Information Fusion of Passive Sensors for Detection of Moving Targets in Dynamic Environments

Yue Li, Devesh K. Jha, *Student Member, IEEE*, Asok Ray, *Fellow, IEEE*, and Thomas A. Wettergren, *Senior Member, IEEE*

***Abstract*—This paper addresses the problem of target detection in dynamic environments in a semi-supervised data-driven setting with low-cost passive sensors. A key challenge here is to simultaneously achieve high probabilities of correct detection with low probabilities of false alarm under the constraints of limited computation and communication resources. In general, the changes in a dynamic environment may significantly affect the performance of target detection due to limited training scenarios and the assumptions made on signal behavior under a static environment. To this end, an algorithm of binary hypothesis testing is proposed based on clustering of features extracted from multiple sensors that may observe the target. First, the features are extracted individually from time-series signals of different sensors by using a recently reported feature extraction tool, called symbolic dynamic filtering. Then, these features are grouped as clusters in the feature space to evaluate homogeneity of the sensor responses. Finally, a decision for target detection is made based on the distance measurements between pairs of sensor clusters. The proposed procedure has been experimentally validated in a laboratory setting for mobile target detection. In the experiments, multiple homogeneous infrared sensors have been used with different orientations in the presence of changing ambient illumination intensities. The experimental results show that the proposed target detection procedure with feature-level sensor fusion is robust and that it outperforms those with decision-level and data-level sensor fusion.**

***Index Terms*—Decision-making in dynamic environments, feature-level sensor fusion, target detection.**

I. Introduction

DETECTION of moving targets in uncertain and dynamic environments is of utmost importance for intelligence, surveillance, and reconnaissance systems (e.g., Secure Border Initiative by the Department of Homeland Security). To this end, large distributed networks of passive sensors are often attractive because they can provide coverage of wide areas at a moderate running cost. Every sensor in the network is provided with communication and computing capabilities for the collective intelligent behavior. A key requirement for cost-effective operation of such distributed sensor networks is to maintain reliable sensing performance, while limiting the amount of communications required for reliable decision fusion and situation awareness across the network [1], [2]. In addition, limited computational capabilities of the individual sensors require low-complexity data-processing algorithms for information extraction from the time series of sensed signals. Decision-making with passive sensing in a dynamic environment is an algorithmically challenging problem as these sensors (e.g., acoustics, seismic, and infrared) are often sensitive to the environmental uncertainties and these effects, in general, are difficult to model using the fundamental principles of physics.

While maintaining reliable sensing performance, the following research challenges need to be simultaneously addressed as key requirements for operating distributed sensor networks in a cost-effective way.

1) Constrained communication at sensing nodes, which would require a feasible framework for information compression and distributed information fusion to reduce the communication overhead.
2) Limited computational capabilities at sensing nodes, which would require low-complexity algorithms for information extraction and decision analysis.
3) Operation under dynamic environments, which may significantly compromise the performance of supervised algorithms for object and situation assessment.

If a large field of passive sensors is deployed, the decision parameters of the individual sensors need to be tuned for accurate detection, classification, and tracking. Most of this paper on distributed decision fusion, which has been based on threshold rules, use decision rules such as voting, Bayes criterion, and Neyman–Pearson methods [3] by using *a priori* knowledge on the probability of target presence [4]–[6].

The above methods rely on available sensor models that often do not include the effects of background noise in the operating environment. However, it is very likely that a dynamic environment would interfere with the sensor data that correspond to the events of interest (e.g., a mobile target); hence, the decision rules based on thresholding of signal variables may not hold in a different background environment.

Manuscript received August 1, 2015; accepted December 8, 2015. Date of publication March 3, 2016; date of current version December 14, 2016. This work was supported in part by the U.S. Office of Naval Research under Grant N00014-14-1-0545, in part by the U.S. Air Force Office of Scientific Research under Grant FA9550-15-1-0400, and in part by the U.S. Army Research Laboratory under Grant W911NF-14-2-0068. This paper was recommended by Associate Editor R. Lynch.

Y. Li, D. K. Jha, and A. Ray are with the Department of Mechanical and Nuclear Engineering, Pennsylvania State University, State College, PA 16802 USA (e-mail: yol5214@psu.edu; dkj5042@psu.edu; axr2@psu.edu).

T. A. Wettergren is with Naval Undersea Warfare Center, Newport, RI 02841 USA, and also with the Department of Mechanical and Nuclear Engineering, Pennsylvania State University, State College, PA 16802 USA (e-mail: t.a.wettergren@ieee.org).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2015.2508024

A more efficient way of event (e.g., target) detection is usage of the pattern matching concept that may require feature extraction, followed by classification of sensor data for event detection. Duarte and Hu [7] evaluated several classification algorithms for vehicle detection and tracking, and other types of implementation can be found in [8]–[11]. However, most of the classification-based detection schemes require significant computational resources for signal conditioning and feature extraction; these techniques may also require scenario-specific training to initialize the classifier. Furthermore, changes in the boundaries of classification decision could lead to erroneous predictions if the background environment is dynamic.

In spite of a large volume of work on collaborative target detection in sensor networks, a majority of the algorithms reported in this paper are not robust relative to environmental changes, especially with the usage of low-fidelity, passive sensors. The rationale is that the passive sensors, used in a network, are sensitive to the background noise of the environment.

In many state-of-the-art techniques, the environmental disturbance is modeled as an additive noise, which is a compromising simplification because the environment is usually nonlinearly coupled with the event of interest. It is also noted that noise in a dynamic environment, albeit being different from high-frequency additive noise, may generate a similar texture in the time series, which could be treated as a possible event of interest in the same time scale. Consequently, false interpretations could be derived based on the hard decision boundaries that are learned in the training phase. The decisions made by using thresholds based on sensed data or classification boundaries trained for identifying target classes may lead to false interpretations during the actual operation. Even if the final decisions are made by collaboration between different nodes [12], they could be significantly inaccurate.

From the above perspectives, a distributed framework for feature-level fusion of sensor time series has been proposed for target detection in dynamic environments [13]; the underlying theory is built upon a recently reported tool of feature extraction, called symbolic dynamic filtering (SDF) [14], [15]. The compressed information in the extracted features is then clustered in an unsupervised fashion for target detection via statistical hypothesis testing. Because of their different spatial locations and orientations, the sensors have different degrees of influence from an event (e.g., a mobile target) of interest. The key idea here is to capture different levels of the target's influence on the individual sensors for fusion by clustering of the collected information. A decision rule is then formulated for target detection based on the behavior of a pair of clusters in the feature space.

In contrast to the conventional notion of probabilistic data association filtering that relies on a dynamic model of the physical process (see [16], [17], and references therein), this paper focuses on symbolic-dynamic-based fusion of the information derived from time series of different sensors for target detection in a dynamic environment. Major contributions of the work reported in this paper, which belongs to a class of dynamic data-driven application systems [18], are outlined as follows.

1) Development of a feature-level sensor fusion algorithm in a distributed setting, which can be executed in real time for target detection in dynamic environments.
2) Experimental validation of the target detection algorithm in a laboratory setting, which demonstrates its superior performance relative to equivalent decision-level and data-level fusion scheme.

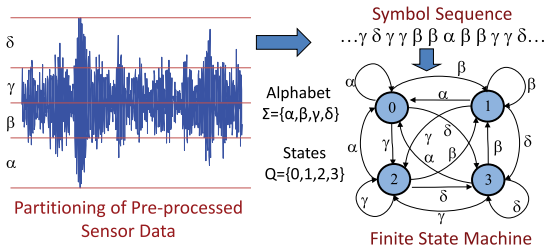


Fig. 1. Underlying concept of FSA.

This paper is organized in seven sections including the current section. Section II briefly describes the underlying principles of SDF that is used for feature extraction of the (possibly preprocessed) time-series signal from sensors. Section III presents the statement of the addressed problem along with a list of major assumptions. Section IV elaborates the proposed approach adopted in this paper. Section V briefly presents the details of the experimental facility and procedure. Section VI presents the results of experimental validation of the proposed concept. Finally, this paper is summarized and concluded in Section VII along with recommendations for future research.

