Level I and Level II Target Valuations for Sensor Management

K. C. CHANG
George Mason University

JOE P. HILL
SRS Technologies

Advanced optimization-based algorithms for sensor resource management have been the research focus area in multisensor tracking and fusion in the last decade. These algorithms for the most part offer the potential for automating the sensor control process in response to level 1 sensor data fusion (object or track-level) estimates. However, previous studies have indicated that these types of sensor resource management algorithms may have limited value in certain operational scenarios involving multi-platform surveillance and strike missions because the response is optimized for track maintenance without any assessment of overall situation context. In this paper, we will develop a framework for representing the expected information value of planned sensor measurements as it contributes to higher-level situation inferences. Specifically, a hierarchical target valuation model that estimates target value on the basis of not only a level 1 valuation function but also on the basis of a level 2 valuation function will be presented. These algorithms will provide for improved tracking and classification performance when identifying higher-level units such as convoys of vehicles. The valuation models rely on a computationally efficient implementation of Bayesian modeling and inference algorithms. Note that the main focus of the paper is on developing a hierarchical cost function that captures both level 1 and level 2 objectives and is not on developing sophisticated techniques for optimizing this objective. Simulation results which validate the approach are also presented.

Manuscript submitted on January 7, 2005; revised November 18, 2005 and September 25, 2006; released for publication February 17, 2007.

Refereeing of this contribution was handled by Dr. Chee-Yee Chong.

Authors' addresses: K. C. Chang, SEOR Department, George Mason University, 4400 University Drive, Fairfax, VA 22030, E-mail: (kchang@gmu.edu); J. P. Hill, SRS Technologies, 15000 Conference Center Drive, Suite 510, Chantilly, VA 220151, E-mail: (joe.hill@wg.srs.com).

1557-6418/06/$17.00 © 2006 JAIF

1. INTRODUCTION

Current military Intelligence Surveillance and Reconnaissance (ISR) systems employ agile, multi-mode sensors, which are capable of producing variable scan patterns within a surveillance region in response to external tasking [18]. A number of platforms are currently available to carry out these missions. For example, long-range surveillance is accomplished with aircraft capable of high coverage rate, high signal to noise moving target indicator (MTI) and high range resolution (HRR) modes (see Fig. 1). Additionally, these systems can perform long dwell synthetic aperture radar scans for purposes of identification. A primary example of such a sensor is a multi-mode, electronically scanned antenna radar capable of tasking individual beams in terms of pointing direction, dwell time, and waveform. As illustrated in Fig. 1, an enhanced radar is capable of not only interleaving various radar beam modes (i.e., wide area search (WAS), sector search (SS), high range resolution (HRR), and synthetic aperture radar (SAR)), but will also be capable of scanning the surveillance region in an asynchronous fashion as the timeline suggests (e.g., irregular revisits could be due to sensor tasking to maintain tracks, search new areas, identify high-value targets, etc.). For such systems, the dynamic management of sensor mode control requires an automated process due to the variable timeline for adaptation.

Exploitation of sensor data from multi-mode sensors is capable of producing tactically significant information that can contribute to battlefield situation awareness. The multi-mode sensor data products contribute attributes of target detection, location, and classification together with environmental characteristics related to clutter. These attributes provide evidence necessary to produce a fused situation estimate. The challenge of sensor resource management for such systems is to characterize the exploitation and data production process according to a consistent model that provides for real-time adaptive sensor management.

In order to support the solution of the sensor management problem, several different solution approaches have been previously developed. They include information theoretic approaches [14], random set approaches [12], and the methods based on stochastic dynamic programming (SDP) [2–5]. The SDP algorithms were developed to address the problem of determining the optimal time sequencing of the radar's SAR (for detecting stationary objects) and MTI modes (for tracking moving objects) that maximizes the total *information value*. A *value function* is basically a function of low level tracking and classification quality states. The scheduler operates in a feedback manner in real time; for example, as objects are detected by the SAR, they may be eliminated from consideration by the scheduler so that the remaining radar resources can be better focused on only those objects remaining undetected or needing track improvement.

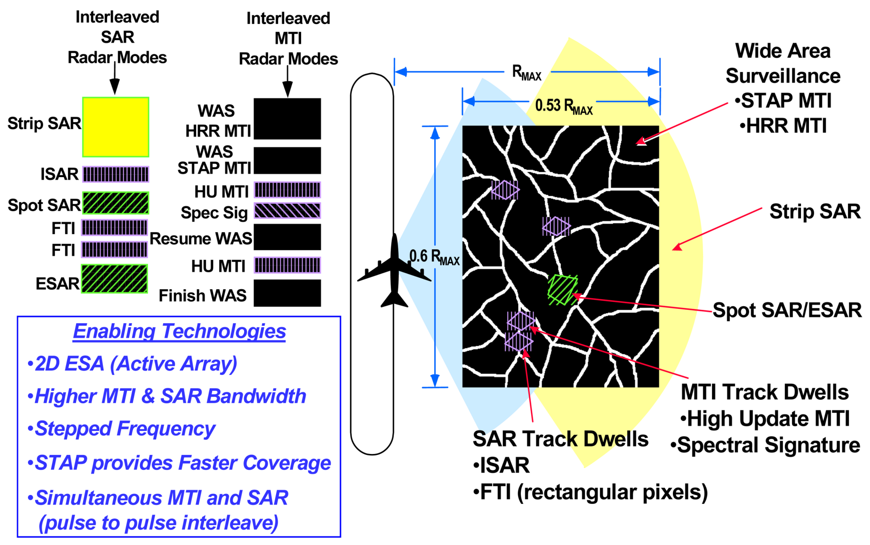

Fig. 1. Advanced airborne surveillance radars will include capabilities to operate in multiple modes and interleave modes.

The output of the scheduler is to determine for each instant of time whether the radar should

- operate in the SAR mode to image a single cell (for a stopped target), or
- operate in the MTI mode to detect and update tracks on all moving objects, or
- operate in the HRR mode to image a single cell (for a moving target) in order to obtain identification information (this is indeed the *only* way to obtain classification information), or,
- not be used at all in order to meet some exposure constraint.

The MTI mode typically requires the shortest dwell times (and thus consumes the least amount of radar resources); however, it has low range resolution and typically a low signal to noise ratio. The MTI mode is capable of detecting targets moving faster than the so-called minimum detectable velocity (MDV) of the sensor. It is well suited for problems such as tracking moving vehicles, characterizing traffic flow, and lines of communication using low-complexity (highest throughput) algorithms. The HRR mode has slightly longer dwell times (still shorter than SAR) but offers higher range resolution and strong signal to noise performance again for characterizing targets whose velocities exceed the MDV of the sensor. It has been proven to be useful for extracting coarse features (e.g. length and width) and as a tool for low-confidence classification. The HRR mode is eminently well suited for track maintenance problems. Finally, the SAR mode is ideal for two dimensional, high confidence classifications of stationary targets as well as change detection.

In the context of the Joint Director of Laboratories (JDL) terminology [17], it is observed that the track fusion models previously considered only address level 1 fusion (Object Assessment). Previous studies also indicated that sensor resource management algorithms utilizing only level 1 information may have limited value in certain operational scenarios involving multi-platform surveillance and strike missions because the response is optimized for track maintenance without any assessment of overall situation context [1, 13]. We contend that the problem of effective SRM for agile, multi-mode sensors will require improved representations of the process exploitation through level 2 information to adequately address the benefits of agile sensor tasking.

