# Buried Underwater Object Classification Using a Collaborative Multiaspect Classifier

Jered Cartmill, *Member, IEEE*, Neil Wachowski, and Mahmood R. Azimi-Sadjadi, *Senior Member, IEEE*

***Abstract*—In this paper, a new collaborative multiaspect classification system (CMAC) is introduced, which utilizes a group of collaborative decision-making agents capable of producing a high-confidence final decision based on features obtained over multiple aspects. It is also shown how CMAC can be modified to perform multiaspect classification using a decision feedback (DF) strategy. The system is then applied to a buried underwater target classification problem. The results show that CMAC provides excellent multiple-ping classification of mine-like objects while reducing the number of false alarms compared to other multiple-ping classification fusion systems such as nonlinear decision-level fusion (DLF).**

***Index Terms*—Bayes classification, buried object scanning sonar system, collaborative decision making, underwater target classification.**

## I. Introduction

CLASSIFICATION of buried underwater objects is challenging owing to several issues that include: variability of target signatures and features with respect to the aspect angle and range of the sonar, presence of competing natural and man-made clutter, bottom reverberation effects, and lack of any *a priori* knowledge about the shape and geometry of abundant nonmine-like objects that can be encountered on the seafloor. Furthermore, variations in the environmental conditions complicate this problem. In order to overcome these challenges, it is desirable to devise feature extraction and target detection and classification methodologies that remain robust to these effects and that can screen the entire data set for potential mine-like objects.

For many orientations, a mine-like object may possess nearly identical sonar characteristics to those of some nonmine-like objects at a single ping. Consequently, in real-life situations, decision about the presence and type of object is made based upon the observation of the properties of the received signals over several sonar pings. Typically, two approaches are used to perform multiple-ping classification. In the first approach, features are extracted across multiple pings in order to characterize common signatures that indicate if an object has mine-like or nonmine-like properties. One way to accomplish this is to use canonical correlation analysis (CCA) [1], which allows coherence-based features to be extracted from pairs of sonar pings. Using these coherence-based features allows for successful discrimination between mine-like and nonmine-like objects [2], [3].

The second approach is classifier based and can be performed using either decision-level fusion (DLF) [4], feature-level fusion [5]–[9], or a combination of feature-level fusion and DLF [10]. In DLF [4], intermediate decisions obtained using a single-ping classifier are fused to yield a final decision. A disadvantage of the DLF is its inflexibility to use a variable number of pings when making a final decision, rendering it unsuitable for use in real-world applications. In feature-level fusion [5]–[9], the class of an object is chosen based on the corresponding sequence of features obtained over multiple pings. Feature-level fusion has successfully been applied to underwater target classification [5]–[9], employing hidden Markov model (HMM) framework. In one such system [5], HMMs are used to find the most likely identity of an object that produced a sequence of observations (feature vectors) based on the transitions from a group of unobservable states. This sequence of features is then applied to two models corresponding to the mine-like and nonmine-like classes, and the model that yields the highest probability of that state sequence decides the class membership. To form a more discriminant and robust classifier, the HMM is combined with a neural network used to estimate the emission probabilities [5]. Another HMM-based feature-level fusion scheme [6], [7] defines the states in an HMM to be target-sensor orientations for which the features are highly correlated. The features are found using a wave-based matching pursuit algorithm [7]. In a wavelet-based feature-level fusion scheme [8], the coefficients obtained by applying a wavelet transform to each signal (corresponding to a target-sensor orientation) form the states in a Markov process. An "outer" HMM accounts for target-sensor angular motion and an "inner" HMM characterizes the statistics of the wavelet coefficients within a state of the outer HMM. The inner HMM model feeds information to the outer HMM model, which in turn makes the final decisions. This dual-HMM algorithm proved [8] to be relatively robust to both colored additive noise as well as timing jitter. Another study [9] considered backscattered signals observed from a sequence of target-sensor orientations, and used an adaptive search procedure to choose the angular displacement between consecutive measurements. In this algorithm, targets can be recognized more accurately and efficiently using fewer but better-chosen measurements. A common disadvantage of feature-level fusion systems [5]–[9] is that they can be very difficult to train adequately, especially if a limited amount of data is available, or when a large variety of mine-like and nonmine-like objects exists.

Manuscript received October 31, 2007; revised July 21, 2008; accepted October 13, 2008. First published March 10, 2009; current version published March 20, 2009. This work was supported by the Office of Naval Research (ONR) 32MCM under Contracts #N00014-05-1-0014 and #N00014-07-1-0542.