II. Signal Feature Extraction

This section briefly describes the underlying concept of feature extraction from time series data, upon which the proposed sensor-data-driven tool of target detection is constructed [14], [15]. The following standard definitions are recalled.

Definition 1 (FSA [19], [20]): A finite state automaton (FSA) G, having a deterministic algebraic structure, is a triple (Σ, Q, δ), where:

1) Σ is a (nonempty) finite alphabet with cardinality $|\Sigma|$;
2) Q is a (nonempty) finite set of states with cardinality $|Q|$;
3) $\delta : Q \times \Sigma \to Q$ is a state transition map.

Fig. 1 illustrates the concept of constructing (deterministic) FSA from time series, which forms the algebraic structure of probabilistic FSA (PFSA) as defined as follows.

Definition 2 (PFSA): A PFSA is a quadruple $K = (\Sigma, Q, \delta, \pi)$, where:

1) the alphabet Σ of symbols is a (nonempty) finite set;
2) the set Q of automaton states is (nonempty) finite;
3) the state transition function $\delta : Q \times \Sigma \to Q$;
4) the morph function $\pi : Q \times \Sigma \to [0, 1]$, where $\sum_{\sigma \in \Sigma} \pi(q, \sigma) = 1$ for all $q \in Q$. The morph function π generates the $(|Q| \times |\Sigma|)$ morph matrix Π.

A. Symbolization of Time Series

This step requires partitioning (also known as quantization) of the time series data of the measured signal. The signal space

is partitioned into a finite number of cells that are labeled as symbols, i.e., the number of cells is identically equal to the cardinality $|\Sigma|$ of the (symbol) alphabet Σ. As an example for the 1-D time series in Fig. 1, the alphabet $\Sigma = \{0, 1, 2, 3\}$, i.e., $|\Sigma| = 4$, and three partitioning lines divide the ordinate (i.e., y-axis) of the time series profile into four mutually exclusive and exhaustive regions. These disjoint regions form a partition, where each region is labeled with one symbol from the alphabet Σ. If the value of time series at a given instant is located in a particular cell, then it is coded with the symbol associated with that cell. (Details are reported in [21].) Thus, a (finite) array of symbols, called a symbol string (or symbol block), is generated from the (finite-length) time series data.

The ensemble of time series data are partitioned by using a partitioning tool [e.g., maximum entropy partitioning (MEP)] that maximizes the entropy of the generated symbols; therefore, the information-rich cells of a data set are partitioned finer and those with sparse information are partitioned coarser [i.e., each cell contains (approximately) equal number of data points under MEP]. The choice of alphabet size $|\Sigma|$ largely depends on the specific data set and the allowable loss of information (e.g., leading to error of detection and classification) [15].

B. D-Markov Modeling

For both algorithmic simplicity and computational efficiency, the D-Markov machine structure [14], [15] has been adopted for construction of PFSA. It is noted that D-Markov machines form a proper subclass of hidden Markov models and have been experimentally validated for applications in various fields of research (e.g., anomaly detection and robot motion classification [22]). A D-Markov chain is modeled as a statistically locally stationary stochastic process $S = \cdots s_{-1}s_0 \cdots s_1 \cdots$, where the probability of occurrence of a new symbol depends only on the last D symbols, that is

$$P\big[s_n \mid \cdots s_{n-D} \cdots s_{n-1}\big] = P\big[s_n \mid s_{n-D} \cdots s_{n-1}\big].$$

Given a finite-length symbol string over a (finite) alphabet Σ, there exist several PFSA construction algorithms to discover the underlying irreducible PFSA model, such as causal-state splitting reconstruction [23] and D-Markov machines [14], [15]. These algorithms start with identifying the structure of the PFSA $K \triangleq (Q, \Sigma, \delta, \pi)$. Then, a $|Q| \times |\Sigma|$ count matrix C is initialized to the matrix, each of whose elements is set to 1.

Let $N(\sigma_j|q_k)$ denote the number of times that a symbol σ_j is generated from the state q_k as the symbol string evolves. The maximum *a posteriori* probability (MAP) estimate of the morph probability (see Definition 2) for the PFSA K is estimated by frequency counting as

$$\hat{\pi}\big(\sigma_j|q_k\big) \triangleq \frac{C\big(\sigma_j|q_k\big)}{\sum_\ell C(\sigma_\ell|q_k)} = \frac{1 + N\big(\sigma_j|q_k\big)}{|\Sigma| + \sum_\ell N(\sigma_\ell|q_k)}. \quad (1)$$

The rationale for initializing each element of the count matrix C to 1 is that if no event is generated at a state $q \in Q$, then there should be no preference to any particular symbol and it is logical to have $\hat{\pi}_{\text{MAP}}(q, \sigma) = (1/|\Sigma|)\ \forall \sigma \in \Sigma$ (i.e., the uniform distribution of event generation at the state q). The above procedure guarantees that the PFSA, constructed from a (finite-length) symbol string, must have an (elementwise) strictly positive morph map π.

Having computed the probabilities $\hat{\pi}(\sigma_j|q_k)$ for all $\sigma_j \in \Sigma$ and $q_k \in Q$, the estimated emission probability matrix of the PFSA is obtained as

$$\widehat{\mathbf{\Pi}} \triangleq \begin{bmatrix} \hat{\pi}(\sigma_1|q_1) & \cdots & \hat{\pi}\big(\sigma_{|\Sigma|}|q_1\big) \\ \vdots & \ddots & \vdots \\ \hat{\pi}\big(\sigma_1|q_{|Q|}\big) & \cdots & \hat{\pi}\big(\sigma_{|\Sigma|}|q_{|Q|}\big) \end{bmatrix}. \quad (2)$$

The stochastic matrix $\widehat{\mathbf{\Pi}}$, estimated from a symbol string, is then treated as a feature of the times series data that represents the behavior of the dynamical system. These features can then be used for different learning applications, such as pattern matching, classification, and clustering. The estimated stochastic matrices $\widehat{\mathbf{\Pi}}$ are converted into row vectors for feature divergence measurement by sequentially concatenating the $|Q|$ rows [i.e., the $|Q| \times |\Sigma|$ matrix $\widehat{\mathbf{\Pi}}$ is vectorized as a $(1 \times |Q||\Sigma|)$ vector].

This completes the signal feature extraction step. It is noted that the only selectable parameters of the feature extraction technique described above are the alphabet size $|\Sigma|$ and memory (depth) D [24].

III. Problem Statement

This section formulates the problem of target detection in a dynamic environment. The presence of a target is determined by analyzing the ensemble of time series data generated from a number of sensors on a network. To this end, a set of sensors is deployed in a local region of the sensor network. The observation ensemble $\mathbb{Y} = \{Y_1, Y_2, \ldots, Y_n\}$ represents the set of time-series data generated by the ensemble of sensors $\mathcal{L} = \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n\}$.

Target detection in this paper is based on a binary hypothesis test of H_0 versus H_1 that represent the hypotheses corresponding to "environmental disturbances only" (i.e., absence of a target) and "presence of a target," respectively. That is, H_0 is referred to as the null hypothesis and H_1 as the alternate hypothesis. In this setting, the problem of target detection in a dynamic environment is to find a decision rule for H_0 versus H_1 by a synergistic combination of the data collected by the sensor set $\mathcal{L}$ under the following assumptions.

1) The sensors are homogeneous, i.e., all sensors are of the same modality.
2) Even though the sensors have the same modality, their deployment is heterogeneous in the sense that they are not co-located and that their orientations are different; therefore, different sensors may yield different outputs in the presence of a single target.
3) Within a given time span of observation, there is at most one (moving) target. In other words, the observed time series outputs of the sensors correspond to the same event (i.e., the same target if it is present).
4) The time scale of environmental changes is largely similar to that of event occurrence (e.g., target motion).

In other words, the environment is dynamically changing during the observation of a local event.

The Assumptions 1)–4), stated above, are often encountered in distributed sensing applications. The usage of homogeneous sensors is cost-effective and placing them noncolocated with different orientations provides heterogeneity that allows a variety of views of the target and environment. Furthermore, the sensors are usually looking for hard-to-find targets, which generally means only one will appear in a field-of-view within a short time span, or else the field of view would be too large to detect a faint target. Finally, the time scale of encountering hard-to-find targets could be similar to that of dynamically varying environmental changes. Thus, the above assumptions are consequences of phenomena that may naturally occur at a sensor-network node in real-life situations.