In this paper, we present a description of an algorithm that could be used to provide hierarchical target valuation based on not only level 1 (e.g., object or track) information, but also level 2 (e.g., groups of objects) information. This algorithm has the potential to improve the target valuation function used in a sensor resource manager by adding a valuation component related to the ability to identify a group of objects, such as convoys. This algorithm builds upon earlier results [10] that only addressed target valuation based on level 1 fusion information. The valuation algorithm is based on a Bayesian approach where a recursive composition inference algorithm was used to compute the hierarchical value function [10]. We have developed an efficient approximation algorithm to solve the combinatorial problem present in the original approach [7–8]. We have also developed an evaluation environment to analyze the performance of this valuation algorithm given a set of ground moving targets. The preliminary simulation results demonstrate the validity of our approach.

Note that the focus of this paper is on developing the hierarchical valuation function, not on deriving a sophisticated optimization algorithm. Essentially, any appropriate optimization algorithm can be applied to obtain the optimal solution. Although the stochastic dynamic programming approach mentioned before [3–5] seems to be a very suitable one. The remainder of the paper is organized as follows. Section 2 introduces and formulates the problem. Section 3 presents a complete description of the valuation function and solution procedure. Section 4 describes the evaluation environment and the test scenarios followed by a set of simulation results given in Section 5 to demonstrate the new approach. Finally, our contribution and future research directions are summarized in Section 6.

2. PROBLEM DESCRIPTION AND SOLUTION CONCEPTS

Current Intelligence Surveillance and Reconnaissance (ISR) sensors can detect and take measurements on individual entities, such as moving vehicles and installations. These measurements can be used to infer the particular class of these individual entities. However, very few collection assets provide direct measurements on the hierarchical force structure of units that the entities comprise. Consequently, it is desirable to develop the capability to produce inferences on the hierarchical structure of military units based on inferences and measurements of individual entities and sub-units.

In many cases, the optimality of a sensor allocation policy is defined in terms of reduced tracking error and the best policy is determined through the solution of an optimization problem. While significant progress has been made in this area in the past, there remain open issues in the synthesis and validation of an approach to sensor resource management capable of utilizing high level fusion information.

A technique that can be used to assess the relative merit of aggregate force hypotheses from observations of a set of entities was presented in [10]. The technique draws inferences about the type of military unit that is present given partial observations of entities that comprise the units. Furthermore, making inferences about the type of military unit provides contextual information that enables improved inference about the type of individual vehicles. However, it was pointed out in [10] that the inference process involves intensive computations where the enumeration of an exponentially growing set is needed. In general, this could be very time consuming and may not be practical. In this paper, we develop an efficient approximate algorithm to resolve the combinatorial problem.

In a hierarchical data fusion functional model, high level processing includes estimation and prediction of relations among entities, force structure and cross force relations, communications and perceptual influences, physical context, etc. The goal of this paper is to develop models that estimate relations among entities which can contribute to force structure/composition assessment and to do this in a manner that enables the adaptive sensor management algorithms that have been previously developed.

Herein, we present a hierarchical value function (encompassing both level 1 and level 2 utility) using Bayesian Networks (BNs) [9] to implement sensor resource management algorithms. The valuation function includes both track level and higher-level (entity, convoy, group, scenario, etc.) information. Although significant research has been done in the area of sensor resource management as well as BNs with sensor fusion applications independently, to our knowledge, these two technologies have not been previously applied together to date to solve the higher-level fusion for sensor management problem.

3. SENSOR RESOURCE MANAGEMENT ALGORITHMS

As discussed earlier, the sensor has a capability of determining whether to collect MTI data at a dwell, or instead to collect HRR data on part of a dwell. The sensor management decisions will be based on information reported by a MHT (Multiple Hypothesis Tracker) [11, 15] which processes the sensor measurements.[1] This MHT also includes an ATR capability [6] which provides information on object estimated classifications based on the HRR measurements. The SRM algorithm can be considered as a controller which uses sensor actions to control the evolution of the information incorporated into the MHT algorithm. In order to make "good" sensor management decisions, it is important to model this evolution so that the SRM algorithm can predict the consequences of the alternative decisions.

However, the set of possible information states reported by the MHT for each track is very large. It consists of continuous variables with uncertainty (position, velocity, etc.), plus a set of probability distributions over target type, and discrete variables such as number of missed detection and status. Furthermore, the evolution of this information is highly uncertain, depending on the specific values of the future sensor measurements. For a large number of targets, the set of possible information states is the cross product of individual target states, leading to a large-dimensional continuous-valued state space. Designing feedback controls using such a state space would not lead to a practical real-time sensor management algorithm. An alternative approach is to characterize the information relevant to a track using an aggregate discrete-valued "information quality state" [1]. The model state has two components: tracking and classification quality. Each of these components can take a discrete number of values; thus, its evolution in response to sensor actions can be described by a finite-state Markov chain.

A. Track Quality State

In order to represent the behavior of the tracker algorithm, one way is to represent the track quality as a combination of tracking quality and classification quality. For example, in [1], the tracking quality state con-

[1]Note that the SRM algorithm could take information from any tracker, not necessary a MHT tracker.

sists of: *Undetected*, *Detected*, *New Track*, *Continuing Track*, *Coast 3 to go*, *Coast 2 to go*, *Coast 1 to go*,[2] and *Dropped*. The transition probabilities between these Markov states were given by parameters, which depend on the specific sensor mode being used by the radar, the sensor beam geometry for the track position indicating its probability of detection, and MHT parameters describing how tracks are nominated, promoted and dropped.

The target classification quality was modeled using a similar approach. For example, the classification information can be aggregated into four confidence levels: *Unclassified*, *Low Confidence*, *Medium Confidence*, and *High Confidence*. Note that the classification quality does not depend on the specific identity of an object; instead, it represents confidence in the identity assertion. Thus, the evolution model predicts the confidence improvement which results from specific sensor actions.

In [1], the transition probabilities of the two models were treated independently and computed as a function of the sensor mode separately. However, it is clear that the tracking and classification quality are correlated and should not be considered independently. We therefore develop a joint tracking and classification (JTC) quality state and develop a Markov model accordingly. By considering all the feasible combinations, the resulting model consists of 24 states as shown in Table I. The values (last column) assigned for each JTC state in Table I represent the relative preference of each state by the user. They are assigned heuristically and can be easily modified. More on the quality state value will be described in the next section. The Markov model transition diagram is shown in Fig. 2 and the corresponding transition matrix is given in Table II. Each entry in Table II represents the transition probability between two JTC states. Note that each row in the matrix needs to be normalized in order to make all the outgoing arcs from a state sum to 1.0. Also note that depending on the sensor mode, the transition matrix will be obtained based on the sensor parameters accordingly. The detailed description of the transition probabilities is given in Appendix A.

Given the representation of the information state described above, we can express the sensor management objectives as follows. First, we define the Tracking and Classification (TC) quality states and assign a numerical value $V(TC\_state)$ for every possible TC quality state. For example, a high numerical value would be assigned to a tracking quality of *Continuing_Track* and a classification quality of *High_Confidence*, whereas the lowest value would be assigned to a tracking quality of *Dropped_Track* (see Table I). It may be more important to track objects having a priority classification assessment, such as *time critical targets*. Thus we assume that there are values assigned to the different object classes. We then define an objective function which represents an overall tracking quality value given a sequence of sensor manager decisions. Note that the advantage of this method is that the aggregate Markov chain representation of information quality allows for fast prediction of MHT performance. The result is a practical, predictive model which can be used to evaluate trades between alternative sensor management decisions in real time.

