Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Signal Processing 85 (2005) 607–624

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Sensor management using an active sensing approach ☆

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

[a]Department of Electrical Engineering and Computer Science, The University of Michigan, 1301 Beal Avenue, Ann Arbor, MI 48109-2122, USA

[b]General Dynamics Advanced Information Systems, P.O. Box 134008, Ann Arbor, MI 48113-4008, USA

Received 30 July 2004; received in revised form 11 November 2004

Abstract

An approach that is common in the machine learning literature, known as active sensing, is applied to provide a method for managing agile sensors in a dynamic environment. We adopt an active sensing approach to scheduling sensors for multiple target tracking applications that combines particle filtering, predictive density estimation, and relative entropy maximization. Specifically, the goal of the system is to learn the number and states of a group of moving targets occupying a surveillance region. At each time step, the system computes a sensing action to take, based on an entropy measure called the Rényi divergence. After the measurement is made, 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. The algorithms developed here extend standard active sensing methodology to dynamically evolving objects and continuous state spaces of high dimension. It is shown using simulated measurements on real recorded target trajectories that this method of sensor management yields more than a ten fold gain in sensor efficiency when compared to periodic scanning.

Keywords: Sensor management; Machine learning; Active sensing; Multitarget tracking; Particle filtering; Joint multitarget probability density

1. 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

☆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.

*Corresponding author. General Dynamics Advanced Information Systems, P.O. Box 134008, Ann Arbor, MI 48113-4008, USA. Tel.: +1 734 994 1200 X2717; fax: +1 734 994 5095.

E-mail addresses: ckreuche@umich.edu (C. Kreucher), keith.kastella@gd-ais.com (K. Kastella), hero@eecs.umich.edu (A.O. Hero III).

0165-1684/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2004.11.004

management are often military in nature [36], but also include things such as wireless networking [28] and robot path planning [29]. There are many objectives that the sensor manager may be tuned to meet, e.g. minimization of track loss, maximization of probability of target detection, minimization of track error/covariance, and maximization of identification accuracy. Each of these different objectives taken alone may lead to a different sensor allocation strategy [36,38].

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 [26,27] uses a multi-arm bandit formulation involving hidden Markov models. In [27], an optimal algorithm is formulated to track multiple targets with an electronically scanned array 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 [26], the problem is reversed, and a single target is observed 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 [5,6] 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 [32] proposes using reinforcement learning as an approximate approach to dynamic programming. Very recently, Hernandez et al. [12] have used posterior Cramer-Rao bounds [41] to control the measurement sequence in a setting similar to that studied here.

Others have proposed using information measures a 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 expected to be 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 (KL) divergence. Schmaedeke and Kastella [42] use the KL divergence to determine optimal sensor-to-target tasking. Kastella [21,23] uses KL divergence to manage a sensor between tracking and identification mode in the multitarget scenario. Mahler [30,31] uses the KL divergence as a metric for optimal multisensor multitarget sensor allocation. Zhao [47] compares several approaches, including simple heuristics, entropy, and relative entropy (KL).

Information-based adaptivity measures such as mutual information (related to the KL divergence) and entropy reduction are a common learning metric that have been used in the machine learning literature in techniques with the names "active object recognition" [8], "active computer vision" [44], and "active sensing" [10]. These techniques are iterative procedures wherein the system has the ability to change sensor parameters to make the learning task easier. The ultimate goal is to learn something about the environment, e.g. the class of an object, the orientation of a robot's tool, robot location.

A specific example of the role of information theoretic measures in machine learning is the repeated interrogation of an object to determine the object class. Denzler et al. [8] study a situation in which a camera has many adjustable parameters, including focal length, pan and tilt angles,

and camera viewing position. They use mutual information to determine the action (set of camera parameters) that will maximally decrease uncertainty about the object class after the measurement is made. This set of parameters is used to acquire the next measurement. That work considers the case where the object comes from one of a finite number of classes, and the state of the object does not change throughout the experiment.

In the context of multitarget tracking, we use information theoretic methods to learn the number of targets present in the surveillance region as well as their states. Unlike object recognition applications, the number and states of the targets is a dynamic process which evolves over time. Therefore, we utilize a target tracking algorithm to recursively estimate the joint multitarget probability density (JMPD) for the set of targets under surveillance. Furthermore, the target states are continuous rather than coming from a collection of discrete possibilities. Due to this, we use a Monte Carlo method known as particle filtering to represent the JMPD.

In a manner analogous to active sensing for object recognition, at each iteration of our algorithm we use an information measure to decide on the optimal sensing action to make. 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 JMPD and the JMPD after a measurement has been made. In this work, we consider a quite general information measure called the Rényi information divergence [39] (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.

This paper contains two main contributions. First, we give a particle filter (PF) multitarget tracking algorithm that by design explicitly enforces the multitarget nature of the problem. Each particle is a sample from the JMPD and thus an estimate of the status of the system—the number of targets in the surveillance area as well as their individual states. Using a single particle to represent the states of multiple targets has been previously done in [17,37] and elsewhere. Our method builds on the independent partition method of [37]. 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 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 an active sensing 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 are expected to 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 more than a ten-fold increase in sensor efficiency over periodic scanning in scenarios considered.

The paper is organized as follows. In Section 2, we give the details of the JMPD target tracking algorithm and examine the numerical difficulties involved in directly implementing JMPD on a grid. In Section 3, we present our PF based implementation of JMPD. We see that this implementation provides computational tractability, allowing realistic scenarios to be considered. Our sensor management scheme, which uses the Rényi divergence as a metric, is extensively detailed in Section 4. A performance analysis of the tracker using sensor management on two model problems of

increasing realism is given in Section 5. 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 thoughts on future direction in Section 6.