IV. Proposed Technical Approach

For collaborative decision-making, it is necessary to construct a framework for information fusion at appropriate levels of granularity or details. From this perspective, the decisions are mostly characterized in terms of information levels at which the fusion takes place. Based on the level of information details, sensor fusion models are classified as the well known triplet: data-level, feature-level, and decision-level [25].

Even though data-level fusion entails maximum details, communication of raw data (i.e., symbols in the proposed approach) may require tight clock synchronization in individual network nodes and high communication overhead, which are expensive to implement. On the other hand, decision-level fusion requires relatively less communication overhead at the expense of loss of vital information, which may lead to incorrect interpretations in a dynamic operating environment. Therefore, the focus of this paper is to formulate an algorithm for feature-level information fusion for target detection.

Each time series Y_i of the observation ensemble $\mathbb{Y}$ is first individually processed (at the sensor site) and then compressed into a low-dimensional feature, as explained in Section II. The symbolic dynamics-based feature extraction method is used to create generative models of the sensed data and also to serve as compact representations of the events that generate the data. Hence, each time-series Y_i for the sensor $\mathcal{L}_i$ is now statistically represented by a unique PFSA.

The set of PFSA, corresponding to the set of sensors $\mathcal{L}$, is denoted as $\mathbb{G} \equiv \{G_1, G_2, \ldots, G_n\}$. The cardinality of the set of states in each PFSA G_i in $\mathbb{G}$ is constrained to be identical (so that they can be compared in the same space). This is further facilitated by having an identical alphabet size $|\Sigma|$ for all PFSA $G_i \in \mathbb{G}$.

Fig. 2 presents a qualitative illustration of the feature evolution in a dynamic environment for a small sensor network. Sensors with local environment-only information (i.e., in the absence of a target) should have similar features at any particular instant of time; however, this information at a given location might be different at two different time instants as well as at different locations at the same time instant. Hence, when compared in the feature space, the sensors with environment-only information should be lumped together as

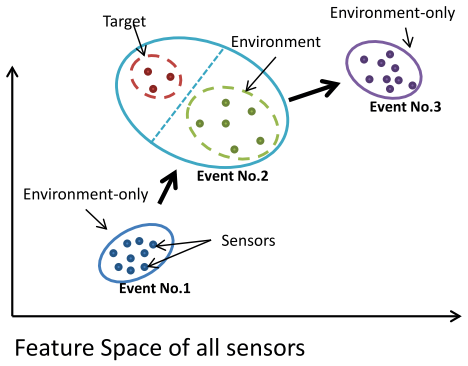


Fig. 2. Feature-space illustration in a dynamic background environment.

a single cluster. If a target is present, response of sensors and thus, the features extracted from the same sensors with target information should be significantly different from the features of sensors with environment-only information. From this perspective, two definitions are introduced to quantify the divergence between the extracted features and the clusters.

Definition 3 (Feature Divergence): Let $\widehat{\mathbf{\Pi}}_i$ be the feature extracted for the PFSA G_i corresponding to the sensor $\mathcal{L}_i$, and let $\widehat{\mathbf{\Pi}}_j$ be the feature extracted for the PFSA G_j corresponding to the sensor $\mathcal{L}_j$. Then, the (scalar) feature divergence between these two sensors is expressed as

$$m\big(\mathcal{L}_i, \mathcal{L}_j\big) \triangleq d\big(\widehat{\mathbf{\Pi}}_i, \widehat{\mathbf{\Pi}}_j\big) \tag{3}$$

where d is a suitable metric and popular choices of a metric [14], [26] are as follows.

1) Euclidean distance, $d_2(a, b) \triangleq (a-b)(a-b)'$.
2) City block distance, $d_1(a, b) \triangleq \sum_{j=1}^{n} |a_j - b_j|$.
3) Cosine distance, $d_c(a, b) \triangleq 1-(ab'/\sqrt{(aa')(bb')})$, where a and b are two row vectors having the same dimension and the column vector a' is the transpose of a.

Definition 4 (Proximity Matrix): Let C_r and C_s be two nonempty clusters, i.e., each of them contains at least one sensor. Let $\mathcal{L}_{ri}$ be the ith sensor in C_r. Let n_r be the number of sensors in C_r. Then, the proximity matrix $A^{\Delta} = [a_{rs}^{\Delta}]$ between any two clusters is obtained elementwise as

$$a_{rs}^{\Delta} \triangleq \Delta(C_r, C_s) \tag{4}$$

where the cluster distance is expressed as

$$\Delta(C_r, C_s) \triangleq \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} m\big(\mathcal{L}_{ri}, \mathcal{L}_{sj}\big). \tag{5}$$

In the following, we present details of three different clustering methods, namely k-means, agglomerative hierarchical, and spectral, that have been used to make collaborative target detection. Each of these three clustering methods has been used to analyze the topological similarity of the PFSA set $\mathbb{G}$ corresponding to the ensemble of observation $\mathbb{Y}$. In all these three methods, we assess the structure of the set $\mathbb{G}$ under the two possible hypotheses, H_0 and H_1. A decision rule is reached by finding the discriminative features in the PFSA set $\mathbb{G}$ for maximum separability of the null and the alternate hypothesis.

The idea is to use the gap statistics [27] of the clusters to learn whether the attributes of the set $\mathbb{G}$ belong to either the environment-only or the target-present scenario.

A. k-Means Clustering

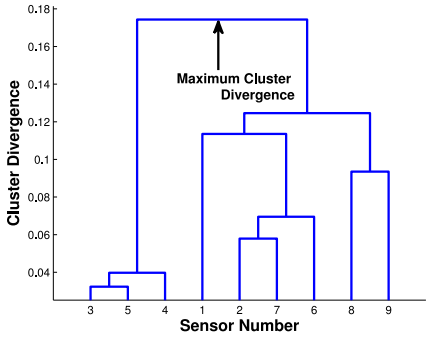
The k-means clustering [28]–[30] is used to group the elements of $\mathbb{G}$ into $k = 2$ mutually exclusive clusters. After initiating the positions of cluster centroids, each G_i is assigned to the closest cluster according to a predefined distance measure (e.g., cosine similarity). The cluster centroids are then updated by minimizing the sum of the distance measures between every G_i and its corresponding cluster centroid. The cluster assignment and distance minimization processes are iteratively repeated until the algorithm converges and there is no further change in assignment of the elements of $\mathbb{G}$ to the clusters. The positions of cluster centroids are initialized by randomizing from the set $\mathbb{G}$. To avoid local minimum due to a bad initialization, ten replicates with individual random initial starts have been used in parallel to ensure that the algorithm returns the lowest sum of distance measures. The outputs of k-means clustering are locations of centroids of the two clusters and element indices of each cluster. The gap (i.e., cluster centroid distance) statistics can be used to design a threshold for declaration of target.

B. Agglomerative Hierarchical Clustering

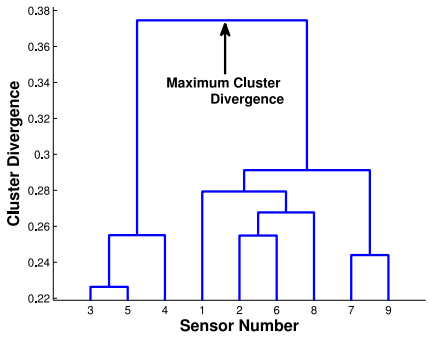
Agglomerative hierarchical clustering [29] is a bottom-up method that generates a sparse network (e.g., a binary tree) of the PFSA set $\mathbb{G}$ by successive addition of edges between the elements of $\mathbb{G}$. Initially, each of the PFSA $G_1, G_2, \ldots, G_n$ is in its own cluster $C_1, C_2, \ldots, C_n$, where $C_i \in \mathcal{C}$, which is the set of all clusters for the hierarchical cluster tree. In terms of the distance measured by (3)–(5), the pair of clusters that are nearest to each other are merged and this step is repeated till only one cluster is left. Fig. 3 presents two examples of binary trees obtained from environment-only and target-present scenarios as an example to make the visualization easier (details of the corresponding experiments are presented in the next section). In both examples, the sensor number 3–5 are grouped together as a single cluster (i.e., they are subjected to high ambient illumination level compared to others). However, the distance between the cluster, consisting of sensor 3–5, and others is significantly larger than that of the environment-only case. This tree structure displays the order of splits in the network, which can be used to find the expected behavior of the network for environment-only and target-present scenarios.