[2]Coast 1 to go is the last tracking quality state before the track will be dropped.

TABLE I
24 State JTC Markov Model

Index	JTC	Tracking	Classification	Values
1	J11	Undetected	Unclassified	0
2	J21	Detection	Unclassified	1
3	J22	Detection	Low confidence	2
4	J31	False	Unclassified	0
5	J41	New track	Unclassified	2
6	J42	New track	Low confidence	3
7	J43	New track	Medium confidence	4
8	J51	Continuing track	Unclassified	7
9	J52	Continuing track	Low confidence	8
10	J53	Continuing track	Medium confidence	9
11	J54	Continuing track	High confidence	10
12	J61	Coast 3 to go	Unclassified	6
13	J62	Coast 3 to go	Low confidence	7
14	J63	Coast 3 to go	Medium confidence	8
15	J64	Coast 3 to go	High confidence	9
16	J71	Coast 2 to go	Unclassified	5
17	J72	Coast 2 to go	Low confidence	6
18	J73	Coast 2 to go	Medium confidence	7
19	J74	Coast 2 to go	High confidence	8
20	J81	Coast 1 to go	Unclassified	4
21	J82	Coast 1 to go	Low confidence	5
22	J83	Coast 1 to go	Medium confidence	6
23	J84	Coast 1 to go	High confidence	7
24	J91	Dropped track	Unclassified	0

B. Sensor Management Objectives

The SRM algorithm is based on an open-loop feedback approach. The basic idea is that at frame t, we generate the desired sequence of decisions for frames t, $t+1,\ldots,t+H$, where H is the planning horizon, based on the aggregate evolution represented by the information quality Markov chains in Fig. 2. We then collect the information from frame t, and receive updated track information from the MHT algorithm. Given this new information, we repeat the process and select decisions for frames $t+1, t+2,\ldots,t+H+1$, receive new information from the MHT and continue the iteration. Thus at each frame t, we compute sensor management decisions for several frames ahead, but use only the next frame's decisions to resolve the SRM problem. An important aspect of the sensor management methodology is that it decides immediate sensor mode commitments with a view towards how these decisions will affect the

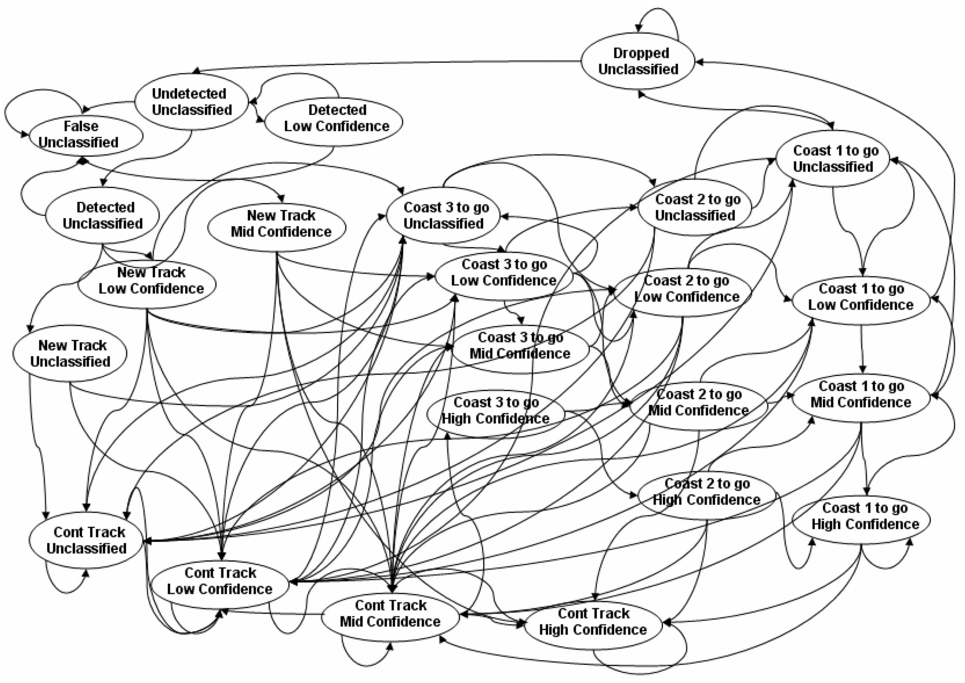
Dropped Unclassified
Undetected Unclassified
Detected Low Confidence
False Unclassified
Detected Unclassified
New Track Mid Confidence
New Track Low Confidence
New Track Unclassified
Coast 3 to go Unclassified
Coast 3 to go Low Confidence
Coast 3 to go Mid Confidence
Coast 3 to go High Confidence
Coast 2 to go Unclassified
Coast 2 to go Low Confidence
Coast 2 to go Mid Confidence
Coast 2 to go High Confidence
Coast 1 to go Unclassified
Coast 1 to go Low Confidence
Coast 1 to go Mid Confidence
Coast 1 to go High Confidence
Cont Track Unclassified
Cont Track Low Confidence
Cont Track Mid Confidence
Cont Track High Confidence

Fig. 2. Markov transition diagram of the joint tracking and classification quality state.

TABLE II
JTC Markov State Transition Matrix

JKC	J11	J21	J22	J31	J41	J42	J43	J51	J52	J53	J54	J61	J62	J63	J64	J71	J72	J73	J74	J81	J82	J83	J84	J91
J11	s1*s4	a1*s4	a1*a4	u1*s4																				
J21				u3*s4	a3*s4	a3*a4																		
J22				u3*u4	a3*u4	a3*s5	a3*a4																	
J31	0.1			0.9																				
J41								a3*s4	a3*a4			u3*s4												
J42								a3*u4	a3*s5	a3*a4		u3*u4	u3*s5											
J43									a3*u4	a3*s5	a3*a4		u3*u4	u3*s5										
J51								a2*s4	a2*u4			u2*s4												
J52								a2*u4	a2*s5	a2*a4		u2*u4	u2*s5											
J53									a2*u4	a2*s5	a2*a4		u2*u4	u2*s5										
J54										a2*u4	a2*s6			u2*u4	u2*s6									
J61								a2*s4	a2*u4			s2*s4	s2*u4			r2*s4								
J62								a2*u4	a2*s5	a2*a4		s2*u4	s2*s5	s2*a4		r2*u4	r2*s5							
J63									a2*u4	a2*s5	a2*a4		s2*u4	s2*s5	s2*a4		r2*u4	r2*s5						
J64										a2*u4	a2*s6			s2*u4	s2*s6			r2*u4	r2*s6					
J71								a2*s4	a2*u4							s2*s4	s2*u4			r2*s4				
J72								a2*u4	a2*s5	a2*a4						s2*u4	s2*s5	s2*a4		r2*u4	r2*s5			
J73									a2*u4	a2*s5	a2*a4						s2*u4	s2*s5	s2*a4		r2*u4	r2*s5		
J74										a2*u4	a2*s6							s2*u4	s2*s6			r2*u4	r2*s6	
J81								a2*s4	a2*u4											s2*s4	s2*u4			r2*s4
J82								a2*u4	a2*s5	a2*a4										s2*u4	s2*s5	s2*a4		r2*u4
J83									a2*u4	a2*s5	a2*a4										s2*u4	s2*s5	s2*a4	
J84										a2*u4	a2*s6											s2*u4	s2*s6	
J91	0.1																							0.9

information state H frames in the future. The size of H reflects a tradeoff between the desire to account for future actions versus the unpredictable evolution of future target motions. Larger values of H introduce more prediction uncertainty into future target positions, thereby making it harder to predict the effect of future sensor actions.