**Associate Editor: J. Potter**.

J. Cartmill was with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA. He is now with the Argon ST, Fairfax, VA 22033 USA (e-mail: jered.cartmill@gmail.com).

N. Wachowski and M. R. Azimi-Sadjadi are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA (e-mail: nswachow@engr.colostate.edu; azimi@engr.colostate.edu).

Digital Object Identifier 10.1109/JOE.2008.2008041

Multiple-ping classification can also be performed using a combination of feature-level fusion and DLF [10]. In this system, the class label is determined using not only the current feature vector, but also the decisions made at the previous pings. Unfortunately, the performance of such a system is somewhat hindered by the overlap in the feature space for some of the objects, as the mechanism used to provide evidence to the system may perform poorly if the feature space is tightly packed. Finally, the cost functions used by this system and the DLF system do not allow for collaboration in the decision-making process among the decision-making agents that produce the intermediate decisions used by these systems, hence resulting in loss of potentially valuable information.

Clearly, a useful system for performing multiple-ping classification would be one in which a group of decision makers collaborate by sharing information regarding their preliminary decisions in order to reach a high-confidence final decision. This system would be even more valuable if it did not suffer from any of the aforementioned shortcomings of the other multiple-ping classification fusion systems. The development of such a system for buried underwater object classification is precisely the goal of this work.

In this study, a new classification system known as the collaborative multiaspect classifier (CMAC) is introduced. Unlike the DLF system, decision collaboration among different agents is made before generating the final decision. This process provides additional discriminatory information that is exploited by the agents to produce a high-confidence final decision. The proposed system is simple to implement and can easily incorporate a variable number of pings to determine the final class label of an object, and hence provides a very versatile means of performing multiple-ping classification fusion. Additionally, CMAC can be configured in such a way to perform a decision feedback (DF)-based multiaspect classification fusion. The newly developed CMAC system is benchmarked against the DLF system using a data set collected by the buried object scanning sonar (BOSS) system for performing multiple-ping classification of buried underwater objects.

This paper is organized as follows. Section II introduces the new CMAC and gives details on its formulation and structure. Section III reviews the BOSS system used to collect the data of this study, as well as the properties of the collected data. The preprocessing and feature extraction methods and a discussion on the extracted features are also provided. Section IV provides the results produced using CMAC and benchmarking with those of other multiaspect classification systems, namely, a nonlinear DLF classifier and a DF classifier. This section also presents the detection and classification results of each system on the data of entire runs through the target field, along with discussions on the advantages and disadvantages of each system. Finally, Section V provides concluding remarks on this study.

## II. A Collaborative Multiaspect Classification System

CMAC is a new multiaspect/ping classification system utilizing a set of collaborating agents, which share information with each other before making a final decision. The development of the CMAC system is motivated by its collaborative ability to minimize a cost function based on overall misclassifications. This property is not shared by any of the other multiple-ping classifiers [4]–[10]. CMAC is inspired from the distributed detection method developed in [11] for sensor networks. The system in [11] also utilizes multiple decision-making agents that collaborate with each other in order to take advantage of each agent's unique knowledge of the environment. The CMAC system, however, extends the method in [11] to collaborative classification problems. Among the extensions are the following: using a pattern vector as the input to each agent instead of a continuous scaler, decoupling the preliminary and final decision rules by using a probabilistic neural network (PNN) as a classifier, and adding a backpropagation neural network (BPNN) for estimating the class conditional probabilities necessary for calculating a threshold used in generating the final decision. Additionally, substantial changes to the formulation have been made to apply the distributed detection idea in [11] to multiaspect collaborative classification problems.

### A. CMAC System Overview

In the proposed CMAC system [shown in Fig. 1(a)], a group of $N$ decision-making agents is used to produce $N$ separate decisions regarding the class membership of $N$ pattern vectors (observations) $\mathbf{x}_i$, $i \in [1, N]$. This is accomplished in the following manner. First, an agent $i$, whose internal structure is shown in Fig. 1(b), makes a preliminary classification decision $u_i$, using a two-class PNN classifier [12] based on the pattern available to the agent. The purpose of preliminary decision $u_i$ is to assist the other $N-1$ agents in making their final decisions. These preliminary decisions are continuous-valued scalars in the range of $[0, 1]$ that represent the confidence of an agent regarding the class membership of its observed pattern (feature) vector.