2. The joint multitarget probability density

In this section, we give the details of a Bayesian method of multitarget tracking via recursive estimation of the JMPD. Others have studied Bayesian methods for multitarget tracking [2,33,45].

Mahler [11,31,35] advocates an approach to multitarget tracking based on random sets called "finite-set statistics" (FISST). Since FISST and the JMPD approach attack some of the same problems, many of the concepts that appear here, such as multitarget motion models and multitarget measurement models, also appear in the work of Mahler et al. [11]. FISST is a theoretical framework for unifying most techniques for reasoning under uncertainty (e.g. Dempster Shafer, fuzzy, Bayes, rules) in a common structure based on random sets. JMPD can be derived in the FISST framework and both strategies can both be traced back to the expectation averaged maximum likelihood estimator (EAMLE) work of Kastella [19–21,35]. The EAMLE work, of course, builds on early multitarget tracking work such as [18,34,43] and others. The JMPD technique does not require the random set formalism of FISST; in particular, in contrast to the random set approach the JMPD technique adopts the view that likelihoods and the JMPD are conventional Bayesian objects to be manipulated by the usual rules of probability and statistics. Therefore, the JMPD approach described here makes no appeal to random sets or related concepts such as Radon–Nikodym derivatives.

The concept of JMPD was discussed in [21,35], 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 Fig. 1. There are two target crossings, a challenging scenario for multitarget trackers.

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 $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k|\mathbf{Z}^k)$ as the probability density for exactly T targets with states $\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ at time k based on a set of observations $\mathbf{Z}^k$. The number of targets T is to be estimated simultaneously with the states of the T targets. The observation set $\mathbf{Z}^k$ is the collection of measurements up to and including time k, i.e. $\mathbf{Z}^k = \{\mathbf{z}^1, \mathbf{z}^2, \ldots, \mathbf{z}^k\}$, where each $\mathbf{z}^i$ 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 $p(\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k, T^k|\mathbf{Z}^k)$ 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 $\mathbf{x}_1^k, \mathbf{x}_2^k, \ldots, \mathbf{x}_{T-1}^k, \mathbf{x}_T^k$ as a partition of the multitarget state $\mathbf{X}$. For convenience, the density will be written more compactly in the traditional manner as $p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$, which implies that the state-vector

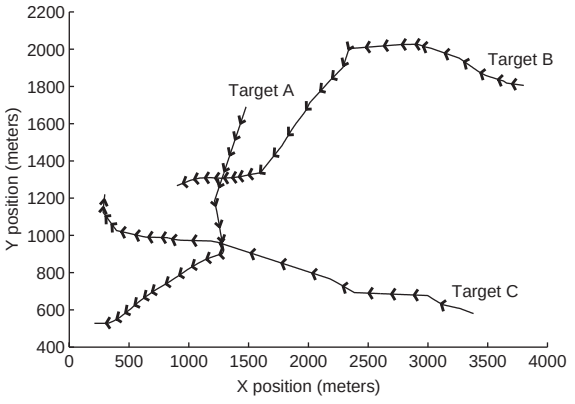

Fig. 1. A scenario with three moving targets. Target paths are indicated by lines, and direction of travel by arrows. There are two instances where target paths cross.

$\mathbf{X}$ represents a variable number of targets each possessing their own state vector. As an illustration, some examples illustrating the sample space of p are

$p(\emptyset, T=0\|\mathbf{Z})$,	posterior probability density for no targets in surveillance area,
$p(\mathbf{x}_1, T=1\|\mathbf{Z})$,	posterior probability density for 1 target with state $\mathbf{x}_1$,
$p(\mathbf{x}_1, \mathbf{x}_2, T=2\|\mathbf{Z})$,	posterior probability density for 2 targets with states $\mathbf{x}_1$ and $\mathbf{x}_2$,
$p(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, T=3\|\mathbf{Z})$,	posterior probability density for 3 targets with states $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$.

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. This issue arises in many other contexts, including Bayesian analysis of mixtures with an unknown number of components [40]. Therefore, all algorithms designed to implement the JMPD (and algorithms that implement the active sensing 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 A.

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

$$p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}) = \sum_{T^{k-1}=0}^{\infty} \int \mathrm{d}\mathbf{X}^{k-1} p(\mathbf{X}^k, T^k|\mathbf{X}^{k-1}, T^{k-1}) \times p(\mathbf{X}^{k-1}, T^{k-1}|\mathbf{Z}^{k-1}) \quad (1)$$

$p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})$ 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 $k-1$. This model describes how the state of the system $(\mathbf{X}, \mathbf{T})$ which includes both the kinematic states of the individual targets and the number of targets evolves with time. Therefore, the time evolution of the JMPD may be a collection of target kinematic models or may also 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, $p(\mathbf{z}^k|\mathbf{X}^k, T^k)$, 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

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

$p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$ is referred to as the posterior as it is the density at time k conditioned on all measurements up to and including time k. The sensor model $p(\mathbf{z}^k|\mathbf{X}^k, T^k)$ describes how measurements couple to the system state (both the number of targets and their individual states). 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 given by Kastella [22] 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 grow 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 yields more tractable computations. In the next section, we show that the Monte Carlo methods collectively known as particle filtering break this computational barrier.

3. The particle filter implementation of JMPD

A PF implementation of JMPD allows us to investigate more realistic problems. Other authors [17,46] have investigated using PF algorithms to approximate a multi-object density in the context of computer vision. The PF implementation we use is an extension of that given by Orton [37]. The similarities and differences are detailed in Appendix A. In particular, the algorithm 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. A more detailed description of our algorithm is given in [24,25].

