\mathbf{X}^k|\mathbf{Z}^k)||p(\mathbf{X}^k|\mathbf{Z}^{k-1})\right) = \frac{1}{\alpha-1} ln \frac{1}{p(\mathbf{z}|\mathbf{Z}^{k-1})^\alpha} \sum_{\mathbf{X}} p(\mathbf{X}^k|\mathbf{Z}^{k-1}) p(\mathbf{z}|\mathbf{X}^k)^\alpha \qquad (13)$$

The integral over the domain reduces to a summation since any discrete approximation of $p(\mathbf{X}^k|\mathbf{Z}^{k-1})$ only has nonzero probability at a finite number of target states. In the particle filter case, the approximation consists of only a set of samples and associated weights from the density. In the special case where the positions of the particles in both sets are identical (which they are in this application since the two densities differ only in that one has been measurement updated and one has not) it is possible to compute the divergence by straightforward calculation.

Our particle filter approximation of the density (eq. 6) reduces equation (13) to

$$D_\alpha \left(p(\mathbf{X}^k|\mathbf{Z}^k)||p(\mathbf{X}^k|\mathbf{Z}^{k-1})\right) = \frac{1}{\alpha-1} ln \frac{1}{p(\mathbf{z})^\alpha} \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X_p})^\alpha \tag{14}$$

where

$$p(\mathbf{z}) = \sum_{p=1}^{N_{part}} w_p p(\mathbf{z}|\mathbf{X}_p) \tag{15}$$

We note here that the sensor model $p(\mathbf{z}|\mathbf{X}_p)$ is used to incorporate everything known about the sensor, including signal to noise ratio, 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, $p(\mathbf{X}^k|\mathbf{Z}^k)$, with respect to the density before a measurement was made, $p(\mathbf{X}^k|\mathbf{Z}^{k-1})$.

We propose, then, as a method of sensor management calculating the expected value of equation (14) 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 (14) may be written as an integral over all possible outcomes z_m when performing sensing action m:

$$< D_\alpha >_m = \int d\mathbf{z}_m p(\mathbf{z}_m|\mathbf{Z}^{k-1}) D_\alpha \left(p(\mathbf{X}^k|\mathbf{Z}^k)||p(\mathbf{X}^k|\mathbf{Z}^{k-1})\right) \tag{16}$$

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

$$< D_\alpha >_m = p(z=0|\mathbf{Z}^{k-1}) D_\alpha|_{m,z=0} + p(z=1|\mathbf{Z}^{k-1}) D_\alpha|_{m,z=1} \tag{17}$$

Which, using equation (14) results in

$$< D_\alpha >_m = \frac{1}{\alpha-1} \sum_{z=0}^{1} p(z) ln \frac{1}{p(z)^\alpha} \sum_{p=1}^{N_{part}} w_p p(z|\mathbf{X}_p)^\alpha \tag{18}$$

Computationally, the value of equation (18) can be calculated for M possible sensing actions in $O(MN_{part})$. Notice that the sensor management algorithm is permutation invariant as it only depends on the likelihood of the measurements given the particles.

We have specialized here to the case where the measurements are thresholded (binary), but make the following general comments about the extension to more complicated scenarios. It is straightforward to extend the binary case to a situation where the measurement z may take on one of a finite number of values. This would be relevant in a situation where, for example, raw sensor returns are passed through an automatic target recognition algorithm and translated into target identifications that come from a discrete set of possibilities. In the case where z is continuous valued, the integral of equation (16) would have to be solved approximately, perhaps using the same importance sampling strategy that the particle filtering technique uses to solve equations (1) and (2).

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 (18) for each possible sensing action m. We then perform the sensing action that gives maximal expected information gain. The measurement made is fed back into the JMPD via Bayes' rule.

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, image georegistration, and target detection [12][11]. These studies provide guidance as to the optimal choice of α.

In the georegistration problem [12] 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 and the Hellinger affinity 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 ($\alpha = 1$). These empirical 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.

An asymptotic analysis [12] shows that $\alpha = .5$ results in the maximum discriminatory ability between two densities that are very similar. The value $\alpha = .5$ provides a weighting which stresses the tails, or the minor differences, between two distributions. In the case where the two densities of interest are very similar (as in our application where one is a prediction density and one is a measurement updated density), the salient differences are in the regions of low probability, and therefore we anticipate that this choice of α will yield the best results.