Each agent then shares this information with other agents in the group via a "coordinator," whose purpose is to facilitate the sorting and dissemination of the appropriate preliminary decisions to and from each agent. Upon the receipt of the $u_i$, $i \in [1, N]$, from all the agents, the coordinator generates $N$ vectors $\mathbf{u}_{ir} = [u_1, \cdots, u_{i-1}, u_{i+1}, \cdots u_N]^T$, $i \in [1, N]$, which consist of the decisions $u_j, j \in [1, N], j \neq i$, of all other agents. Note that the index term $ir$ is used to denote that this vector will be "received" by the $i$th agent for use in its final decision generation. Once these vectors are formed, the coordinator subsequently provides each to the appropriate agent's probability estimator [shown in Fig. 1(b)], which evaluates each preliminary decision in $\mathbf{u}_{ir}$ separately to produce estimates of the class conditional probabilities necessary for calculating the threshold used in the final decision making of each agent. These class conditional probabilities $p(u_j|C_k)$, $j \in [1, N]$, $j \neq i$, $k \in \{0, 1\}$, are then passed to the agent's data fusion center, which in turn renders a final decision $u_{fi}$ about the class of its input pattern $\mathbf{x}_i$. This final decision at the $i$th agent is made based on a likelihood ratio test [11], [13] that utilizes not only the class conditional probabilities provided by the other agents, but also the preliminary decision made on the input pattern $\mathbf{x}_i$.

### B. Final Decision Rule Formulation

In order to determine the optimal final decision rule for each agent, our goal is to minimize the overall expected cost of mis-

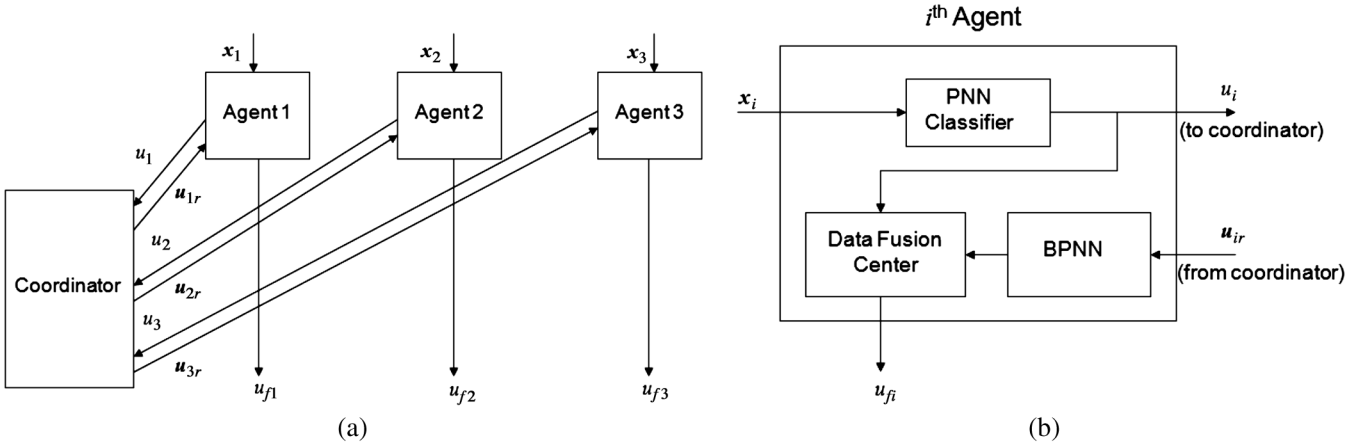


Fig. 1. (a) CMAC system ($N = 3$ case). (b) Internal structure of the $i$th agent.

classification for each agent. Several assumptions are made in our formulation. The first assumption is that the set of input vectors $\{\mathbf{x}_i, i \in [1, N]\}$ is conditionally independent given the true class $C_k$, i.e.,

$$p(\mathbf{x}_1, \ldots, \mathbf{x}_N | C_k) = \prod_{i=1}^{N} p(\mathbf{x}_i | C_k) \tag{1}$$

where $p(\mathbf{x}_i|C_k)$ is the *a priori* class conditional density. We also assume that based only on its observation $\mathbf{x}_i$, the $i$th agent makes a single local preliminary decision $u_i$ using

$$u_i = \gamma_i(\mathbf{x}_i), \qquad i \in [1, N] \tag{2}$$

where $\gamma_i(\cdot)$ is the mapping function of the PNN classifier that captures the decision rule. To obtain a final decision $u_{fi}$, a decision rule $\gamma_{fi}(u_i)$ is also used such that

$$u_{fi} = \gamma_{fi}(\mathbf{x}_i, \mathbf{u}_{ir}) \tag{3}$$

where $\mathbf{u}_{ir}$ denotes the set of preliminary decisions of all other agents (except the $i$th one) used by the $i$th agent's fusion center obtained via the coordinator.