To implement JMPD via a PF, we approximate $p(\mathbf{X}, T|\mathbf{Z})$ by a set of N_{part} weighted samples (particles). Let the multitarget state vector be written $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{T-1}, \mathbf{x}_T]$ and be defined for all T, $T = 0, \ldots, \infty$. Next, let the particle state vector be written $\mathbf{X}_p = [\mathbf{x}_{p,1}, \mathbf{x}_{p,2}, \ldots, \mathbf{x}_{p,T_p}]$ where T_p is the estimate particle p has for the number of targets in the region. Letting δ_{D} denote the 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

$$\delta(\mathbf{X} - \mathbf{X}_p) = \begin{cases} 0 & T \neq T_p, \\ \delta_{\text{D}}(\mathbf{X} - \mathbf{X}_p) & \text{otherwise.} \end{cases} \tag{3}$$

Then the PF approximation to the JMPD is given by

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

Different particles in the approximation may have different estimates target number, 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 [9]. 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.

This method differs from other PF algorithms where a single particle corresponds to a single target, as it explicitly enforces the multitarget nature of the problem by encoding in each particle an estimate of the number of targets and the states of those targets. The permutation symmetry discussed in Section 2 is inherited by the PF. Each particle contains many partitions (the estimate of target number) 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 density rather than merely a factorized version 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 tractability of the PF 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.

Estimating the multitarget states from the PF representation of JMPD is done in a way that is invariant to particle permutation. 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 Appendix A) are already correctly ordered and are not included in the clustering procedure. Second, since this ordering occurs at each time step, those partitions that are coupled are nearly ordered already, so one iteration of the K-means algorithm is enough to find the best permutation.

4. An active sensing approach to 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 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 action to take. 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 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 [14,39], also known as the α-divergence:

$$D_\alpha(f_1||f_0) = \frac{1}{\alpha-1}\ln\int f_1^\alpha(x) f_0^{1-\alpha}(x)\,\mathrm{d}x. \tag{5}$$

The adoption of the Rényi divergence as a sensor scheduling criterion can be motivated by universal hypothesis testing results of large deviation theory [4,7]. Specifically, consider the problem of testing between the hypotheses

$$H_0 : p(\mathbf{X}^k, T^k|\mathbf{Z}^k) = p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}),$$

$$H_1 : p(\mathbf{X}^k, T^k|\mathbf{Z}^k) \neq p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}) \tag{6}$$

based on an i.i.d. sample $\{\mathbf{X}_{(j)}^k\}_{j=1}^n$ from the posterior $p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$, e.g., as generated by the PF algorithm described above. H_1 is the hypothesis that the new measurement has changed the target state density, while H_0 is the hypothesis that the new measurement has not changed the target state density. The performance of any test of H_0 versus H_1 is specified by its receiver operating characteristic (α_n, β_n) where we have defined the false alarm probability $\alpha_n = p(\text{decide } H_1|H_0)$ and the miss probability $\beta_n = p(\text{decide } H_0|H_1)$. If the true posterior distribution under H_1 were known, the Neyman–Pearson likelihood ratio test, parameterized by a decision threshold γ, is optimal in the sense of achieving minimum β_n for any specified level α_n (determined by γ). Under broad assumptions, for large n the error rates of the optimal test satisfy Theorem 3.4.3 in [7]:

$$\frac{1}{n}\log\alpha_n \approx -\sup_{\alpha\in[0,1]}\{\alpha\gamma - (1-\alpha)D_\alpha(p_1\|p_0)\}, \tag{7}$$

$$\frac{1}{n}\log\beta_n \approx -\sup_{\alpha\in[0,1]}\{-\alpha\gamma - (1-\alpha)D_\alpha(p_0\|p_1)\}. \tag{8}$$

where p_0 and p_1 denote the posterior $p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$ under H_0 and H_1, respectively.

In particular, if one selects $\gamma = 0$ then the two error rate exponents (right sides of Eqs. (7) and (8)) are identically $-(1-\alpha^*)D_{\alpha^*}(p_1\|p_0)$ for some $\alpha^* \in [0,1]$. Thus the Rényi α-divergence specifies the error exponents of the Neyman–Pearson optimal test.

For the composite hypotheses (Eq. (6)) the generalized likelihood ratio test (GLRT) of H_0 versus H_1 is asymptotically (large n) optimal and can be implemented by thresholding an empirical estimate of the error rate exponent (Eq. (7)) to achieve a specified level of false alarm α_n (Theorem 7.1.3 in [7]). This lends strong theoretical justification for using the Rényi divergence sensor selection criterion proposed in this paper.

Returning to Eq. (5), we note that the α parameter 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 commonly utilized (KL) discrimination (9).

$$\lim_{\alpha\to 1} D_\alpha(f_1||f_0) = \int f_0(x) \ln \frac{f_0(x)}{f_1(x)}\,\mathrm{d}x. \tag{9}$$

In the case that $\alpha = 0.5$, the Rényi information divergence is related to the Hellinger–Battacharya distance squared [13]

$$d_H(f_1,f_0) = \frac{1}{2}\int \left(\sqrt{f_1(x)} - \sqrt{f_0(x)}\right)^2 \mathrm{d}x. \tag{10}$$

The function D_α in Eq. (5) 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 $p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})$ and the updated density after a measurement is made, $p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$. Therefore, we write

$$\begin{aligned} &D_\alpha(p(\cdot|\mathbf{Z}^k)||p(\cdot|\mathbf{Z}^{k-1})) \\ &\quad = \frac{1}{\alpha-1}\ln \sum_{T^k=0}^{\infty} \int \mathrm{d}\mathbf{X}^k p(\mathbf{X}^k, T^k|\mathbf{Z}^k)^\alpha \\ &\qquad \times p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})^{1-\alpha}, \end{aligned} \tag{11}$$

which implies that the JMPD must be integrated over all possible (discrete) target numbers T and all possible configurations of T targets.