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. In some scenarios, the greedy approach may be close to optimal. However, in scenarios where the dynamics of the problem are changing in a predictable manner, tracking performance may benefit from non-myopic scheduling. For example, if a target is about to become invisible to a sensor (e.g. by passing into an area where the target to sensor line of sight is obstructed) extra sensor dwells should be tasked immediately before the target disappears. This will tend to reduce the uncertainty about this target at the expense of the other targets, but is justified because the target will be unable to be measured at the next epoch due to obstruction. Our ability to predict times when targets will become invisible is of course tied to having accurate ancillary information, such as sensor trajectories and ground elevation maps. We propose as a first step towards non-myopic sensor management a Monte Carlo rollout technique like that given by Castanon [1].

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.

It should be noted that due to the nature of the sensor management problem, the number of decision trajectories is exponential in the number of time steps that the algorithm wishes to look ahead. However, as many of these trajectories are clearly poor paths, optimization techniques (e.g. A-stars) may be useful to prune the set of trajectories that need to be searched to find the best path.

V. Simulation Results

In this section, we provide simulation results to show the benefit of sensor management in the multitarget tracking scenario. We first present a synthetic scenario and then proceed to a more realistic scenario using real recorded target trajectories from a military battle simulation. In both cases, we assume the sensor is limited by time, bandwidth and other physical constraints which only allow it to measure a subset of the surveillance area at any epoch. We conclude with preliminary results on the benefit of non-myopic sensor scheduling.

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

We 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 large plant noise. Motion for each target is independent. The trajectories have been shifted and time delayed so there are two times during the simulation where targets cross paths (i.e. come within sensor resolution).

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) which is governed by a probability of detection (P_d) and a per-cell false alarm rate (P_f). The signal to noise ratio (SNR) links these values together. 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 [2]. When there are T targets in the same cell, the detection probability increases according to $P_d(T)$=$P_d^{\frac{1+SNR}{1+T*SNR}}$. This model is known by the filter and used to evaluate equation (2). 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. Multiple measurements made in the same cell are independent (i.e. each measurement in a target containing cell returns a detection with probability P_d irrespective of whether earlier measurements resulted in a detection).

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

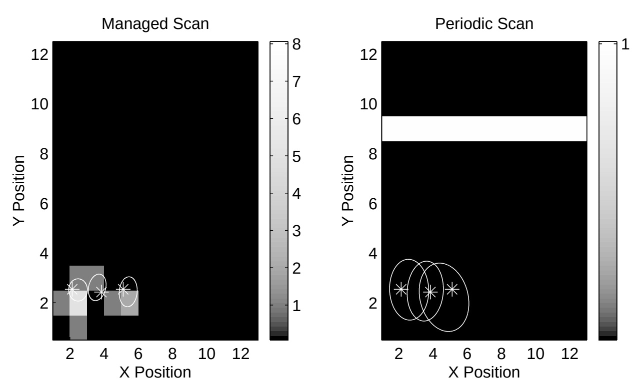

Fig. 2. Comparison of managed and non-managed tracking performance. (L) Using sensor management, and (R) A periodic scheme. Targets are marked with an asterisk, the covariance of the filter estimate is given by the ellipse, and grey scale is used to indicate the number of times each cell has been measured at this time step (the total number of looks is identical in each scenario). In the periodic scenario, one twelfth of the region is scanned at each time step starting at the bottom and proceeding to the top before repeating (cells scanned at this epoch are indicated by the white stripe). With sensor management, measurements are used only in areas that contain targets.

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 (9). It is found that the non-managed scenario needs approximately 312 looks to equal the performance of the managed algorithm in terms of RMS error. Multitarget RMS position error is computed by taking the average RMS error across all targets. 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.

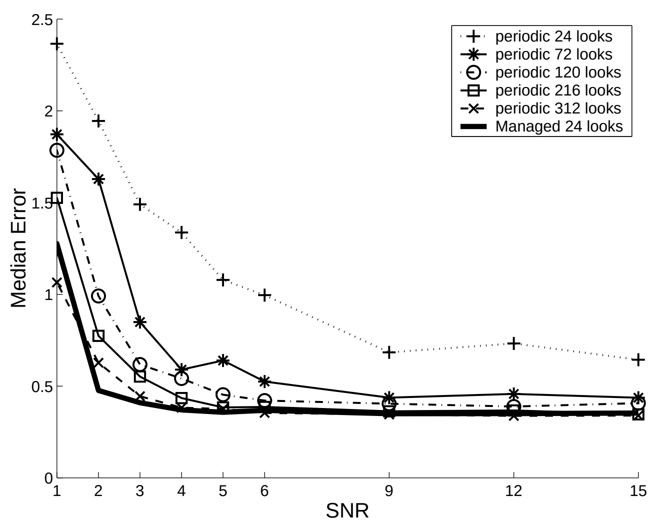

Fig. 3. The median error versus signal to noise ratio (SNR). Managed performance with 24 looks is similar to non-managed with 312 looks.