C. Spectral Clustering

Spectral clustering, in general, performs a dimensionality reduction by using spectral properties (i.e., eigenvalues and eigenvectors) of the similarity matrix generated from the feature set (extracted from the sensed data). The PFSA set $\mathbb{G}$ is modeled as a fully connected weighted graph, i.e., every G_i is a node of the graph and the weight for a edge between i and j is calculated from the attributes (or features) of the PFSAs G_i and G_j. In particular, the graph properties are analyzed using two commonly used similarity functions: 1) cosine



(a)



(b)

Fig. 3. Examples of binary tree structure formed by multiple sensors. Binary tree of: (a) environment-only and (b) target-present scenarios.

similarity function $s^c(\cdot,\cdot)$ and 2) the Gaussian kernel similarity function $s^g(\cdot,\cdot)$, both expressed in the following equation:

$$s^c(a_k, a_\ell) \triangleq \frac{a_k a_\ell'}{\|a_k\|\|a_\ell\|} \quad \text{and} \quad s^g(a_k, a_\ell) \triangleq e^{-\frac{\|a_k - a_\ell\|^2}{2\sigma^2}} \tag{6}$$

where the $(1 \times |Q||\Sigma|)$ vectors a_k and a_ℓ are vectorized forms of the matrices $\widehat{\mathbf{\Pi}}_k$ and $\widehat{\mathbf{\Pi}}_\ell$, respectively; the column vector a_ℓ' is the transpose of a_ℓ; the Euclidean norm of the vector a_k is denoted as $\|a_k\|$; and σ is a nonzero real parameter.

Once an $n \times n$ similarity matrix A (i.e., $A^c \triangleq [s^c(a_k, a_\ell)]$ or $A^g \triangleq [s^g(a_k, a_\ell)]$) is constructed, which is symmetric-positive-definite, it is normalized to obtain the graph Laplacian as

$$L = I - M^{-1/2} A M^{-1/2} \tag{7}$$

where M is a positive-definite diagonal matrix and is computed as

$$M_{ii} = \sum_j A_{ij}. \tag{8}$$

By the Rayleigh–Ritz theorem [31], the clustering results are obtained in terms of the $|Q|$-dimensional eigenvector f corresponding to the second smallest eigenvalue of the (positive semi-definite) Laplacian matrix L. The graph partition in clusters is constructed by using the sign of the elements of the eigenvector f as an indicator function, that is

$$\begin{cases} \mathcal{L}_i \in \mathcal{C}_1, & \text{if } f_i \geq 0 \\ \mathcal{L}_i \in \mathcal{C}_0, & \text{if } f_i < 0 \end{cases} \tag{9}$$

where C_1 and C_0 represent the two clusters in the set $\mathbb{G}$.

D. Hypothesis Testing With Feature-Level Sensor Fusion

The two most dissimilar clusters in the PFSA set $\mathbb{G}$ are obtained for each clustering algorithm; for example, see the two clusters at the top of the hierarchical trees in Fig. 3. A decision on the presence of target is made based on the divergence (or gap) between these two clusters. In the presence of a target, the PFSA set $\mathbb{G}$ should be more heterogeneously partitioned into two clusters, one carrying largely the information on the environment-only scenario and the other carrying largely the information on the target-present scenario. It is noted that sensors with environment-only information are always present because the presence of a target is spatially sparse. For the environment-only scenario, even the two maximally separated clusters (e.g., the ones with maximum cluster divergence in the hierarchical cluster tree) should be close to each other in comparison to the target-present scenario. Hence, the maximum cluster divergence for the cluster set $\mathbf{C}_1$ would be higher as compared to that for the set $\mathbf{C}_0$, where the subscripts 0 and 1 represent the null hypothesis H_0 of environment-only (i.e., no target) scenario and the alternate hypothesis H_1 of target-present scenario, respectively. It is concluded from these observations that the maximum cluster divergence, $\Delta(\cdot\,,\cdot)$, which is achieved for the top two clusters in the hierarchical clustering tree, should be strongly influenced by the presence or absence of a target. Spatial sparsity of the event implies that the cluster divergence should be larger in the presence of a target, that is

$$\max_{C_r,\ C_s\in\mathbf{C}_1} \Delta(C_r, C_s) > \max_{C_r,\ C_s\in\mathbf{C}_0} \Delta(C_r, C_s). \tag{10}$$

A decision for target presence could be made with appropriate boundaries to distinguish the cluster gap statistics in the absence and presence of targets. To this end, the distribution of cluster divergence $\Delta(\cdot\,,\cdot)$ (e.g., divergence between the two clusters as seen at the top of the hierarchical tree in Fig. 3) for each hypothesis could be obtained during a training process. Then, the target detection problem is formulated as a binary hypothesis test in terms of the hypothesis pair as

$$\begin{cases} H_0 : X \sim \mathcal{P}_0 \\ H_1 : X \sim \mathcal{P}_1 \end{cases} \tag{11}$$

where X is the range space of the random variable representing the divergence between the two clusters in the PFSA set $\mathbb{G}$; and the probability measures $\mathcal{P}_0$ and $\mathcal{P}_1$ represent the cluster divergence under the null hypothesis H_0 (i.e., environment-only scenario) and the alternate hypotheses H_1 (i.e., target-present scenario), respectively. Decision boundaries can then be obtained by choosing a threshold η on the likelihood ratio Λ, which is obtained for (11) as follows:

$$\Lambda(x) = \frac{d\mathcal{P}_1(x)}{d\mathcal{P}_0(x)} \quad \text{for } x \in X \tag{12}$$

where it is assumed that the probability measure $\mathcal{P}_1$ is absolutely continuous [3] with respect to the probability measure $\mathcal{P}_0$, i.e., for any event E belonging to the σ-algebra of X, if $\mathcal{P}_0(E) = 0$, then it is implied that $\mathcal{P}_1(E) = 0$. Under this condition, the likelihood ratio Λ in (12) is the Radon–Nikodym derivative of $\mathcal{P}_1$ with respect to $\mathcal{P}_0$ and is denoted as $(d\mathcal{P}_1/d\mathcal{P}_0)$.

Sensor networks are designed to satisfy certain performance requirements (e.g., not exceeding a maximum allowable false alarm probability). Under such constraints, the Neyman–Pearson hypothesis testing procedure [3] maximizes the probability of correct detection P_D, while the false alarm probabilities P_F are not allowed to exceed a specified bound α, where $\alpha \in (0, 1)$ is called the significance level. From this perspective, an appropriate choice of the decision rule δ is made to maximize the detection probability as

$$\max_{\delta} P_D(\delta) \text{ subject to } P_F(\delta) \leq \alpha. \tag{13}$$

A threshold η for rejecting the null hypothesis H_0 in the favor of the alternate hypothesis H_1 could be identified by using the following equation:

$$\mathcal{P}_0(\{x \in X : \Lambda(x) > \eta\}) = \alpha. \tag{14}$$

It follows from (13), (14), and the Neyman–Pearson lemma [3] that there exists a unique optimal decision rule $\delta^\star$ with an associated threshold parameter $\eta > 0$, which is dependent on the significance level α, such that:

$$\delta^\star(x) = \begin{cases} \text{Select } H_1 & \text{if } \Lambda(x) > \eta(\alpha) \\ \text{Select } H_0 & \text{if } \Lambda(x) \leq \eta(\alpha) \end{cases} \quad \text{for almost all } x \in X. \tag{15}$$

V. Description of the Experiments

The experimental apparatus consists of a small sensor network, a moving target, and ambient light. The sensor network is a ring of nine TCRT5000 infrared sensors. A computer-instrumented and computer-controlled Khepera III mobile robots [32] serves as a single moving target, the dynamic environment, and the associated disturbances are emulated as variations in the daylight intensities on partially cloudy days.