After algebra and incorporation of Bayes' rule, Eq. (11) can be simplified to

$$\begin{aligned} &D_\alpha(p(\cdot|\mathbf{Z}^k)||p(\cdot|\mathbf{Z}^{k-1})) \\ &\quad = \frac{1}{\alpha-1}\ln \frac{1}{p(\mathbf{z}|\mathbf{Z}^{k-1})^\alpha} \sum p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1}) \\ &\qquad \times p(\mathbf{z}|\mathbf{X}^k, T^k)^\alpha. \end{aligned} \tag{12}$$

The integral reduces to a summation since any discrete approximation of $p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})$ only has non-zero probability at a finite number of target states. In the PF 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.

The particle approximation of the density (Eq. (4)) reduces Eq. (12) to

$$\begin{aligned} &D_\alpha(p(\cdot|\mathbf{Z}^k)||p(\cdot|\mathbf{Z}^{k-1})) \\ &\quad = \frac{1}{\alpha-1}\ln \frac{1}{p(\mathbf{z})^\alpha} \sum_{p=1}^{N_{\text{part}}} w_p p(\mathbf{z}|\mathbf{X}_\mathbf{p})^\alpha, \end{aligned} \tag{13}$$

where $p(\mathbf{z}) = \sum_{p=1}^{N_{\text{part}}} w_p p(\mathbf{z}|\mathbf{X}_p)$.

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

We propose as a method of sensor management calculating the expected value of Eq. (13) for each of the M possible sensing actions and choosing the action that maximizes the expectation. In this notation m $(m = 1,\ldots,M)$ will refer to a 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 Eq. (13) may be written as an integral over all possible outcomes $\mathbf{z}_m$ when

performing sensing action m:

$$\langle D_\alpha \rangle_m = \int \mathrm{d}\mathbf{z} p(\mathbf{z}|\mathbf{Z}^{k-1}, m) D_\alpha(p(\cdot|\mathbf{Z}^k)||p(\cdot|\mathbf{Z}^{k-1})). \tag{14}$$

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

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

which, using Eq. (13) results in

$$\langle D_\alpha \rangle_m = \frac{1}{\alpha - 1} \sum_{z=0}^{1} p(z) \ln \frac{1}{p(z)^\alpha} \sum_{p=1}^{N_{\text{part}}} w_p p(z|\mathbf{X}_p)^\alpha. \tag{16}$$

Computationally, Eq. (16) is calculated for M possible sensing actions in $O(MN_{\text{part}})$. Notice the sensor management algorithm is permutation invariant as it only depends on the likelihood of the measurements given the particles.

We have specialized to the case where measurements are thresholded, but make the following comments about 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. When z is continuous the integral of Eq. (14) is in principle very challenging to evaluate. However, we have found that quantizing z using an algorithm such as LGB and then calculating the sensor management actions assuming a discrete set of possible outcomes results in very little degradation in tracking performance.

In summary, our sensor management algorithm proceeds as follows. At each occasion where a sensing action is to be made, we evaluate the expected information gain as given by Eq. (16) 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. The complete particle filtering and sensor management algorithm is outlined in Table 1.

Table 1
The sensor management algorithm

(1) Generate particles representing $p(\mathbf{X}^k, T^k|\mathbf{Z}^{k-1})$ by proposing particles representing $p(\mathbf{X}^{k-1}, T^{k-1}|\mathbf{Z}^{k-1})$ forward via the kinematic prior.
(2) Compute the expected gain in information for each possible sensing action (i.e. evaluate Eq. (17) using the particles generated in step (1) for all m).
(3) Use particles representing $p(\mathbf{X}^{k-1}, T^{k-1}|\mathbf{Z}^{k-1})$ and $\mathbf{z}^k$ to propose a new set of particles representing $p(\mathbf{X}^k, T^k|\mathbf{Z}^k)$. (The method of sampling is briefly described in the Appendix—see [24,25] for complete details)
  (a) Segment targets into coupled and independent partitions
  (b) Propose all partitions forward using the measurements
  (c) Weight particles appropriately (via Bayes' rule)
(4) set $k \leftarrow k+1$, and go to step (1)

4.1. On the value of α in the Rényi divergence

The Rényi divergence has been used in many diverse applications, including content-based image retrieval, image georegistration, and target detection [13,14]. These studies provide guidance as to the optimal choice of α.

In the georegistration problem [14] 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 than the KL ($\alpha = 1$). 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 5.

An asymptotic analysis [14] shows that $\alpha = 0.5$ results in maximum discriminatory ability between two very similar densities. The value $\alpha = 0.5$

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.

4.2. Extensions to non-myopic sensor management

The sensor management algorithm proposed here is myopic as it does not take into account long-term ramifications when deciding the optimal action. In some scenarios, the greedy approach may be close to optimal. However, in scenarios where the problem dynamics 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 before the target disappears. This will reduce 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 described by [1].

At each time a decision is to be made, we enumerate all possible measurements and 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, and compute the actual information gain received under this simulated measurement. We 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 (two-step) information gain for making the particular measurement. Running this procedure many times gives a Monte Carlo estimate of ramification of making a particular measurement. N-step extensions are straightforward, but computationally burdensome.

5. 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.