The cost incurred in making the classification decision $u_{fi} = m$ for the $i$th agent is denoted by $J_{mk}$, where $k$ is the true class, $m, k \in \{0, 1\}$, for two class problems. We assume that the cost of making an incorrect classification is greater than the cost of making a correct classification, i.e.,

$$J_{m=n\ k} \geq J_{m=k\ k}, \qquad n \in \{0, 1\}. \tag{4}$$

Also, we assume that for some $\epsilon > 0$

$$p\left(|u_{fi} - u_l| \leq \epsilon \middle| \mathbf{u}_{ir}, \mathbf{x}_i\right) \geq p\left(|u_{fi} - u_l| > \epsilon \middle| \mathbf{u}_{ir}, \mathbf{x}_i\right), \quad i, l \in [1, N], \qquad l \neq i \tag{5}$$

where $u_l$ is the $l$th element of $\mathbf{u}_{ir}$. Essentially, (5) states that when the final decision made by the $i$th agent is compared to the preliminary decision made by the $l$th agent, it is more likely that they will agree than disagree. This makes sense in light of the fact that each agent makes a decision based on a different aspect of the same object, so the decisions are more likely to be consistent with each other.

Using these assumptions, the problem is to obtain the optimum strategies $\gamma_{fi}, i \in [1, N]$, in (3) to minimize the expected cost of making a misclassification for each of the agents, i.e.,

$$\text{Min}\left\{E\left[J\left\{\gamma_{fi}(\mathbf{x}_i, \mathbf{u}_{ir}), C_k\right\}\right] = J_{ik}\right\} \quad \text{subject to (4) and (5).} \tag{6}$$

Solving (6) leads to the following optimal decision rule for the $i$th agent, $i \in [1, N]$. At the $i$th agent's data fusion center, the final decision $u_{fi}$ is produced using the likelihood ratio test

$$\Lambda(\mathbf{x}_i) = \frac{p(\mathbf{x}_i|C_1)}{p(\mathbf{x}_i|C_0)} \underset{u_{fi}=0}{\overset{u_{fi}=1}{\gtrless}} t_{fi} \tag{7}$$

where $\Lambda(\mathbf{x}_i)$ is the likelihood ratio and the threshold $t_{fi}(\mathbf{u}_{ir})$ is found using (see the Appendix)

$$t_{fi}(\mathbf{u}_{ir}) = \frac{P(C_0) \prod_{j=1, j\neq i}^{N} p(u_j|C_0)[J_{10} - J_{00}]}{P(C_1) \prod_{j=1, j\neq i}^{N} p(u_j|C_1)[J_{01} - J_{11}]} \tag{8}$$

where $p(u_j|C_k), k \in \{0, 1\}$, is the conditional density function of generating a preliminary decision $u_j$ when the input pattern belongs to class $k$, $P(C_k)$ is the prior probability that a given feature vector belongs to class $k$, $J_{01}$ and $J_{10}$ are the cost of making wrong decisions when the true classes are $C_1$ and $C_0$, respectively, and $J_{11}$ and $J_{00}$ are the corresponding costs for making the right decisions.

In (8), the dependence on $\mathbf{u}_{ir}$ is implicitly assumed because the probability density function of the preliminary decisions $u_j$, $j \in [1, N]$, $j \neq i$, is directly obtained using the elements of $\mathbf{u}_{ir}$. Thus, to generate the final decision $u_{fi}, i \in [1, N]$, for each agent, one needs to compute $\Lambda(\mathbf{x}_i)$ and $t_{fi}(\mathbf{u}_{ir})$. The latter can be computed using a density estimator such as a BPNN trained to estimate the conditional density functions $p(u_j|C_k)$ of the preliminary decisions of the PNN decision maker, which are the same for all agents. This is discussed in Section II-D. The

likelihood term $\Lambda(\mathbf{x}_i)$ is generated by the PNN classifier as will be discussed in Section II-C.