These sensors embed an infrared light emitter and a receiver. The current experiments make use of the ambient light measurements and the performance of environmental surveillance is inherently dependent on the maximum sensor updating rates. The update time between measurements of all nine sensors is 33 ms. During this 33 ms interval, the nine sensors are read in a sequential way every 3 ms. The central workstation is linked with sensor nodes via Bluetooth. After receiving each new measurement, the station converts and stores the sensor readings before collecting new data. Due to the processing and communication delays, the average interval between two readings is ∼65.3 ms with standard deviation of 5.2 ms, which makes the average updating frequency as ∼18.5 Hz with standard deviation of ∼2.9 Hz.

This paper considers two types of scenarios for multisensor target detection under dynamic environments. Fig. 4 depicts the layout of experimentation. The sensor network is placed at the center of a square room in which only one wall has open windows that are exposed to the sun. During the experiments, the moving target travels at a constant speed in straight lines between the sensor network and the ambient light source. The infrared readings of sensors oriented toward the moving target are subjected to disturbances when the target is moving

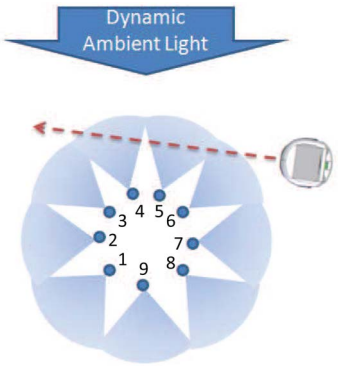


Fig. 4. Experimental setup for target detection in a dynamic environment.

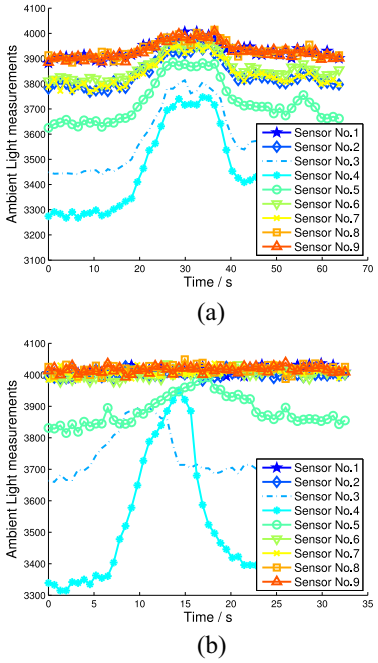


Fig. 5. Typical profiles for collected data. (a) Environment-only (no target). (b) Target-present.

in and out, which causes intermittent blocking of the ambient light source. Fig. 5 presents examples of sensor readings for both environment-only and target-present scenarios.

All experiments have been conducted during days under partially cloudy conditions, where the sunlight is intermittently blocked by clouds. This situation affects the readings of infrared sensors, as depicted in Fig. 5, specially for those sensors that are oriented toward the windows. Sensor nos. 3–5 (that are subjected to stronger ambient illumination) and the remaining sensors, i.e., nos. 1, 2, 6, 7, 8, and 9 (that are subjected to weaker ambient illumination) are respectively grouped together as two single clusters (see the top rows in two plates of Fig. 3). However, although these two clusters appear to be exactly the same for both scenarios, the distance between them is significantly larger in the target-present scenario than that in the environment-only scenario (5). The tree structure in Fig. 3 displays the order of splits in the network, which is gainfully used to identify the behaviorial difference of the network for the afore-said two scenarios.

In total, 50 experiments with environment-only (i.e., absence of a target) and another 50 experiments in the presence of a target have been conducted. For each experiment, nine sensors record synchronized data for about 65 s. Due to the orientation of sensors, the effects of the environment are different for those of the individual sensors. Sensors which are facing the windows (i.e., the ambient light source) would have different levels of reading when compared with the rest of the sensors. When a target is moving in, ambient light sources are partially blocked for some of the sensors, which increases their readings temporarily. As shown in Fig. 5, changes in the ambient light affect the moving target as the readings of some of the sensors fluctuate more significantly. Thus, it becomes difficult to correctly detect a target by simply using a thresholding method on the sensor readings under environmental changes.

VI. Results and Discussion

This section presents the experimental results of multisensor fusion at both decision and feature levels for target detection in a dynamic environment. For decision-level fusion, decisions of target detection are made by voting based on the time series of individual sensors. On the other hand, target detection by feature-level fusion is based on the hypothesis testing of the proposed clustering algorithms. Features are generated from sensor time series based on SDF (see Algorithm 2) with the choice of alphabet size $|\Sigma| = 10$ and PFSA model depth (i.e., memory) $D = 1$.

Time series data, used for target detection, are first normalized by subtracting the mean and then dividing by the standard deviation; this step ensures that the algorithm is not just picking up local spikes in mean or variance for decision-making. For robust performance evaluation of both decision-level and feature-level fusion, cross validation is performed by using multiple combinations of training and test data. For each iteration, the experimental data of both environment-only and target-present cases are randomly divided into training and test data sets: 80% of each scenario (i.e., 40 sets) are assigned to the training set, while the remaining 20% (i.e., 10 sets) are assigned to the test set. The results are generated as the average over 100 such combinations of training and test sets.

In this paper, two different pattern classifiers have been used such as support vector machine (SVM) and k-nearest neighbors (k-NN) that are among the well-known and most frequently used standard tools [33]. The parameter of neighborhood size for k-NN algorithm is chosen as $k = 3$; and a linear kernel is used for SVM classification. The results obtained by usage of these two classifiers could be different due to the inherent structure of the extracted features.

A. Target Detection by Decision-Level Fusion

Decision-level fusion consists of learning and test phases. During the learning phase, each sensor is trained by features extracted from labeled training data. In the testing phase, each sensor reports the results obtained from pattern classification of the features extracted from its test readings. Accordingly, a central workstation declares a positive target detection if

TABLE I
AVERAGE CONFUSION MATRICES OF CROSS VALIDATION FOR DECISION-LEVEL TARGET DETECTION

		k-NN		SVM	
		Target Present	**No Target**	**Target Present**	**No Target**
$N_{tg} \geq 1$	**Target Present**	87.4%	12.6%	92.0%	8.0%
	No Target	29.2%	70.8%	50.8%	49.2%
$N_{tg} \geq 2$	**Target Present**	64.9%	35.1%	63.8%	36.2%
	No Target	16.6%	83.4%	30.8%	69.2%
$N_{tg} \geq 3$	**Target Present**	35.0%	65.0%	26.0%	74.0%
	No Target	9.8%	90.2%	19.4%	80.6%
$N_{tg} \geq 4$	**Target Present**	1.2%	98.8%	3.1%	96.9%
	No Target	5.2%	94.8%	14.2%	85.8%

the total number of sensors reporting the presence of a target satisfies the detection criterion.

The results of target detection with decision-level fusion and using SVM and *k*-NN are shown as confusion matrices in Table I, which shows that both classifiers yield decreasing false alarm rates as the requirement on the total number N_{tg} of sensors reporting the presence of a target is increased. However, the positive detection is severely compromised when the requirement on N_{tg} is greater than 2. As the requirement for correct detection becomes more critical (e.g., $N_{\text{tg}} \geq 4$), the detection performance suffers from low successful detection, even though the near-zero false alarm rate is achieved. The target detection problem, addressed in this paper, is not sensitive to these two different classifiers, SVM and *k*-NN. It appears that the poor performance of decision-level fusion is not dependent on the choice of classifiers.

B. Target Detection by Data-Level Fusion

This section presents the technical approach and results of a data-level fusion scheme for target detection. The data sets used in the training and testing phases are the same as those used for decision-level fusion. However, features are extracted from the ensemble of all sensors instead of being generated individually.

To apply feature extraction on the time series of multiple sensor nodes, a multidimensional symbolization method has been developed. The steps of the feature extraction algorithm are as follows.

1) *Step 1 (Preprocessing):* The time series data of each sensor is transformed as a zero-mean and unit-variance manner (i.e., subtract its mean and then divide by its standard deviation).
2) *Step 2 [MEP (Section II-A)]:* Individual processed time series data are partitioned to obtain the symbol sequence for each sensor.
3) *Step 3 (Application of PCA [33]):* The time series data from multiple sensors are treated as a multidimensional observation vector that is decomposed into linearly independent components using principal component analysis (PCA).
4) *Step 4 (k-Means Clustering [33]):* It is applied on the reduced-dimension PCA feature vectors based on a predefined set of clusters.