5.1. An extensive evaluation of sensor management performance using three simulated targets

We first illustrate 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 nearly 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 $P_f =$

$P_d^{(1+\mathrm{SNR})}$, which is a standard model for thresholded detection of Rayleigh returns [3]. When there are T targets in the same cell, the detection probability increases according to $P_d(T) = P_d^{(1+\mathrm{SNR})/(1+T*\mathrm{SNR})}$. This model is known by the filter and used to evaluate $p(z|\mathbf{X}, T)$ in Eq. (2). Specifically, the likelihood of a measurement z in cell c under hypothesis $\mathbf{X}$ is evaluated by first determining the number of targets T that $\mathbf{X}$ predicts are in cell c, and then using

$$p(z|\mathbf{X}) = \begin{cases} P_d(T) & z = 1, \\ 1 - P_d(T) & z = 0. \end{cases}$$

The filter uses 500 particles and 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 divergence based scheme presented in Section 4. The periodic scheme measures each cell in sequence. At time 1, cells $1, \ldots, L$ are measured. At time 2, cells $L+1, \ldots, 2L$ 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 earlier measurement outcomes).

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

Qualitatively, managed measurements are focused in or near the cells the targets are in. Furthermore, covariance ellipses, which reflect the current state of knowledge of the tracker conditioned on all previous measurements, are 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 Fig. 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

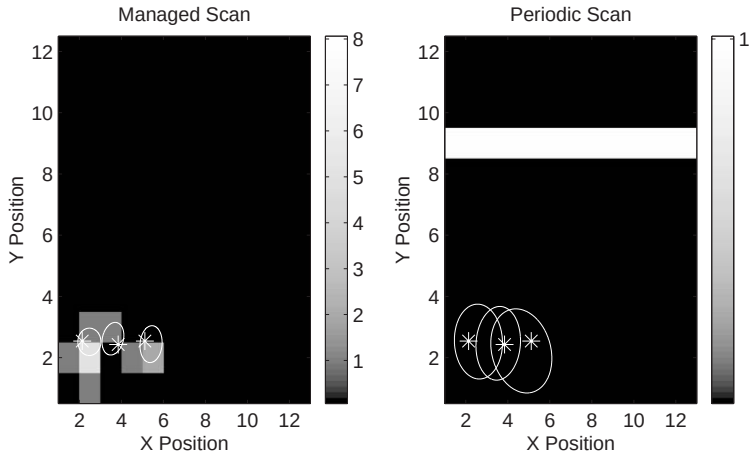

Fig. 2. Comparison of managed and non-managed tracking. (L) Using sensor management, and (R) A periodic scheme. Targets are marked with an asterisk, the covariance of the filter estimate is given by the ellipse, and grey scale is used to indicate the number of times each cell has been measured at this 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 are indicated by the white stripe). With sensor management, measurements are used only in areas that contain targets.

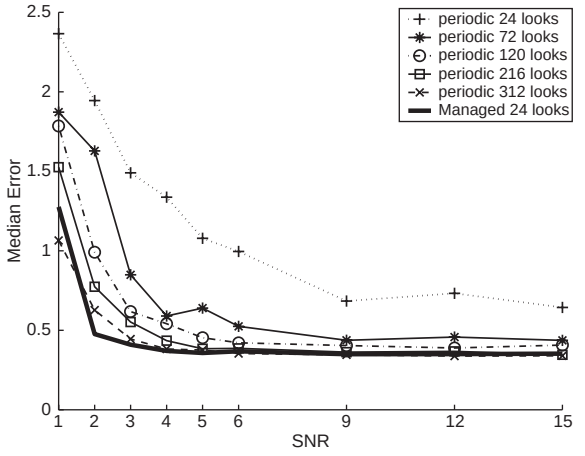

Fig. 3. The median error versus signal to noise ratio (SNR). Managed performance with 24 looks is similar to non-managed with 312 looks.

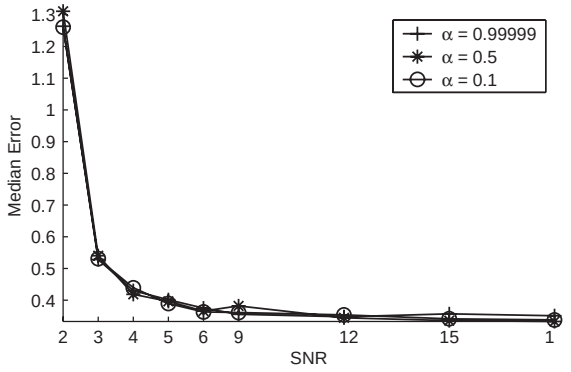

Fig. 4. The performance of the sensor management algorithm with different values of α. We find that in the case where the filter dynamics match the actual target dynamics, the algorithm is insensitive to the choice of α.

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 Eq. (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.

To determine the sensitivity of the sensor management algorithm to the choice of α, we test the performance with $\alpha = 0.1$, $\alpha = 0.5$, and $\alpha \approx 1$. Fig. 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.

5.2. A comparison using ten real targets

We test the sensor management algorithm again using a modified version of the above simulation, which is intended to demonstrate the technique in a scenario of increased realism. Here we have ten targets moving in a 5000 m × 5000 m 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. cross). 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 s intervals, and the simulation duration is 1000 time steps.

The filter again assumes nearly 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 (i.e. the filter knows there are ten targets in the region).