The SRM uses this information as follows. First, for each SRM track created from an MHT track, the classification probabilities of each track are used to assign a value to this SRM track, as follows:

$$V(SRM\_Track) = \sum_{class \in classes} V(class) P(class \mid MHT\_Track) \tag{1}$$

where $class \in \{classes\}$ represents each of the possible target classification, $V(class)$ represents the decision maker's preference/priority on each target *class* and is assumed to be available.

The SRM objective for decisions selected at frame t is as follows. Given a set of SRM decisions for frames $t, t+1, \ldots, t+H$, for each SRM track, we can predict the probability distributions for the track quality state, $P_Q(\,)$, after the information from frame $t+H$ is processed. Denote these decisions as $u_t, u_{t+1}, \ldots, u_{t+H}$, the overall value of this sequence of decisions is computed as [1],

$$\begin{aligned} &J(u_t, u_{t+1}, \ldots, u_{t+H}) \\ &\quad = \sum_{\substack{SRM\_Track \\ \in SRM\_Tracks}} V(SRM\_Track) \\ &\qquad \times \sum_{\substack{JTC\_state \\ \in JTC\_states}} P_Q(JTC\_state \mid SRM\_Track) V(JTC\_state) \end{aligned} \tag{2}$$

where *SRM_Tracks* is the set of all *SRM_Track* and *JTC_States* is the set of all *JTC_State*. In (2), it is implicit that the track quality probabilities depend on the sequence of decisions. The SRM objective function described above represents an assignment of *value* to information quality and to classification of objects. This formulation couples the values of tracking and classification quality.

We will next show how to extend the target valuation models to include a hierarchical structure. The approach will be to modify the target valuation function (2) to include a higher level (cluster or unit) component. We rewrite (2) as

$$\begin{aligned} &J(u_t, u_{t+1}, \ldots, u_{t+H}) \\ &\quad = \sum_{\substack{SRM\_Cluster \\ \in SRM\_Clusters}} V(SRM\_Cluster) \left[\sum_{\substack{Cluster\_Track \\ \in Cluster\_Tracks}} V(Cluster\_Track) \sum_{\substack{JTC\_state \\ \in JTC\_states}} P(JTC\_state \mid Cluster\_Track) V(JTC\_state) \right] \end{aligned} \tag{3}$$

where

$$\begin{aligned} &V(SRM\_Cluster) \\ &\quad = \sum_{Unit\_types} V(Unit\_Type) P(Unit\_Type \mid Cluster\_Tracks) \end{aligned} \tag{4}$$

and *SMR_Cluster* is a group of tracks, denoted as *Cluster_Tracks*, linked together by proximity of a particular type of military unit.

As shown in (3), a *hierarchical target valuation model* is a function of both level 1 (Object Assessment) and level 2 (Situation Assessment) fusion quantities. The model is developed as follows:

1. First, we group the current MHT tracks into clusters.
2. For each cluster, we use a force structure model to infer unit type. We will construct a stochastic model by representing uncertainty (e.g. detection probability and unit composition variation).
3. We then develop a unit-level value function in addition to the entity level tracking and classification value functions as shown in (3). Note that the unit-level valuation function consists of two parts, $V(Unit\_Type)$ and $P(Unit\_Type \mid Cluster\_Tracks)$, as given in (4). $V(Unit\_Type)$ is the default value specified by the decision maker based on their preference/priority on each unit type and $P(Unit\_Type \mid Cluster\_Tracks)$ is the *Unit_Type* probability given a set of tracks computed by the BN force structure model to be discussed in the next section.

C. Bayesian Network Force Structure Model

With Bayes rule, the *Unit_Type* probability given a cluster of tracks can be computed by,

$$\begin{aligned} &P(Unit\_Type \mid Cluster\_Tracks) \\ &\quad = \frac{1}{c} P(Cluster\_Tracks \mid Unit\_Type) P(Unit\_Type). \end{aligned} \tag{5}$$

The solution to (5) represents one of the key contributions described in [10]. $P(Unit\_Type)$ represents the prior probabilities, and $P(Cluster\_Tracks \mid Unit\_Type)$ can be computed as follows:

$$\begin{aligned} &P(Cluster\_Tracks \mid Unit\_Type) \\ &\quad = \sum_{d \in D(n, r+1)} P(Cluster\_Tracks \mid d) P(d \mid Unit\_Type). \end{aligned} \tag{6}$$

In (6), $D(n, r+1)$ is the set of all possible distributions of the n detected vehicles into the $r+1$ possible vehicle classes (including the false-alarm class), which is a

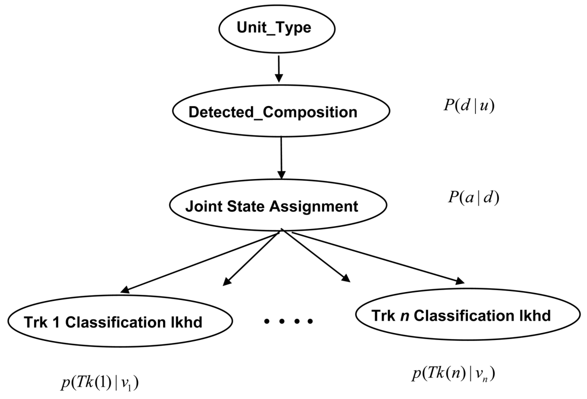

Fig. 3. A Bayesian network model for composition inference.

space with $(r+1)^n$ elements, $P(Cluster\_Tracks \mid d)$ is the likelihood of tracks classification states given the specific detection composition d, and $P(d \mid Unit\_Type)$ is the probability of detection composition given the unit type (see (8)). Note that in (6),

$$\begin{aligned} p(Cluster\_Tracks \mid d) &= \sum_a p(Cluster\_Tracks \mid a) P(a \mid d) \\ &= \sum_a \left[\prod_{k=1}^{n} p(Cluster\_Track(k) \mid v_k) \right] P(a \mid d) \quad (7) \end{aligned}$$

where in (7) $p(Cluster\_Track(k) \mid v_k)$ is the likelihood of producing a track classification state $Cluster\_Track(k)$ given a class v_k vehicle, and a is the joint assignment of a set of vehicles types to a set of tracks. Assuming all joint assignments consistent with the composition constraint are equally likely, then

$$P(a \mid d) = \begin{cases} 1/|\Omega(d)|, & a \in \Omega(d) \\ 0, & \text{otherwise} \end{cases}$$

where

$$|\Omega(d)| = C^{n(d)}_{n(1;d),\ldots,n(r;d)} = \frac{[n(1;d) + \cdots + n(r;d)]!}{n(1;d)! \cdots n(r;d)!}$$

is the set of all joint assignments in which $n(v;d)$ is the number of detected class v vehicles in d.