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.

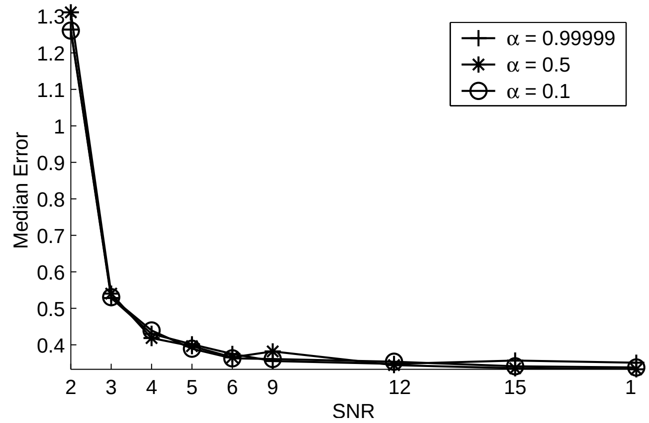

Fig. 4. The performance of the sensor management algorithm with different values of α. We find that in the case where the filter dynamics match the actual target dynamics, the algorithm is insensitive to the choice of α.

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. Images showing the road network and the positions of the targets at three different times is given in Figure 5.

The filter again assumes constant velocity motion with large plant noise 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 previously. 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 $100\text{m} \times 100\text{m}$ 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 ten resolution cells deep. Each time a beam is formed, a vector of measurements (a vector zeros and ones corresponding to non-detections and detections) is returned, one measurement for each of the ten resolution cells. In this simulation, we refer to each beam that is laid down as a “Look”.

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

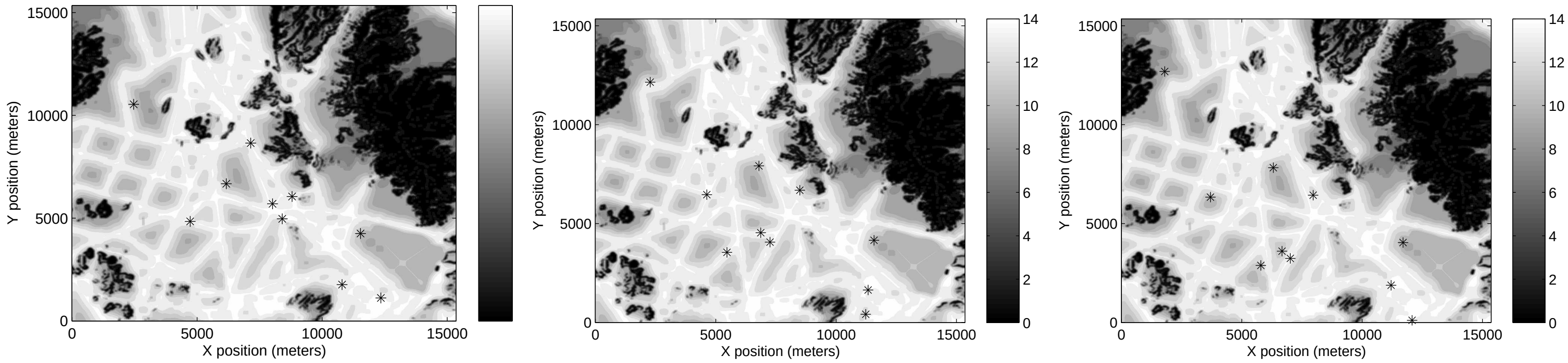

Fig. 5. Three time sequential snapshots showing the ten-target case under consideration. The positions of the targets are each marked with an asterisk. The backdrop is the hospitability – a military product that indicates the terrain drivability. Road networks are visible (high drivability, hence a white color).

either a 0 (no detection) or a 1 (detection) governed by P_d, P_f, and SNR. In this illustration, we take $P_d = 0.5$, $SNR = 2$, and $P_f = P_d^{(1+SNR)}$. When there are T targets in the same cell, the detection probability increases according to $P_d(T)$=$P_d^{\frac{1+SNR}{1+T*SNR}}$.