At each time step, an imager is able to measure cells in the surveillance area by making measurements on a grid with 100 m × 100 m detection cell resolution. The sensor emulates a moving target indicator (MTI) sensor. A real MTI sensor measures a long narrow beam in cross range consisting of many returns in the range direction. Our model emulates this in that the sensor lays 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 either a 0 (no detection) or a 1 (detection) governed by P_d, P_f, and SNR. In this illustration, we take $P_d = 0.5$, $\mathrm{SNR} = 2$ (3 dB), and $P_f = P_d^{(1+\mathrm{SNR})}$. When there are T targets in the same cell, the detection probability increases according to $P_d(T) = P_d^{(1+\mathrm{SNR})/(1+T*\mathrm{SNR})}$.

We compare first the performance of the sensor management algorithm under different values of α in Eq. (5). This problem is more challenging then the simulation of Section 5.1 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 4.1) leads us to believe that $\alpha = 0.5$ is the right choice in this scenario.

In Fig. 5, we show the results of 50 Monte Carlo trials using our sensor management technique with $\alpha = 0.1$, $\alpha = 0.5$, and $\alpha = 0.99999$. The statistics are summarized in Table 2. 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 m after some point in time. Both the $\alpha \approx 1$ and $\alpha = 0.1$ case lose track of targets on several occasions.

Due to the asymptotic analysis and these empirical results, we employ $\alpha = 0.5$ 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.

Management algorithm "A" manages the sensor by pointing it at or near the estimated location of

Table 2
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

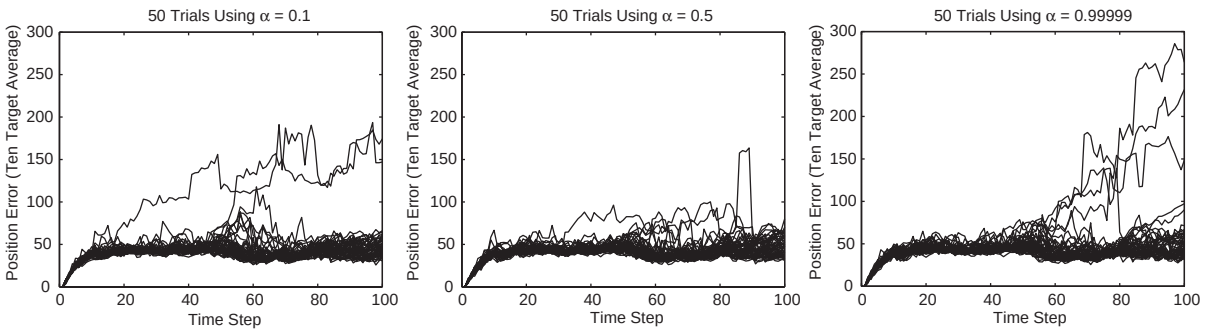

Fig. 5. A comparison of sensor management performance under different values of the Rényi divergence parameter, α.

the targets. Specifically, algorithm "A" performs a gating to restrict the portion of the surveillance area that the sensor will consider measuring. The PF approximation of the time updated JMPD (Eq. (1)) is used to predict the location of each target at the current time. The set of cells considered 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.

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 Fig. 6 by looking at RMS error versus number of sensor dwells ("looks"). 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.

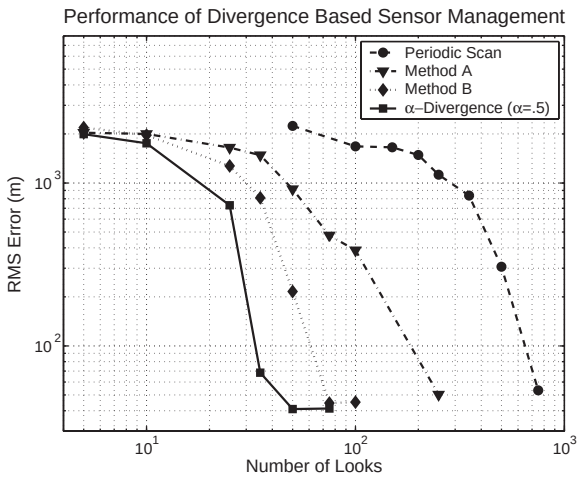

Fig. 6. A comparison of the performance of the various managed strategies and the periodic scheme in terms of RMS error versus number of looks. The α-divergence strategy out performs the other strategies, and at 35 looks performs similarly to non-managed with 750 looks.

Fig. 6 shows that the non-managed scenario at 750 looks is approximately the same as the managed algorithm at 35 looks in terms of RMSE error. We say that the sensor manager is approximately 20 times as efficient as allocating the sensors without management. Furthermore, the additional sensor management schemes are outperformed by the divergence driven method.

As mentioned in Section 4, this technique provides for an efficient algorithm computationally. A combination of modestly optimized MatLab^{TM} code and C-code running on an off-the-shelf 3 GHz Linux machine tracks ten targets using the sensor management algorithm in about 10% longer than real-time.

5.3. The effect of non-myopic scheduling

Finally, we give preliminary results on the ramifications of non-myopic 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 Section 4.2 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

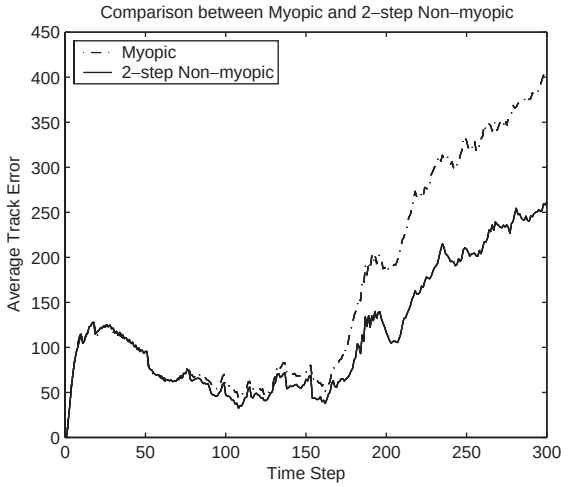

Fig. 7. A comparison of sensor management performance in the myopic (greedy) case and in the 2-step non-myopic case.

measure targets that are about to become obscured since this will result in the maximum total (2-step) information gain. We show in Fig. 7 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.