TABLE II
AVERAGE CONFUSION MATRICES OF CROSS VALIDATION FOR DATA-LEVEL TARGET DETECTION WITH NUMBER OF CLUSTERS $N_C = 3$

	k-NN		SVM	
Event Description	**Target Present**	**No Target**	**Target Present**	**No Target**
Target Present	90.64%	9.36%	89.52%	10.48%
No Target	10.56%	89.44%	12.24%	87.76%

The symbol sequences are compressed representations of the information contents in the ensemble of time series from all sensor nodes, where the PCA serves the role of dimension reduction. In the analysis based on the original high-dimensional time series ensemble over all experimental data sets, top three principal components are chosen for PCA; these three principal components contribute to more than 95% of the variance (i.e., total energy of all principal components).

A major role of data-level fusion is to obtain the correlation among time series data from different sensor nodes in two different scenarios: 1) no target and 2) target present. The data-level fusion yields better performance relative to the decision-level fusion, because more information is available. In data-level fusion, there are three information compression steps during the symbolization process: 1) partition for individual time series from each sensor; 2) projection on top PCA principle components; and 3) final symbolization from clustering of PCA feature sequence. This information compression could be helpful for communication and computation efficiency, and the reported results are promising for this particular application of target detection under dynamic environments.

The results of target detection have been optimized with appropriate choices of cluster number in the *k*-means clustering algorithm for PCA feature sequence. In this paper, the results are obtained with the number of cluster chosen to be 3. Similar to decision-level fusion, two pattern classification tools, namely SVM and *k*-NN [33], are applied and Table II lists the results data-level fusion.

C. Target Detection by Feature-Level Fusion

This section presents performance evaluation of target detection by feature-level fusion using the three different techniques of clustering, namely *k*-means, agglomerative hierarchical, and spectral, discussed earlier. In each of these approaches,

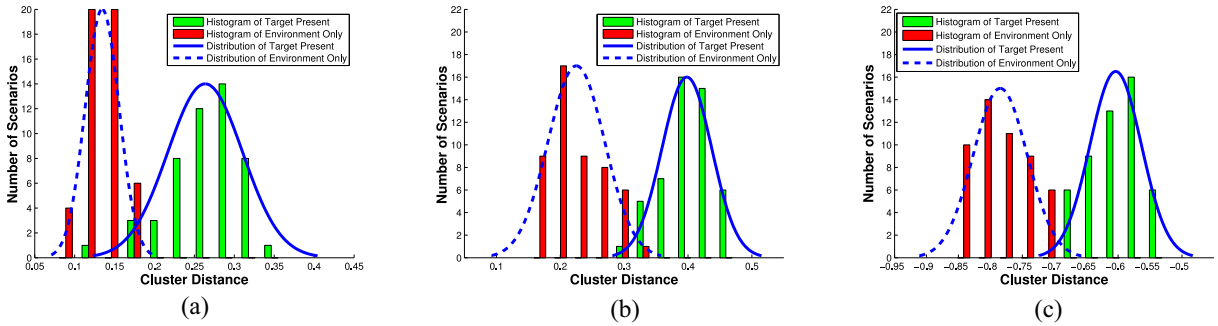


Fig. 6. Distribution of cluster distances for environment-only (H_0) and target-present (H_1) scenarios in the feature space with cosine distance (similarity). (a) k-means clustering. (b) Agglomerative hierarchical clustering. (c) Spectral clustering.

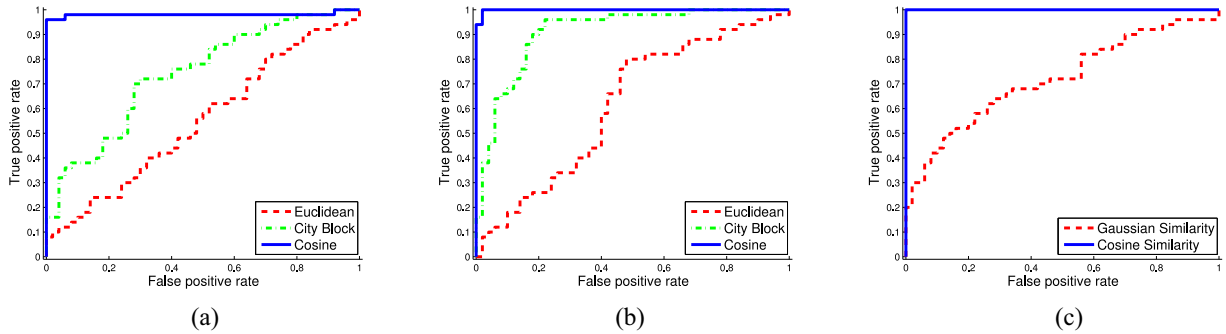


Fig. 7. ROC curves for feature-level target detection. (a) k-means clustering. (b) Agglomerative hierarchical clustering. (c) Spectral clustering.

a decision on target detection is made based on a threshold on the divergence (or gap) between the two most dissimilar (i.e., maximally separated) clusters in the PFSA set $\mathbb{G}$. Since the attributes (i.e., features) of multisensor data are, on the average, more similar to each other in the environment-only scenario than in the target-present scenario, a target is declared to be present if the divergence between the two clusters of interest is higher than a specified discriminating threshold T.

Fig. 7 depicts the receiver operating characteristic (ROC) curves to illustrate the performance of hypothesis testing for target detection using the entire data set. For all three proposed clustering algorithms, the performance of cosine distance (or similarity) measure significantly exceeds those of other feature divergence measures. For Euclidean and city block divergences in k-means and hierarchical clustering, the former provides better true positive rates for the same false positive rate along the entire ROC curve. From the ROC curve obtained for spectral clustering, it is seen that the spectral properties of the Laplacian matrices (with cosine similarity measure) obtained for the features can separate detect targets without any confusion (or error). To see this more clearly, the statistics of cluster divergence (or gap) which are obtained for all three approaches by using the different metrics are approximately fitted with the most-likely Gaussian distribution. The separability of the distributions with cosine distance (similarity) function for the two scenarios are visualized in Fig. 6.

To obtain the probability distributions, first the characteristics of the divergence (or gap) among the members of the PFSA set $\mathbb{G}$ are generated for all the 100 cases using the three clustering techniques. Then, the gap statistics between the two most dissimilar members of $\mathbb{G}$ are obtained and fitted with a Gaussian distribution for both environment-only and target-present scenarios. The distribution obtained for cosine distance (or similarity) for all three clustering methods is distinctly separable. The difference in the mean cluster divergence of the two events is greater than at least twice the standard deviation of the respective distributions, as seen in the histogram and estimated distributions of Fig. 6 for cosine measurement cases.

Both k-means and hierarchical clustering are applied with distance measurements, while spectral clustering uses similarity measurements. Hence, the cluster distance for spectral clustering represents the negative similarity measurements, which are presented as negative real values.