We compare first the performance of the sensor management algorithm under different values of α in equation (8). 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 6, 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 lose track on any of the 10 targets during any of the 50 simulation runs. We define the track to be lost when the filter error remains above 100 meters after some point in time. Both the $\alpha \approx 1$ and $\alpha = 0.1$ case lose track of targets on several occasions.

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

Due to the asymptotic analysis and these empirical results, we employ $\alpha = .5$ for the rest of the comparisons involving this scenario.

In addition to a comparison between the divergence based sensor management algorithm and a naive periodic scheme, we consider two additional methods of sensor management.

Sensor management algorithm "A" manages the sensor by pointing it at or near the estimated location of the targets. Specifically, algorithm "A" performs a gating procedure to

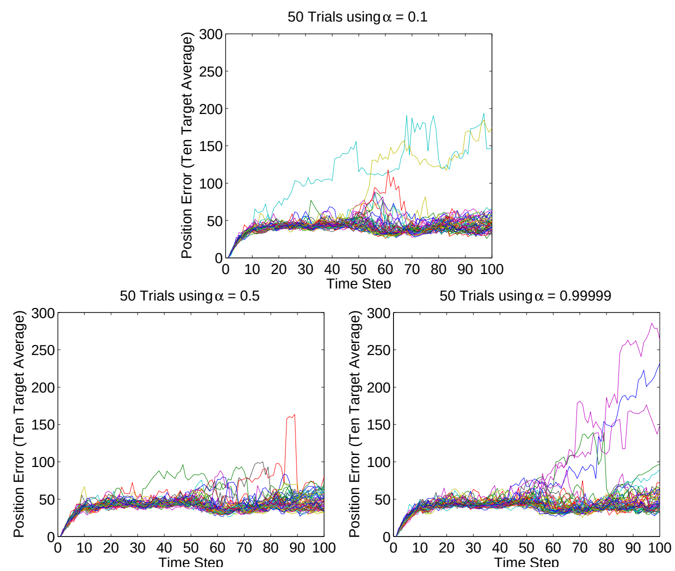

Fig. 6. A comparison of sensor management performance under different values of the Rényi divergence parameter, α.

restrict the portion of the surveillance area that the sensor will consider measuring. The particle filter approximation of the time updated JMPD (equation 1) is used to predict the location of each of the targets at the current time. The set of cells that the sensor manager considers is then restricted to those cells containing targets plus the surrounding cells, for a total of 9 cells in consideration per target. The dwells are then allocated randomly among the gated cells.

Sensor management algorithm "B" tasks the sensor based on the estimated number of targets in each sensor cell. Specifically, the particle approximation of the time updated JMPD is projected into sensor space to determine the filter's estimate of the number of targets in each sensor cell. The cell to measure is then selected probabilistically, favoring cells that are estimated to contain more targets. In the single target case, this method breaks down to measuring the cell that is most likely to contain the target.

We compare the performance of the various managed strategies and the periodic scheme in Figure 7 by looking at RMS error versus number of sensor dwells. As before, multitarget

RMS error is computed by taking the average RMS error across all targets. In all cases, the filter is initialized with the true number and states of the targets.

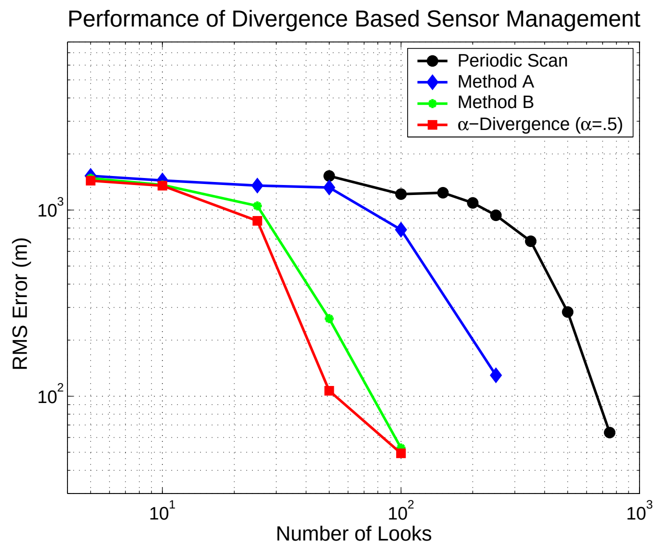

Fig. 7. A comparison of the performance of the various managed strategies and the periodic scheme in terms of RMS error versus number of looks. The α-divergence strategy out performs the other strategies, and at 50 looks performs similarly to non-managed with 750 looks.