6. Discussion

We have applied an approach common in the machine learning literature, known as active sensing, 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 $p(\mathbf{X}, T|\mathbf{Z})$ approximated by the multitarget tracker via particle filtering. The posterior is used in conjunction with target kinematic 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 at the targets expected location.

The future direction for this work includes expanding the information based sensor scheduling to long term planning. We are currently investigating approximation techniques and learning methods. In addition, a second possible future direction is decentralized estimation and decision making. In many realistic scenarios, sensor management must be accomplished over a network of collaborating platforms with limited communication abilities.

Appendix A. Adaptive sampling for particle proposal

Estimating the entire joint density rather than a factorized approximation provides the advantage that 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 [24,25].

The standard method of particle proposal used in the literature, referred to as sampling from the kinematic prior, proposes new particles at time k using only the particles at time $k-1$ and the model of target kinematics. This method has the benefit that it is simple to implement and is computationally inexpensive. However, it neither makes use of the fact that the state vector in fact represents many targets nor the current measurements. 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 strategies propose each 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 toward 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.

A.1. Independent-partition (IP) method

The independent partition (IP) method given by Orton [37] is a convenient way to propose particles when part or all of the joint multitarget density factorizes. When applicable, we apply the IP method of Orton to propose new partitions independently as follows. For a partition i, each particle at time $k-1$ has it's ith partition proposed via the kinematic prior and weighted by the measurements. From this set of N_{part} weighted estimates of the state of the ith target, we select N_{part} samples with replacement to form the ith partition of the particles at time k. 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.

A.2. Coupled partition (CP) proposal method

When targets are close together in sensor space, the joint multitarget density does not completely factorize. We say the corresponding partitions are coupled, and the IP method is no longer applicable to those partitions. In these cases we use instead a method called coupled partitions (CP).

We apply the CP 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 ith 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.

A.3. Adaptive particle proposal method

At any particular time, some of partitions are coupled while others are independent. We use a hybrid scheme, called the adaptive partitions (AP). AP again considers each partition separately. Those partitions sufficiently separated from all other partitions are treated as independent and proposed using IP. When targets are not sufficiently distant, the CP method is used. In practice, it has been found that using a Euclidian distance criterion is sufficient to determine when targets are well separated. Therefore, the AP method is permutation independent, as it only uses IP when target partitions are already identically ordered.

The AP method dramatically increases the efficiency of each particle, by automatically factorizing the state into a product of independent and coupled partitions. In the extreme case where all targets are well separated AP 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, D. Castanon, Rollout algorithms for stochastic scheduling problems, J. Heuristics 5 (1) (1999) 89–108.
[2] R.E. Bethel, G.J. Paras, A PDF multisensor multitarget tracker, IEEE Trans. Aerospace Electron. Systems 34 (1998) 153–168.
[3] S. Blackman, Multiple-Target Tracking with Radar Applications, Artech House, Dedham, MA, 1986.
[4] R. Blahut, Principles and Practice of Information Theory, Addison-Wesley, Reading, MA, 1987.
[5] D. Castanon, Optimal search strategies for dynamic hypothesis testing, IEEE Trans. Systems Man Cybernet. 25 (1995) 1130–1138.
[6] D. Castanon, Approximate dynamic programming for sensor management, Proceedings of the 1997 Conference on Decision and Control, 1997.
[7] A. Dembo, O. Zeitouni, Large Deviations Techniques and Applications, Springer, NY, 1998.
[8] J. Denzler, C.M. Brown, Information theoretic sensor data selection for active object recognition and state estimation,