Also in (6), from the detection model (for simplicity, $u \equiv Unit\_Type$ will be used in the following equations),

$$P(d \mid u) = p_o(n(0;d); \lambda_{FA}) \prod_{v=1}^{r} P(n(v;d) \mid n(v;u)). \quad (8)$$

In (8), $n(0;d)$ is the number of false detections, $n(v;u)$ is the number of class v vehicles in a type u unit,[3] $p_o(k;\lambda) = \lambda^k e^{-\lambda}/k!$ is the Poisson distribution for false alarm detection probability, and

$$\begin{aligned} &P(n(v;d) \mid n(v;u)) \\ &\quad = \sum_{k=0}^{\min[n(v;u),n(v;d)]} B(k; n(v;u), P_D(v)) \cdot p_o(n(v;d) - k; \lambda_C(v)) \end{aligned} \quad (9)$$

[3]Note that the composition of each unit type is assumed to be given.

is the probability of target detection with $B(k;n,p) = C^n_k p^k (1-p)^{n-k}$, a Binomial distribution, where $\lambda_C(v)$ is the density of confusers of class v vehicle.

To implement the hierarchical valuation function, one way is to use the BN model constructed based on (6)–(9) as shown in Fig. 3. Many efficient algorithms exist for BN probabilistic inference [16]. However, (6)–(9) involve intensive computations where the enumeration of an exponentially growing set is needed. In general, this could be very time consuming and may not be practical. We have thus developed an approximate method to simplify the approach, namely,

$$\begin{aligned} &P(Cluster\_Tracks \mid Unit\_Type) \\ &\qquad \approx P(d(Cluster\_Tracks) \mid Unit\_Type) \quad (10) \end{aligned}$$

where $d(Cluster\_Tracks)$ is defined as the joint detection-classification state by collapsing all the track classification probability distributions into one. Namely,

$$d(Cluster\_Tracks) = \sum_{\substack{Cluster\_Track \\ \in Cluster\_Tracks}} P_C(Cluster\_Track),$$

where $P_C(Cluster\_Track)$ is the classification probability distribution of the track $Cluster\_Track$. Note that $d(Cluster\_Tracks)$ is the expected number of targets of each class. Essentially, the approximation amounts to replacing the distribution over numbers of each vehicle type by the mean number of each vehicle type. Then

$$\begin{aligned} &P(d(Cluster\_Tracks) \mid Unit\_Type) \\ &\qquad = \prod_{v=1}^{r} P_B(n(v; d(Cluster\_Tracks)) \mid n(v;u)) \end{aligned} \quad (11)$$

where $P_B(n(v;d(Cluster\_Tracks)) \mid n(v;u))$ is defined similarly to (9). However, since $d(Cluster\_Tracks)$ is a vector of positive real numbers (not necessary integers), it may not be possible to perform the calculation in (9). We therefore approximate it by

$$P_B(n(v;d(Tks)) \mid n(v;u)) \approx N(n(v;d(Tks)); \bar{n}, \sigma_n^2) \quad (12)$$

where $N(n;\bar{n},\sigma_n^2)$ is a Gaussian distribution, $\bar{n} = n(v;u) \cdot P_D(v) + \lambda_C(v)$ is the expected number of detected class v targets, and $\sigma_n^2 = \max[\sigma_{\min}, n(v;u) P_D(v)(1 - P_D(v)) + \lambda_C(v)]$ is the approximate associated variance. With (10)–(12), the composition inference becomes significantly simpler and much more efficient to compute.

4. SENSOR RESOURCE MANAGEMENT EVALUATION ENVIRONMENT

In order to test our target valuation algorithms, we implemented a simple test system as shown in Fig. 4. Note that the purpose of this architecture was not to provide high fidelity modeling conditions; rather, it was designed to be quickly constructed for the purpose of evaluating the proof-of-concept target valuation algorithm(s) that were developed in this paper.

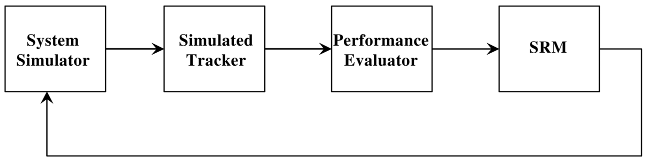

Fig. 4. A SRM system architecture.

The architecture contains an outer loop where at each instant of time, the system simulator creates and sends the current scenario information (related to the targets and sensors) to the simulated tracker. The simulated tracker then produces the simulated tracks with proper joint (tracking and classification) quality states and classification vectors based the Markov transition model as well as the true target/sensor parameters. The simulated tracker then sends the track results to the evaluator. The evaluator uses the tracking results to determine the best sensor mode and pointing direction for next instant of sampling time and sends that decision back to the system simulator. In the remainder of this section, we will provide a description of each of the components in Fig. 4.

A. System Simulator

The system simulator is the overall driver of the system. It generates the ground truth scenarios including target trajectories, group/convoy composition, sensor placements, and sensor observations based on sensor mode/characteristics, as well as sensor/target geometry. Note that the two important aspects of the simulator are: (a) the ability to simulate group/convoy behavior, and (b) the ability to switch sensor operating modes based on the information supplied by the Performance Evaluator and SRM components.

The system simulator sends parameters such as the number of targets, their locations and classes, group composition/identity, sensor mode, detection probability, false alarm density, target density, and confusion matrix/classification probability to the Simulated Tracker. The relative target/sensor geometry (which accounts for a target dropping below the sensor's Minimum Detectable Velocity (MDV) for MTI) is incorporated by the system simulator to produce the required operating parameters.

For simplicity, we leave the burden of representing the convoys/units to the system simulator. Namely, the system simulator will send both target and convoy/group information to the simulated tracker. In the simulation, we assume coverage of all targets in the test scenario area of interest (AOI) when the sensor is in the MTI mode. On the other hand, the HRR mode has a more limited FOV depending on the sensor pointing direction.

B. Simulated Tracker

The purpose of this module is to produce simulated tracking results for performance evaluation without implementing a real tracker. The simulated tracker receives inputs such as ground truth, sensor models, etc. from the system simulator and produce a set of tracks each with tracking and classification joint quality state. Note that for every ground truth target, there is a "track" which will be in one of the joint quality states at each instance of time. For test and validation purposes, we did not attempt to use a complete Multiple Hypothesis Tracker (MHT) for tracking moving ground targets. Rather, the simulated tracker was designed to estimate the joint tracking and classification quality state for each target track.

As described in Section 3, the track quality states behave according to a Finite State Markov transition model based on sensor mode and operating conditions. There are three choices available to the SRM: to use the MTI mode, to use the HRR mode, or to not use the sensor at all. Each sensor mode represents an action that the sensor can take to observe targets. Note that this is a simplified form of the state transitions that were presented earlier, which were chosen on the basis of their ease of implementation.

In addition to the joint quality states, in order to evaluate the track valuation function, each track needs to have an *a posteriori* classification probability distribution. The simulated tracker produces the classification probability vector based on the true target class, previous track class probability, and the current sensor mode and operating conditions. For example, at the beginning of the simulation, each track is at untracked/unclassified state with a uniform classification probability. Depending on the sensor mode (of the next sampling time) and operating characteristics, the track quality state at the next sampling time will be simulated stochastically based on a Finite State Markov transition model. The track classification probability is also updated (accrued over time) using the confusion matrix of the particular sensor mode and the simulated sensor observations. For example, the confusion matrix for GMTI mode is a matrix consisting of uniform probabilities since GMTI mode has no ability to classify target. On the other hand, each row of the HRR mode confusion matrix is the probability of observed classification given a true target class. Note that for the kinematic state, the simulated tracker does not represent the tracks in the target state space, but rather only on the quality state space. The simulated tracker then sends the track results to the evaluator.