Figure 7 shows that the non-managed scenario needs approximately 750 looks to equal the performance of the managed algorithm in terms of RMSE error. We say that the sensor manager is approximately 15 times as efficient as allocating the sensors without management. Furthermore, the additional sensor management schemes perform more poorly than the divergence driven method.

As mentioned in Section IV, this technique provides for an efficient algorithm computationally. Modestly optimized MatLab^{TM} code running on an off-the-shelf 2.8 GHz Linux machine tracks using the sensor management algorithm on three targets (section V-A) in about 10% longer than real-time. Tracking the targets using just a periodic scan in the 3 target scenario is done 3-4 times faster than real time. Tracking ten targets with sensor management is about 10 times slower than real time. This owes mainly to the fact that more sensor dwells are required to be scheduled and that there is more opportunity for targets to be coupled (closely spaced) which requires the CP algorithm rather than the IP algorithm (see the Appendix). The non-managed tracking algorithm runs in real time.

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.

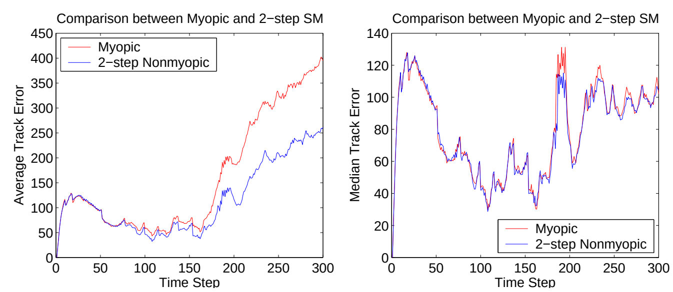

Fig. 8. A comparison of sensor management performance in the myopic (greedy) case and in the 2-step non-myopic case.

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. Furthermore, the algorithm outperforms simplistic sensor management strategies that are predicated on looking where the target is expected to be.

Appendix I
Adaptive Sampling for Particle Proposal

Estimating the entire joint density rather than a factorized approximation provides the advantage that the correlations between targets are modelled. However, the dramatic increase in dimensionality requires advanced sampling schemes to prevent undue computational burden. We detail herein the adaptive sampling scheme we utilize to provide computational tractability. A more thorough treatment of this topic is given in [20].

The standard method of particle proposal used in the literature, which we will refer to as sampling from the kinematic prior, proposes new particles $\mathbf{X}_p^k$ at time k using only the particles at time $k-1$, $\mathbf{X}_p^{k-1}$, and the model of target kinematics $p(\mathbf{X}^k|\mathbf{X}^{k-1})$. For each target in each particle at time $k-1$, the model of single target kinematics is used to propose a new state. This method has the benefit that it is simple to implement and is computationally inexpensive. However, one obvious drawback is that it does not take advantage of the fact that the state vector in fact represents many targets. Targets that are far apart in measurement space behave independently and should be treated as such. A second drawback, common to many particle filtering applications, is that the current measurements are not used when proposing new particles. These two considerations taken together result in a very inefficient use of particles and therefore require large numbers of particles to successfully track.

To overcome these deficiencies, we have employed an alternative particle proposal technique which biases the proposal process towards the measurements and allows for factorization of the target state when permissible. These techniques are collectively called multi-partition proposal (MPP) strategies. The MPP strategies propose each partition (target) in a particle separately, and form new particles as the combination of the proposed partitions. Particle weighting is then appropriately adjusted to account for this biased sampling.

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

1) Independent-Partition (IP) Method: The independent partition (IP) method given by Orton [28] is a convenient way to propose particles when part or all of the joint multitarget density factorizes. When applicable, we apply the Independent-Partition (IP) method of Orton to propose new partitions independently as follows. For a partition i, each particle at time $k-1$ has it's i^{th} partition proposed via the Kinematic prior and weighted by the measurements. From this set of N_{part} weighted estimates of the state of the i^{th} target, we select N_{part} samples with replacement to form the i^{th} partition of the particles at time k.

Recall that in our paradigm a partition corresponds to a target. See equation (7) for a concrete example of a particle and its partitions. A particle is then built by combining the individual partitions selected and reweighting correctly to compensate for the biased sampling.