IEEE Trans. Pattern Anal. Machine Intell. 24 (2) (2002) 145–157.
[9] A. Doucet, N. de Freitas, N. Gordon, Sequential Monte Carlo Methods in Practice, Springer Publishing, New York, 2001.
[10] D. Fox, W. Burgard, S. Thrun, Active Markov localization for mobile robots, Robotics Auton. Systems 25 (1998) 195–207.
[11] I.R. Goodman, R.P.S. Mahler, H.T. Nguyen, Mathematics of Data Fusion, Kluwer Academic Publishers, Dordrecht, 1997.
[12] M.L. Hernandez, T. Kirubarajan, Y. Bar-Shalom, Multisensor resource deployment using posterior Cramer-Rao bounds, IEEE Trans. Aerospace Electron. Engng. 40 (2) (2004) 399–416.
[13] A.O. Hero, B. Ma, O. Michel, J.D. Gorman, Alpha divergence for classification, indexing and retrieval, Technical Report 328, Communications and Signal Processing Laboratory (CSPL), Department of EECS, University of Michigan, Ann Arbor, May, 2001.
[14] A.O. Hero, B. Ma, O. Michel, J. Gorman, Applications of entropic spanning graphs, IEEE Signal Process. Mag. (Special Issue on Math. Imaging) 19 (5) (2002) 85–95.
[15] K.J. Hintz, A measure of the information gain attributable to cueing, IEEE Trans. Systems Man Cybernet. 21 (1991) 237–244.
[16] K.J. Hintz, E.S. McVey, Multi-process constrained estimation, IEEE Trans. Man Systems Cybernet. 21 (1991) 434–442.
[17] M. Isard, J. MacCormick, BraMBLe: a Bayesian multiple-blob tracker, Proceedings of the Eighth International Conference on Computer Vision, 2001, pp. 34–41.
[18] E.W. Kamen, Multiple target tracking based on symmetric measurement functions, IEEE Trans. Automatic Control 37 (1992) 371–374.
[19] K. Kastella, A maximum likelihood estimator for report-to-track association, SPIE Proc. 1954 (1993) 386–393.
[20] K. Kastella, Event averaged maximum likelihood estimation and mean-field theory in multitarget tracking, IEEE Trans. Automatic Control 50 (6) (1995) 1070–1073.
[21] K. Kastella, Discrimination gain for sensor management in multitarget detection and tracking, IEEE-SMC and IMACS Multiconference CESA '96, vol. 1, Lille France, 9–12 July 1996, pp. 167–172.
[22] K. Kastella, Joint multitarget probabilities for detection and tracking, SPIE Proceedings of the Acquisition, Tracking and Pointing XI, Orlando, FL, April 1997.
[23] K. Kastella, Discrimination gain to optimize classification, IEEE Trans. Systems Man Cybernet. Systems and Humans 27 (1) (1997) 112–116.
[24] C. Kreucher, K. Kastella, A.O. Hero III, Tracking multiple targets using a particle filter representation of the joint multitarget probability density, IEEE Trans. Aerospace Electron. Systems, under review.
[25] C. Kreucher, K. Kastella, A.O. Hero III, Tracking multiple targets using a particle filter representation of the joint multitarget probability density, SPIE International Symposium on Optical Science and Technology, San Diego, California, August 2003.
[26] V. Krishnamurthy, Algorithms for optimal scheduling and management of hidden Markov model sensors, IEEE Trans. Signal Process. 50 (6) (2002) 1382–1397.
[27] V. Krishnamurthy, D. Evans, Hidden Markov model multiarm bandits: a methodology for beam scheduling in multitarget tracking, IEEE Trans. Signal Process. 49 (12) (2001) 2893–2908.
[28] J. Liu, P. Cheung, L. Guibas, 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.
[29] V.J. Lumelsky, S. Mukhopadhyay, K. Sun, Dynamic path planning in sensor-based terrain acquisition, IEEE Trans. Robotics Automation 6 (4) (1990) 462–472.
[30] R. Mahler, Global optimal sensor allocation, Proceedings of the Nineth National Symposium on Sensor Fusion, vol. I, 12–14 March, 1996, Naval Postgraduate School, Monterey CA, pp. 347–366.
[31] R. Mahler, A unified foundation for data fusion, in: Selected Papers on Sensor and Data Fusion, vol. MS-124, SPIE, 1996, pp. 325–345.
[32] R. Malhotra, Temporal considerations in sensor management, Proceedings of the IEEE 1995 National Aerospace and Electronics Conference, NAECON, vol. 1, Dayton, OH, 22–26 May 1995, pp. 86–93.
[33] M.I. Miller, A. Sruvastava, U. Grenander, Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition, IEEE Trans. Signal Process. 43 (11) (1995) 2678–2690.
[34] S. Mori, C.Y. Shong, E. Tse, R.P. Wishner, Tracking and classifying multiple targets without a prior identification, IEEE Trans. Automatic Control AC31 (May 1986).
[35] S. Musick, K. Kastella, R. Mahler, A practical implementation of joint multitarget probabilities, in: I. Kadar (Ed.), Signal Processing, Sensor Fusion, and Target Recognition VII, vol. 3374, SPIE, 1998, pp. 26–37.
[36] S. Musick, R. Malhotra, Chasing the elusive sensor manager, Proceedings of NAECON, Dayton, OH, May 1994, pp. 606–613.
[37] M. Orton, W. Fitzgerald, A Bayesian approach to tracking multiple targets using sensor arrays and particle filters, IEEE Trans. Signal Process. 50 (2) (2002) 216–223.
[38] R. Popoli, The sensor management imperative, in: Y. Bar-Shalom (Ed.), Multitarget-Multisensor Tracking: Advanced Applications, vol. II, Artech House, Boston, MA, 1992.
[39] A. Rényi, On measures of entropy and information, Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, 1961, pp. 547–561.
[40] S. Richardson, P.J. Green, On bayesian analysis of mixtures with an unknown number of components, J. Roy. Statist. Soc. B 59 (1997) 731–792.

[41] B. Ristic, A. Farina, M. Hernandez, Cramer-Rao lower bound for tracking multiple targets, Proceeding of the IEE, Radar, Sonar and Navigation, vol. 151 (3), 2004, pp. 129–134.
[42] W. Schmaedeke, K. Kastella, Event-averaged maximum likelihood estimation and information-based sensor management, Proceedings of SPIE, vol. 2232, Orlando, FL, 1994, pp. 91–96.
[43] H.M. Shertukde, Y. Bar-Shalom, Tracking of crossing targets with imaging sensors, IEEE Trans. Aeropace Electronic Systems 27 (1991) 582–592.
[44] M.A. Sipe, D. Casasent, Feature space trajectory methods for active computer vision, IEEE Trans. Pattern Anal. Mach. Intell. 24 (12) (2002) 1634–1643.
[45] L.D. Stone, T.L. Corwin, C.A. Barlow, Bayesian Multiple Target Tracking, Artech House, 1999.
[46] H. Tao, H.S. Sawhney, R. Kumar, A sampling algorithm for tracking multiple objects, Workshop on Vision Algorithms 1999, pp. 53–68.
[47] F. Zhao, J. Shin, J. Reich, Information-driven dynamic sensor collaboration, IEEE Signal Process. Mag. (March 2002) 61–72.