C. Performance Evaluator

As mentioned previously, larger values of H (the planning horizon) introduce more prediction uncertainty into future target positions, thereby making it harder to predict the effect of future sensor actions. However, for test purpose we assumed that $H = 1$, (i.e., a one-step look ahead) in order to simplify the implementation (this avoids the need for implementing a dynamic programming algorithm) and to focus on validating the higher level valuation algorithm.

The evaluator receives track results from the simulated tracker. In the results, there is a set of clusters/groups/convoys where each group contains a set of tracks. As described before, each track has a joint quality state and a classification probability vector. The evaluator will use the track results to determine the best sensor mode and pointing direction for the next detection time. In order to do so, in addition to the information from the tracker, truth-related domain knowledge such as unit composition, decision maker's preference value (for each target class and convoy/unit type) are needed as well. The evaluator first finds all the possible detected compositions (based on the possible unit types) and map these to the track compositions of each cluster, which are produced by the simulated tracker. It then computes the value function based on a hierarchical Bayesian model/inference (as described in Section 3) which takes into account the track classification likelihood of a joint state assignment. The evaluator then produces a best sensor mode decision (HRR or MTI) and pointing direction based on the resulting information value and sends these decisions back to the SRM to be used by the system simulator for the next sampling time.

Note that to evaluate the overall system performance, the probability distributions of each unit and individual target are produced by the simulated tracker based on the selected sensor decision. The tracking results are then averaged over multiple Monte Carlo runs as will be described in the next section.

5. SIMULATION RESULTS

We implemented the system described in Fig. 4 and also defined a set of metrics that can be used for evaluation purposes:

- *Sensor allocated resources*—the percentage of time that the sensor is operating in HRR mode (HRR Rate)
- *Average probability of correct unit classification*—the average correct unit classification probability over the simulation time (P_{cc})
- *Average percentage of correctly identified targets*—the average percentage of correct target classification in each unit over the simulation time (Trk Rate)

In order to test these algorithms, we implemented a simple ground moving target scenario containing convoy units, each consisting of a different combination of target types. There are a total of 4 possible types of unit: Scud (class 1), C2 (class 2), Tank (class 3), and Unknown, as well as 6 target classes: UAZ-469 (class 1), ZIL-151 (class 2), GAZ-66 (class 3), MAZ-543 (class 4), T-72 (class 5), and Other (class 6). However, in this particular scenario, ground truth only contains two units and four target classes: unit 1 (containing 2 UAZ-469 vehicles, as well as one of ZIL-151, GAZ-66, and MAZ-543 each) and unit 2 (containing 5 UAZ-469 vehicles).

The parameters used for the simulation are summarized in Appendix B. We made several test trials with different value functions and strategies. Specifically, in each of the combinations below, the notation "Case xy" refers to $x = unit\_type$ and $y = target\_class$. Also, the notation $[a\ b\ c\ d]$ refers to the valuation of each unit (since there are 4 possible units for this particular scenario) and the notation $[a\ b\ c\ d\ e\ f\ g]$ similarly refers to the valuation of individual targets, since there are 6 possible targets. Note that for simplicity, the target valuation is assumed to be a "binary" variable (i.e., valued at either 0 or 1); other combinations of target values are certainly possible, but were not considered here. This means that, for this particular scenario, there are a total of 15 possible combinations, as follows: $2\ (\textit{unit classes}) * 4\ (\textit{target classes}) + 4\ (\textit{level 1 valuation only}) + 2\ (\textit{level 2 valuation only}) + 1\ (\textit{neither level 1 nor level 2 valuation}) = 15$. Also, we assume the cost of using the HRR mode is about twice as expensive than the MTI mode.

Level 1 value function only, unit class value [1 1 1 1]
- Case 01: Focus on target class 1, track class value: [1 0 0 0 0 0]
- Case 02: Focus on target class 2, track class value: [0 1 0 0 0 0]
- Case 03: Focus on target class 3, track class value: [0 0 1 0 0 0]
- Case 04: Focus on target class 4, track class value: [0 0 0 1 0 0]

Level 2 value function only, target class value [1 1 1 1 1 1]
- Case 10: Focus on unit class 1, unit class value: [1 0 0 0]
- Case 20: Focus on unit class 2, unit class value: [0 1 0 0]

Level 1 and 2 value functions
- Case 11: Focus on unit class 1, [1 0 0 0], and target class 1, [1 0 0 0 0 0]
- Case 12: Focus on unit class 1, [1 0 0 0], and target class 2, [0 1 0 0 0 0]
- Case 13: Focus on unit class 1, [1 0 0 0], and target class 3, [0 0 1 0 0 0]
- Case 14: Focus on unit class 1, [1 0 0 0], and target class 4, [0 0 0 1 0 0]
- Case 21: Focus on unit class 2, [0 1 0 0], and target class 1, [1 0 0 0 0 0]
- Case 22: Focus on unit class 2, [0 1 0 0], and target class 2, [0 1 0 0 0 0]
- Case 23: Focus on unit class 2, [0 1 0 0], and target class 3, [0 0 1 0 0 0]
- Case 24: Focus on unit class 2, [0 1 0 0], and target class 4, [0 0 0 1 0 0]

Neither level 1 nor level 2 value functions
- Case 00: unit class value, [1 1 1 1], and target class value, [1 1 1 1 1 1]

Note that for Case 00, in order to ignore target (or unit) value, all of the entities are equally valued and have a value of 1, e.g., a target class of [1 1 1 1 1 1].

TABLE III
Performance Results using Only Level 1 Value Function

Case	HRR U1 Rate	HRR U2 Rate	U1 Avg P_{cc}	U2 Avg P_{cc}	U1 Trk Rate	U2 Trk Rate
00	0.0032	0.0020	0.5671	0.5466	0.3173	0.3140
01	0.0000	0.0260	0.5000	0.8596	0.2500	0.8585
02	0.0036	0.0000	0.5838	0.5000	0.3405	0.2500
03	0.0068	0.0000	0.6502	0.5000	0.4242	0.2500
04	0.0076	0.0000	0.6656	0.5000	0.4243	0.2500

TABLE IV
Performance Results using Only Level 2 Value Functions

Case	HRR U1 Rate	HRR U2 Rate	U1 Avg P_{cc}	U2 Avg P_{cc}	U1 Trk Rate	U2 Trk Rate
00	0.0032	0.0020	0.5671	0.5466	0.3173	0.3140
10	0.0236	0.0000	0.9746	0.5000	0.8150	0.2500
20	0.0000	0.0236	0.5000	0.8925	0.2500	0.7777

TABLE V
Performance Results using Both Level 1 and Level 2 Value Functions

Case	HRR U1 Rate	HRR U2 Rate	U1 Avg P_{cc}	U2 Avg P_{cc}	U1 Trk Rate	U2 Trk Rate
10	0.0236	0.0000	0.9746	0.5000	0.8150	0.2500
11	0.0240	0.0000	0.9868	0.5000	0.8162	0.2500
12	0.0224	0.0000	0.9127	0.5000	0.7482	0.2500
13	0.0212	0.0000	0.9181	0.5000	0.7790	0.2500
14	0.0244	0.0000	0.9695	0.5000	0.8148	0.2500
20	0.0000	0.0236	0.5000	0.8925	0.2500	0.7777
21	0.0000	0.0336	0.5000	0.9288	0.2500	0.9088
22	0.0008	0.0056	0.5182	0.6089	0.2721	0.3922
23	0.0000	0.0048	0.5000	0.5730	0.2500	0.3586
24	0.0004	0.0068	0.5000	0.6219	0.2500	0.4301

With 50 Monte Carlo simulations, the average performance for several different cases are summarized in the following tables. Table III shows the performance results using only level 1 valuation functions. Note that case 00, by definition, contains neither level 1 nor level 2 valuation functions. In the table, the HRR rates (percentage of time spent in HRR mode versus GMTI mode) for unit 1 and unit 2 are shown in columns 2 and 3, average correct classification probabilities for unit 1 and unit 2 are given in cloumns 4 and 5, and the average correct track classification probabilities of each unit are shown in the last two cloumns respectively.