It is observed that the variance parameters in the probability distribution of cluster divergence for both environment-only and the target-present scenarios are approximately similar in most cases (except for the cosine distance function under k-means clustering), and that the mean value for target-present is always larger than that for environment-only. With Gaussian distribution models for both hypotheses H_0 and H_1, it follows from (11) that:

$$\begin{cases} H_0 : X \sim \mathcal{N}(\mu_0, \sigma^2) \\ H_1 : X \sim \mathcal{N}(\mu_1, \sigma^2) \end{cases} \tag{16}$$

where the hypotheses H_0 and H_1 share same variance σ^2 and different expected values $\mu_1 > \mu_0$. It is noted that the variance σ^2 is obtained by averaging the variances of the distributions

TABLE III
AVERAGE CONFUSION MATRICES OF CROSS VALIDATION FOR FEATURE-LEVEL FUSION TARGET DETECTION

		k-means		Hierarchical		Spectral	
		Target Present	**No Target**	**Target Present**	**No Target**	**Target Present**	**No Target**
Euclidean Distance	**Target Present**	20.1%	79.9%	20.7%	79.3%		
	No Target	24.9%	75.1%	20.0%	80.0%		
City Block Distance	**Target Present**	60.0%	40.0%	69.6%	30.4%		
	No Target	1.5%	98.5%	4.9%	95.1%		
Cosine Distance	**Target Present**	100%	0%	100%	0%		
	No Target	1.0%	99.0%	0%	100%		
Gaussian Similarity	**Target Present**					46.3%	53.7%
	No Target					11.6%	88.4%
Cosine Similarity	**Target Present**					100%	0%
	No Target					0%	100%

corresponding to the two hypotheses. Then, the likelihood ratio Λ in (12) is expressed in a closed form as

$$\Lambda(x) = \exp\left(\left(\frac{\mu_1 - \mu_0}{\sigma^2}\right)\left(x - \frac{\mu_1 + \mu_0}{2}\right)\right) \quad \forall x \in \mathbb{R}. \tag{17}$$

Since Λ in (17) is a continuous and strictly increasing function, the decision threshold η for likelihood ratio Λ in (15) can be converted to the decision threshold T for observation x as

$$T = \Lambda^{-1}(\eta). \tag{18}$$

Accordingly, (14) can be reformulated for this particular application as

$$\mathcal{P}_0(\{x > \ T\}) = \alpha \tag{19}$$

and the parameter T is computed for Gaussian distribution $\mathcal{N}(\mu_0, \sigma^2)$ as a function of the significance level α as

$$T(\alpha) = \sigma \ Q^{-1}(\alpha) + \mu_0 \text{ for } \ \alpha \in (0, 1) \tag{20}$$

where $Q(\theta) \triangleq (1/\sqrt{2\pi}) \int_\theta^\infty dt \ e^{-(t^2/2)} \quad \forall \theta \in \mathbb{R}$.

It follows from (19) that an α-level Neyman–Pearson test for this case yields the optimal decision rule as:

$$\delta^\star(x) = \begin{cases} \text{Select } H_1 & \text{if } x > T(\alpha) \\ \text{Select } H_0 & \text{if } x \leq T(\alpha). \end{cases} \tag{21}$$

To evaluate the robustness of the feature-level fusion, cross validation is performed by different combinations of training and test data. In the training phase, an ROC curve is constructed from 80 cases (40 from each of the environment-only and target-present scenarios) of training data. The threshold T is obtained from (20) so that the false-alarm probability does not exceed the assigned significance level $\alpha = 0.05$ for Neyman–Pearson criterion [3]. Then, this threshold value is used for decision-making in the remaining 20 cases for both scenarios to compute the confusion matrices in the test phase. In total, 100 different combinations of training and test data are used and the results are summarized in Table III.

While an arbitrary classifier could have chosen wildly different decision boundaries in a dynamic environment, the proposed method achieves stable decision boundaries with multisensor fusion. The results of cross validation show that the performance of cosine distance (similarity) measure is consistently good regardless of the choice of the clustering algorithm. The false positive rates of cosine similarity in the test phase are all below 5% as required, while other measures fail to meet this criterion. The true positive rate for cosine measures are above 95% for all clustering algorithms, while the other metrics show a rather poor performance for all clustering algorithms.

TABLE IV
CLUSTER PURITY FOR TARGET-PRESENT SCENARIOS

Feature Measure	k-means	Hierarchical	Spectral
Euclidean Distance	84.44%	80.00%	
City Block Distance	85.11%	86.44%	
Cosine Distance	86.67%	87.11%	
Gaussian Similarity			84.22%
Cosine Similarity			88.00%

Table IV presents the results of cluster purity [34] for 50 cases with a moving target. Due to spatial sparsity of the moving target, only a few sensors can detect targets while the rest collect environment-only data. The nine sensors are clustered into two groups as: 1) environment-only and 2) target-present. Then, the clustering purity is measured by calculating the correct grouping of sensors in the two clusters. Among all metrics (i.e., distance and similarity measures) used, the cosine distance (similarity) measure provides the highest cluster purity for all three clustering algorithms. On an average, all three algorithms misclassify only one or two sensor(s) out of nine for each event. However, the performance difference is not that profound under different combinations of similarity measures and clustering algorithms.

The comparison between the cosine similarity measure and a typical distance metric (e.g., Euclidean) has been reported in the open literature [26], [35]. Although the characteristics of feature vectors could be more favored by one measure than the other [36], cosine measures usually outperform Euclidean measure [37], [38] in clustering problems. The reason is that distance metrics (e.g., Euclidean and City Block) are measures of the distance between a pair of points in the space, while cosine similarity is a measure of the angle between the pair of vectors. For the clustering purpose, cosine similarity provides the capability of distinguishing changes in the direction of the feature vector in contrast to the absolute numerical difference of individual elements of the feature vector.

VII. Conclusion

This paper presents a generalized framework for robust multisensor target detection in dynamic environments, where low-dimensional features are extracted from different sensors by making use of SDF [14], [15]. A data-driven method is proposed for sensor fusion, which does not rely on availability of sensor and process models. The tool of SDF [14] is applied for feature extraction from time-series data, collected from different sensors. The structure of the set of features is then assessed using three different clustering methods while using different metrics for them. Due to a dynamic background environment, the structure of each of the clustered features changes at different time instants; however, within the time span of an event of interest, all sensors share the same environment dynamics. Under this observation, a decision on the presence of a target is made by first using the divergence between the two most dissimilar clusters and then by learning the cluster gap statistics for the binary hypothesis during the training process. It is also shown how to use different criteria (e.g., maximum false-alarm probabilities) to calculate a threshold for the deciding rule.

Performance of the proposed target detection method is experimentally evaluated by hypothesis testing in a laboratory setting for a single moving target, where detection is made by multiple infrared sensors under fluctuating ambient light. Results of experimentation show that the proposed feature-level method outperforms a supervised decision-level fusion scheme in uncertain dynamic environments. The robustness of the performance is evaluated by a cross validation process and the results show perfect positive detection with low false alarm probabilities. The decision boundaries have been obtained under the constraints of maximum allowable false alarm probabilities by applying Neyman–Pearson decision criterion [3]. The results (i.e., the decision boundaries) are found to be stable with (almost) perfect accuracy for detection of targets with appropriate choices of the distance function and similarity matrix (i.e., cosine distance and cosine similarity). Pertinent conclusions from the work reported in this paper are summarized as follows.

1) The feature-level clustering scheme provides a semi-supervised robust framework for target detection in a dynamic background environment with limited communication capability.
2) SDF provides a reliable low-complexity feature-extraction tool that can be used for information compression and fusion at the nodes (with limited computation capabilities) of a sensor network.
3) The target detection method is robust relative to exact placement of the sensors (i.e., there are no restrictions imposed on sensor placement).

While there are several issues that need to resolved by further theoretical and experimental research, the following topics are recommended for future research.

1) *Sequential Hypothesis Testing:* A framework is required for more complicated scenarios of target classification when multiple homogeneous and/or heterogeneous targets are present.
2) *Enhancement of Feature Extraction:* The parameters (e.g., alphabet size and depth for stochastic modeling for PFSA construction) of SDF need to be optimized.
3) *Enhanced Experimental Validation:* Further experimental work on a large-scale sensor network would increase the credibility of the proposed method of target detection in different applications.