It is important to carefully account for the permutation symmetry issue discussed in Section III here. The IP method of [28] is not permutation independent as it assumes that the i^{th} partition of each particle corresponds to the same target. Therefore the partitions in each particle must be identically positioned before this method is applied. If IP is applied to particles that have different orderings of partitions, multiple targets will be grouped together and erroneously used to propose the location of a single target.

In the case of well separated targets, this method allows many targets to be tracked with the same number of particles needed to track a single target. Indeed, as mentioned earlier, in the case of well separated targets, the multitarget tracking problem breaks down into many single-target problems. The IP method is useful for just this case, as it allows the targets to be treated independently when their relative spacing deems that appropriate. Note, however, that this method is not applicable when there is any measurement-to-target association ambiguity. Therefore, when targets are close together in sensor space, an alternative method must be used.

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

We apply the coupled partitions method as follows. To propose partition i of particle p, CP proposes M possible realizations of the future state using the kinematic prior. The M proposed futures are then given weights according to the current measurements and a single representative is selected. This process is repeated for each particle until the i^{th} partition for all particles has been formed. As in the IP method, the final particle weights must be adjusted for this biased sampling.

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

3) Adaptive Particle Proposal Method: Since at any particular time, some of the partitions are coupled while others are independent, we propose a hybrid sampling scheme, called the Adaptive-Partition (AP) method. The adaptive-partition method again considers each partition separately. Those partitions that are sufficiently well separated according to a given metric from all other partitions (see below) are treated as independent and proposed using the IP method. When targets are not sufficiently distant, the CP method is used. Therefore, the AP method is permutation independent, as it only uses IP when target partitions are already identically ordered.

To determine when targets are sufficiently separated, we have employed distance measures between the estimated centers of the i^{th} partition and the j^{th} partition (both a Euclidian metric and the Mahalanobis metric) as well as a simple nearest-neighbor type criterion. In practice, it is found that simply using a nearest neighbor criterion is sufficient and less computationally burdensome.

The AP method dramatically increases the efficiency of each particle, by automatically factorizing the state into a product of independent partitions and coupled partitions. In the extreme case where all targets are well separated the AP method operates like a set of single target filters. In the opposite extreme where all targets are coupled the AP method correctly models the correlation between targets.

References

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

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

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

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

[5] 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.

[6] J. Denzler, C. M. Brown, "Information Theoretic Sensor Data Selection for Active Object Recognition and State Estimatation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, February 2002, pp. 145-157.

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

[8] 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.

[9] 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.

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] M. Isard and J. MacCormick, "BraMBLe: A Bayesian Multiple-Blob Tracker", Proceedings of the 8th International Conference on Computer Vision, 2001.

[17] 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.

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

[19] 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.

[20] C. Kreucher, K. Kastella, A. O. and Hero III, "Tracking Multiple Targets Using a Particle Filter Representation of the Joint Multitarget Probability Density", under review at *IEEE Transactions on Aerospace and Electronic Systems*.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] M. I. Miller, A. Sruvastava and U. Grenander, "Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition", *IEEE Transactions on Signal Processing*, vol. 43, no. 11, pp. 2678-2690, November 1995.

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

[28] M. Orton and W. Fitzgerald, "A Bayesian Approach to Tracking Multiple Targets Using Sensor Arrays and Particle Filters", *IEEE Transactions on Signal Processing*, vol. 50, no. 2, pp. 216-223, 2002.

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

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

[31] W. Schmaedeke and K. Kastella, "Event-averaged maximum likelihood estimation and information-based sensor management", Signal Processing, Sensor Fusion, and Target Recognition III. Proceedings of the SPIE - The International Society for Optical Engineering, vol. 2232, Orlando, FL, June 1994, pp. 91-96.

[32] M. A. Sipe, D. Casasent, "Feature Space Trajectory Methods for Active Computer Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, December 2002.

[33] L. D. Stone, T. L. Corwin, C. A. Barlow, *Bayesian Multiple Target Tracking*, Artech House, 1999.

[34] H. Tao, H. S. Sawhney, and R. Kumar, "A Sampling Algorithm for Tracking Multiple Objects", Workshop on Vision Algorithms 1999, pp. 53-68.

[35] 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.

[36] 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.

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

[38] X. S. Zhou and T. S. Huang, "Relevance Feedback in Image Retrieval : A Comprehensive Review", *ACM Multimedia Journal*, vol. 8, no. 6, April 2003, pp. 536-544.