The results show that the classification performance of both units are in the range of 50–60%. Both unit 1 and unit 2 classifications improve somewhat when adding the "right" level 1 valuation. For example, when adding target class 2, 3, and 4 valuation functions, unit 1 classification increases from $50+\%$ to around $60+\%$ while unit 2 classification drop slightly from 55% to 50%. Similarly, when adding a target class 1 valuation function, unit 1 classification decreases to 50% while unit 2 classification increases significantly to 86%. This is understandable since unit 1 consists of all 4 classes of targets and unit 2 contains only class 1 targets.

Table IV shows the corresponding performance results using only level 2 value functions. It can be seen that the classification performance improves significantly compared to the level 1 performance. For example, for case 10, since the emphasis (i.e., valuation) is on unit 1, the resulting sensor strategy improves the unit 1 classification performance significantly from 57% to 97%. Similarly, for case 20, unit 2 classification increases from 55% to 89%. Note here that the track level classification probabilities also improve significantly from around 30% to 80% despite no level 1 valuation being used.

Table V shows the performance results using both level 1 and level 2 value functions. It can be seen that the classification performance values are similar to those obtained using only level 2 value functions. For example, in cases 11 through 14, since the emphasis is on unit 1, the unit 1 classification performance is similar to that of case 10. However, for cases 21 through 24, only case 21 performs similarly to case 20—the others perform significantly worse in being able to classify unit 2. This is because, interestingly enough, unit 2 includes only target class 1. Adding a level 1 track value function of the other target class (not included in unit 2) not only does not help in classifying unit 2, but it also manages to confuse the sensor manager and subsequently deteriorates the performance (relative to case 20) significantly.

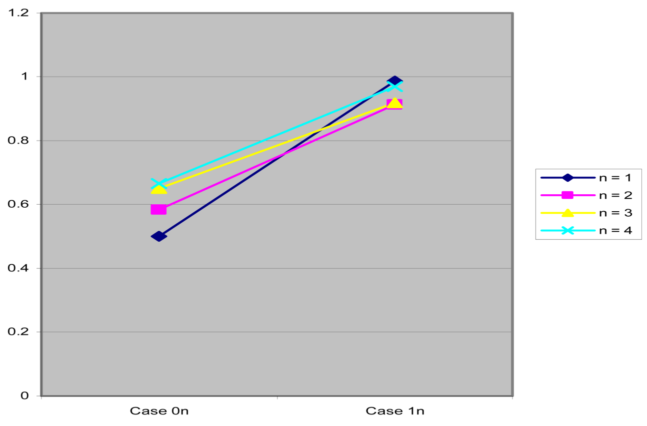

Fig. 5. Unit 1 average Pcc with level 1 and level 2 valuation functions.

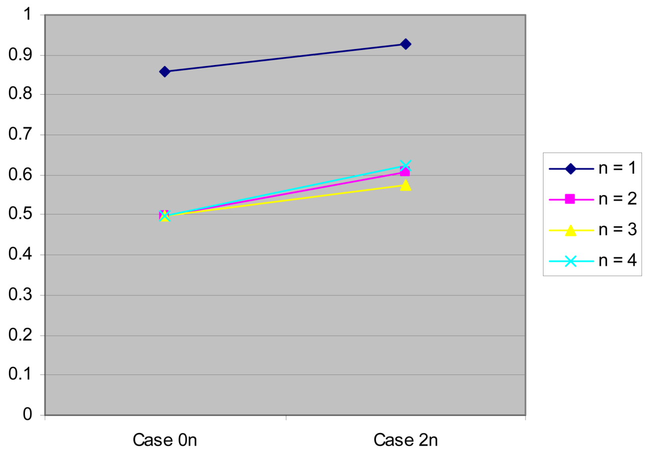

Fig. 6. Unit 2 average Pcc with level 1 and level 2 valuation functions.

Another useful comparison is to determine the possible benefit of adding a level 2 valuation function to the level 1 function. This comparison is particularly relevant because use of level 1 (target-based) valuation can be considered to be the typical baseline or current operating mode for most sensor management systems. In order to perform this comparison, we extract comparable portions of Table III and Table V and display these values in Figs. 5 and 6, where the comparison is between case $0n$ with cases $1n$ and $2n$ respectively, where $n = 1,\ldots,4$. Note that case $0n$ represents level 1 valuation that emphasizes target class n and case jn represents level 1 and level 2 valuations that emphasize unit type j and target class n.

As seen from the figures, in all 4 cases, unit 1 P_{cc} improve significantly (from about 60% to 90%) when adding level 2 valuation function. However, the unit 2 P_{cc} only improve moderately (about 10%) when adding a level 2 function that emphasizes unit 2 valuation. Again, this is because unit 2 consists of only target type 1, the combination of "inconsistent" unit level and target level values (such as 22, 23, and 24) simply will not help the classification performance.

In summary, as seen in the results, by adding a level 2 valuation function, the performance improves significantly. Particularly, without level 2 function, the performance for unit level classification is mostly unsatisfactory. It is interesting to note that, with a level 2 valuation function, the track level performance also improve slightly when the objective functions of the two levels are consistent. When the objective functions are inconsistent or contradictory as we described above, the performance may not improve as expected. Note that since we are not comparing performance between the proposed approach and an alternative baseline, the numerical results imply, not so much that performance uniformly improves when optimizing the proposed objective criterion but that the objective criterion is a reasonable one to optimize to meet both level 1 and 2 fusion objectives.

6. SUMMARY

In this paper, we have presented an approach for dynamically choosing sensor mode and pointing direction based on both level 1 and level 2 information. Specifically, a hierarchical target valuation model based on track quality value was presented. The valuation algorithm relies on a Bayesian approach where a recursive composition inference algorithm was used to compute the hierarchical valuation function. This approach not only will provide for adequate object identification and tracking performance, but also can provide the ability to be able to identify higher-level entities such as convoys.

We have also developed an evaluation environment to analyze the performance of this valuation algorithm given a set of ground moving targets. The preliminary simulation results demonstrate the validity of our approach. In order to completely validate algorithm performance, it will be necessary to implement the algorithm in a higher fidelity modeling environment, including more complex algorithms for the tracker and SRM. Nevertheless, the algorithms presented in this paper represent a significant step toward efficient sensor management using higher level valuation and objective functions. Some useful future research directions include extending the hierarchical valuation model to account for level 3 (eg., intent assessment) function and developing a analytical prediction model to estimate the SRM performance without extensive Monte Carlo simulations.

APPENDIX A

The parameters in the transition matrix of the tracking and classification quality Markov chains are defined as the follows.