Acknowledgment

Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[1] X. Wang, S. Wang, and D. Bi, "Distributed visual-target-surveillance system in wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 5, pp. 1134–1146, Oct. 2009.
[2] D. Gu, "A game theory approach to target tracking in sensor networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 1, pp. 2–13, Feb. 2011.
[3] H. V. Poor, *An Introduction to Signal Detection and Estimation*. New York, NY, USA: Springer-Verlag, 1988.
[4] Z. Yuan, H. Xue, Y. Caon, and X. Chang, "Exploiting optimal threshold for decision fusion in wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 2014, 2014, Art. ID 506318.
[5] M. Zhu *et al.*, "Fusion of threshold rules for target detection in wireless sensor networks," *ACM Trans. Sensor Netw. (TOSN)*, vol. 6, no. 2, 2010, Art. ID 18.
[6] W. Liu, Y. Lu, and J. Fu, "Threshold optimization algorithm for weak signal in distributed-sensor fusion system," *Signal Process.*, vol. 82, no. 10, pp. 1329–1336, 2002.
[7] M. F. Duarte and Y.-H. Hu, "Vehicle classification in distributed sensor networks," *J. Parallel Distrib. Comput.*, vol. 64, no. 7, pp. 826–838, 2004.
[8] A. Tavakoli, J. Zhang, and S. H. Son, "Group-based event detection in undersea sensor networks," in *Proc. 2nd Int. Workshop Netw. Sensing Syst.*, San Diego, CA, USA, Jun. 2005.
[9] X. Jin, S. Gupta, K. Mukherjee, and A. Ray, "Wavelet-based feature extraction using probabilistic finite state automata for pattern classification," *Pattern Recognit.*, vol. 44, no. 7, pp. 1343–1356, 2011.
[10] X. Jin, S. Sarkar, A. Ray, S. Gupta, and T. Damarla, "Target detection and classification using seismic and PIR sensors," *IEEE Sensors J.*, vol. 12, no. 6, pp. 1709–1718, Jun. 2012.
[11] A. Y. Yang *et al.*, "Distributed segmentation and classification of human actions using a wearable motion sensor network," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Anchorage, AK, USA, 2008, pp. 1–8.
[12] T. Clouqueur, K. K. Saluja, and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," *IEEE Trans. Comput.*, vol. 53, no. 3, pp. 320–333, Mar. 2004.
[13] Y. Li, D. K. Jha, A. Ray, and T. A. Wettergren, "Feature level sensor fusion for target detection in dynamic environments," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, Jul. 2015, pp. 2433–2438.
[14] A. Ray, "Symbolic dynamic analysis of complex systems for anomaly detection," *Signal Process.*, vol. 84, no. 7, pp. 1115–1130, Jul. 2004.
[15] K. Mukherjee and A. Ray, "State splitting and merging in probabilistic finite state automata for signal representation and analysis," *Signal Process.*, vol. 104, pp. 105–119, Nov. 2014.
[16] Y. Bar-Shalom, F. Daum, and J. Huang, "The probabilistic data association filter," *IEEE Control Syst.*, vol. 29, no. 6, pp. 82–100, Dec. 2009.
[17] N. F. Sandell and R. Olfati-Saber, "Distributed data association for multi-target tracking in sensor networks," in *Proc. 47th IEEE Conf. Decis. Control*, Cancún, Mexico, 2008, pp. 1085–1090.
[18] F. Darema, "Dynamic data driven applications systems: New capabilities for application simulations and measurements," in *Proc. 5th Int. Conf. Comput. Sci. (ICCS)*, Atlanta, GA, USA, 2005, pp. 610–615.
[19] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd ed. New York, NY, USA: ACM, 2001, pp. 45–138.

[20] M. Sipser, *Introduction to the Theory of Computation*, 3rd ed. Boston, MA, USA: Cengage, 2013.
[21] V. Rajagopalan and A. Ray, "Symbolic time series analysis via wavelet-based partitioning," *Signal Process.*, vol. 86, no. 11, pp. 3309–3320, Nov. 2006.
[22] G. Mallapragada, A. Ray, and X. Jin, "Symbolic dynamic filtering and language measure for behavior identification of mobile robots," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 42, no. 3, pp. 647–659, Jun. 2012.
[23] C. R. Shalizi and K. L. Shalizi, "Blind construction of optimal nonlinear recursive predictors for discrete sequences," in *Proc. 20th Conf. Uncertainty Artif. Intell. (AUAI)*, 2004, pp. 504–511.
[24] D. K. Jha, A. Srivastav, K. Mukherjee, and A. Ray, "Depth estimation in Markov models of time-series data via spectral analysis," in *Proc. Amer. Control Conf.*, Chicago, IL, USA, 2015, pp. 5812–5817.
[25] B. V. Dasarathy, "Sensor fusion potential exploitation-innovative architectures and illustrative applications," *Proc. IEEE*, vol. 85, no. 1, pp. 24–38, Jan. 1997.
[26] S.-S. Choi, S.-H. Cha, and C. C. Tappert, "A survey of binary similarity and distance measures," *J. Syst. Cybern. Informat.*, vol. 8, no. 1, pp. 43–48, 2010.
[27] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. Roy. Stat. Soc. B (Stat. Methodol.)*, vol. 63, no. 2, pp. 411–423, 2001.
[28] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probabil.*, vol. 1. Berkeley, CA, USA, 1967, pp. 281–297.
[29] R. Xu and D. Wunsch, II, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005.
[30] D. Arthur and S. Vassilvitskii, "*k*-means++: The advantages of careful seeding," in *Proc. 8th Annu. ACM-SIAM Symp. Disc. Algorithms*, Minneapolis, MN, USA, 2007, pp. 1027–1035.
[31] H. Lütkepol, *Handbook of Matrices*. New York, NY, USA: Wiley Intersci., 1997.
[32] KhaperaRobot, *Khepera III User Manual Version 2.1*. K-Team SA, Vallorbe, Switzerland, 2008.
[33] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). New York, NY, USA: Springer-Verlag, 2006.
[34] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, vol. 1. Cambridge, U.K.: Cambridge Univ. Press, 2008.
[35] T. Korenius, J. Laurikkala, and M. Juhola, "On principal component analysis, cosine and Euclidean measures in information retrieval," *Inf. Sci.*, vol. 177, no. 22, pp. 4893–4905, Nov. 2007.
[36] G. Qian, S. Sural, and S. Pramanik, "A comparative analysis of two distance measures in color image databases," in *Proc. Int. Conf. Image Process.*, vol. 1. Rochester, NY, USA, 2002, pp. I-401–I-404.
[37] A. Huang, "Similarity measures for text document clustering," in *Proc. 6th New Zealand Comput. Sci. Res. Student Conf. (NZCSRSC2008)*, Christchurch, New Zealand, 2008, pp. 49–56.
[38] A. Strehl, J. Ghosh, and R. Mooney, "Impact of similarity measures on Web-page clustering," in *Proc. Workshop Artif. Intell. Web Search (AAAI)*, Austin, TX, USA, 2000, pp. 58–64.

Yue Li received the B.S. degree in mechanical engineering from Shanghai Jiao Tong University, Shanghai, China, in 2010. He is currently pursuing the Ph.D. degree with the Department of Mechanical and Nuclear Engineering, Pennsylvania State University, State College, PA, USA.

He is enrolled in the master's program in mathematics with Pennsylvania State University. His current research interests include signal processing and machine learning for applications to battery systems and sensor networks.

Devesh K. Jha (S'13) received the B.E. degree in mechanical engineering from Jadavpur University, Kolkata, India, in 2010. He is currently pursuing the Ph.D. degree with the Department of Mechanical and Nuclear Engineering, Pennsylvania State University, State College, PA, USA, where he is also pursuing the master's degree in mathematics.

His current research interests include machine learning, time series analysis, and robotics.

Mr. Jha is a Student Member of the American Society of Mechanical Engineers.

Asok Ray (SM'83–F'02) received the graduate degrees in electrical engineering, mathematics, and computer science, and the Ph.D. degree in mechanical engineering, all from Northeastern University, Boston, MA, USA.

He joined the Pennsylvania State University, State College, PA, USA, in 1985, where he is currently a Distinguished Professor of Mechanical Engineering and Mathematics, and a Graduate Faculty Member of Electrical Engineering and Nuclear Engineering. He has authored or co-authored over 600 research publications, including over 280 scholarly articles in refereed journals and research monographs.

Dr. Ray is a fellow of the American Society of Mechanical Engineers and World Innovative Foundation.

Thomas A. Wettergren (A'95–M'98–SM'06) received the B.S. degree in electrical engineering and the Ph.D. degree in applied mathematics from Rensselaer Polytechnic Institute, Troy, NY, USA.

He joined the Naval Undersea Warfare Center, Newport, RI, USA, in 1995, where he has served as a Research Scientist with the Departments of Torpedo Systems, Sonar Systems, and Undersea Combat Systems. He currently serves as the U.S. Navy Senior Technologist for Operational and Information Science as well as a Senior Research Scientist with the Newport Laboratory, Worthington, MN, USA.

Dr. Wettergren is a member of the Society for Applied and Industrial Mathematics.