(1) $a_1 = P_{new}P_d$ (object, sensor_mode), $u_1 = 1 - a_1$, $s_1 = 0$, when a potential track is covered by the sensor

beam, where P_{new} is the probability of new target arrivals per unit position, and P_d is the probability of detection calculated based on the relative predicted target-sensor geometry and the sensor mode as well as the target radial velocity (for MDV purposes).

(2) $u_1 = a_1 = 0$ and $s_1 = 1$ when a potential track is not covered by the sensor.

(3) $a_3 = P_d$ (*object, sensor_mode*) is the probability that a second beam look results in an initial track, $u_3 = 1 - a_3$.

(4) $a_2 = P_d$ (*object, sensor_mode*), $r_2 = 1 - a_2$, $s_2 = 0$, when the object is covered by the sensor beam.

(5) When the object is unobservable, $a_2 = 0$, $r_2 = rate$, $s_2 = 1 - r_2$, where $rate = 3F_d/T_{drop}$ is the probability an unobservable object will reach the dropped state in T_{drop} expected time. Note that F_d is the frame duration and T_{drop} is the maximum time that a track can be kept coasting before the MHT algorithm drops the track.

(6) a_4 $(HRR) = P_{improve}$, a_4 $(MTI) = 0$, $s_4 = 1 - a_4$ where $P_{improve}$ is the probability that classification quality will improve if an HRR model is used.

(7) u_4 $(MTI) = P_{degrade}$, u_4 $(HRR) = 0$, $s_5 = 1 - a_4 - u_4$, $s_6 = 1 - u_4$ where $P_{degrade}$ is the probability that classification quality will degrade if an MTI model is used.

APPENDIX B

The parameters in the simulation are given as the follows.

(1) probability of detection: P_d $(GMTI) = 0.9$, P_d $(HRR) = 0.5$

(2) probability of classification: P_C $(GMTI) = 1/n$, P_C $(HRR) = 0.9$[4]

(3) probability of new target arrivals per unit position: $P_{new} = 0.1$

(4) the probability that classification quality will improve with HRR mode: $P_{improve} = 0.8$

(5) the probability that classification quality will degrade with GMTI mode: $P_{degrade} = 0.8$

(6) the probability an unobservable object will reach the dropped state: $rate = 0.1$

(7) the values of joint quality state: given in the last column of Table I.

(8) Decision maker's preference value for each tareget and unit: varied in each test case, see Section 5.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers and Editor for their constructive comments that helped improve the paper.

[4]It is assumed that the probability of misclassification is uniform across target types.

REFERENCES

[1] T. Allen, D. Castañon and R. Washburn
Stochastic dynamic programming for farsighted sensor management.
Phase II SBIR Final Report, ALPHATECH Technical Report, TR-974, June 2000.

[2] R. Popp, A. Bailey and J. Tsitsiklis
Dynamic airborne sensor resource management for ground moving target tracking and classification.
In *Proceedings of IEEE Aerospace Conference*, Big Sky, MO, Mar. 2000.

[3] D. Bertsekas and D. Castañon
Rollout algorithms for stochastic scheduling.
Heuristics, **5** (1999).

[4] D. Bertsekas and J. Tsitsiklis
Dynamic Programming and Optimal Control vol. I, II.
Belmont MA: Athena Scientific, 2001.

[5] D. Castañon
Approximate dynamic programming for sensor management.
In *Proceedings of the 1997 Conference on Decision and Control*.

[6] K. Chang and R. Fung
Target identification with Bayesian networks in a multiple hypothesis tracking system.
SPIE Optical Engineering Journal, **36**, 3 (Mar. 1997), 684–691.

[7] J. Hill and K. Chang
Improved representation of sensor exploitation for automatic sensor management.
In *Proceedings of Sixth International Conference on Information Fusion*, Cairns, Queensland, Australia, July 2003, 688–694.

[8] J. Hill and K. Chang
Sensor resource management with hierarchical target valuation models.
In *Proceedings of Seventh International Conference on Information Fusion*, Stockholm, Sweden, July 2004.

[9] F. Jensen
An Introduction to Bayesian Networks.
Spinger, NY, 1996.

[10] J. Johnson and R. Chaney
Recursive composition inference for force aggregation.
In *Proceedings of Second International Conference on Information Fusion*, July 1999.

[11] T. Kurien
Issues in the design of practical multitarget tracking algorithms.
In Y. Bar-Shalom (Ed.), *Multitarget-Multisensor Tracking: Applications*, Artech House, 1990, ch. 2.

[12] R. Mahler
Random set theory for target tracking and identification.
In D. L. Hall and J. Llinas (Eds.), *Handbook of Multisensor Data Fusion*, Boca Raton FL: CRC Press, 2002, ch. 14.

[13] R. Mahler
Target preference in multitarget sensor management: A unified approach.
In *Proceedings of SPIE conference*, Orlando, FL, 2004.

[14] J. Manyika
An information-theoretic approach to data fusion and sensor management.
Ph.D. thesis, University of Oxford, 1993.

[15] S. Mori, C. Y. Chong, E. Tse and R. Wishner
Tracking and classifying multiple targets without a priori identification.
IEEE Transaction on Automatic Control, **AC-31**, 5 (1985), 401–409.

[16] J. Pearl
Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.
Morgan Kaufmann, 1989.

[17] A. Steinberg, C. Bowman and F. White
Revisions to the JDL data fusion model.
In *Proceedings of SPIE, Sensor Fusion: Architectures, Algorithms, and Applications III*, 1999.

[18] R. B. Washburn, M. K. Schneider and J. J. Fox
Stochastic dynamic programming based approaches to sensor resource management.
In *Proceedings of the Fifth International Conference on Information Fusion*, Annapolis, MD, July 2002.

K. C. Chang received the M.S. and Ph.D. degrees in electrical engineering from the University of Connecticut in 1983 and 1986 respectively.

From 1983 to 1992, he was a senior research scientist in the Advanced Decision Systems (ADS) division, Booz-Allen & Hamilton, California. In 1992, he joined the Systems Engineering and Operations Research Department, George Mason University where he is currently a professor. His research interests include estimation theory, optimization, signal processing, and multisensor data fusion. He is particularly interested in applying unconventional techniques to the conventional decision and control systems. He has more than 25 years of industrial and academic experience and has published more than one hundred and forty papers in the areas of multitarget tracking, distributed sensor fusion, and Bayesian Network technologies. He was the editor on Large Scale Systems of *IEEE Transactions on Aerospace and Electronic Systems* and an associate editor of *IEEE Transactions on Systems, Man, and Cybernetics.*

Dr. Chang is a member of Etta Kappa Nu and Tau Beta Pi.

Joe Hill received the M.S. degree in electrical engineering (Systems and Applied Mathematics) from the University of Notre Dame in 1982 and his B.S.E.E. degree from Valparaiso University in 1979, with highest distinction.

He has developed error covariance models for Submarine Launched Ballistic Missile weapon and Navigation systems (TASC, 1984–1991), developed hit to kill models of missile/interceptor endgame performance in support of the Missile Defense Agency (Coleman Research, 1991–1996), and developed and tested collection performance/geolocation models for several overhead sensor systems (Ultrasystems/Logicon/Northrop Grumman Information Technology, 1996–2001). From 2001 to 2007, he was a senior engineer with ALPHATECH, Inc. (now a part of BAE Systems). He is currently a senior engineering specialist with SRS Technologies. His current areas of interest are sensor modeling, scheduling, tracking performance, route planning, resource management, control, and optimization.

He is a member of Tau Beta Pi, and ISIF.