One of the interesting benefits of the cost function (6) used in deriving the final decision rule in the CMAC system is that it minimizes the average risk of misclassification for a given decision maker by taking into account the preliminary decisions (i.e., $\mathbf{u}_{ir}$) produced by the other $N-1$ decision makers. This implies that the system is able to use collaboration among the agents to reduce the chance of generating an incorrect final decision. In contrast, in other multiple-ping fusion systems [4], [5], [10], no collaborative decision making takes place, either implicitly or explicitly, among the decision-making agents. The lack of collaboration prevents certain discriminatory evidence about the relationship among the feature vectors to be used, and hence stymies these systems' classification performance.

### C. PNN-Based Decision Maker (Classifier)

In CMAC, the classifier is a PNN [12], which implements the Parzen nonparametric probability density function estimation and Bayes classification rule. Note that the PNN used by each agent in the system is identical, so only one PNN needs to be trained. The PNN consists of three feedforward layers: the input layer, the pattern layer, and the summation layer [12]. Feature vectors are applied to the input layer, which passes them to each neuron in the pattern layer. The pattern layer consists of $K$ pools of pattern neurons, where $K$ is the number of classes. In each pool $k \in [0, K-1]$, there are $N_k$ pattern neurons, each of which represents exactly one pattern from the training set for class $C_k$. For the input feature vector $\mathbf{x}$ with dimension $d$, the output of each pattern layer neuron is

$$f\left(\mathbf{x}; \mathbf{w}_k^{(j)}, \sigma\right) = \frac{1}{N_k (2\pi)^{d/2} \sigma^d} \times \exp\left[-\frac{\left(\mathbf{x}-\mathbf{w}_k^{(j)}\right)^T \left(\mathbf{x}-\mathbf{w}_k^{(j)}\right)}{2\sigma^2}\right] \quad (9)$$

where $\mathbf{w}_k^{(j)}$ is the weight vector of the $j$th neuron in the $k$th pool, and the nonlinear function $f(\cdot)$ represents the activation functions of the neurons. In the summation layer, the $k$th neuron associated with class $C_k$, $k \in [0, K-1]$, forms the weighted sum of all the outputs from the neurons in the $k$th pool in the pattern layer. The weights in the summation layer are determined by the decision cost function and the *a priori* class distribution. For the "0-1" cost function and equally likely classes, the weights will be one for all the neurons in the summation layer. For the input pattern $\mathbf{x}$ of an unknown class, a final decision is made through a simple comparison of the PNN outputs $O_k(\mathbf{x})$, i.e.,

$$\mathbf{x} \in C_k, \quad \text{if } O_k(\mathbf{x}) > O_i(\mathbf{x}), \qquad i \in [0, K-1], \qquad k \neq i. \quad (10)$$

Under certain conditions [12], the outputs of the PNN correspond to the *a posteriori* conditional probabilities, i.e., $O_k(\mathbf{x}) \approx p(C_k|\mathbf{x})$, when $\mathbf{w}_k^{(j)} = \mathbf{x}_k^{(j)}$, i.e., the weight vector of the $j$th neuron in pool $k$ is set to the training sample $\mathbf{x}_k^{(j)}$ belonging to class $C_k$. Although this training process is very fast, a very large network may potentially be formed, especially if the number of samples in the training set is large.

In order to generate an agent's preliminary decision $u_i$ from the two PNN outputs (two-class problem), we simply use $u_i = \hat{O}_1(\mathbf{x}_i)$, where $\hat{\cdot}$ means normalized such that $\hat{O}_0(\mathbf{x}_i) + \hat{O}_1(\mathbf{x}_i) = 1$. This accounts for inaccuracies in estimating the conditional densities using the PNN. Thus, the case $u_i \approx 1$ implies that the agent strongly believes $\mathbf{x}_i \in C_1$, while $u_i \approx 0$ means that the agent strongly believes $\mathbf{x}_i \in C_0$. In the case when $u_i \approx 0.5$, the agent is unsure of the class of the feature vector.

### D. BPNN Estimation of Class Conditional Probabilities

The conditional density function $p(u_j|C_k)$, $j \in [1, N]$, $j \neq i$, is related to $p(C_k|u_j)$ via the Bayes rule, i.e., $p(u_j|C_k) = p(C_k|u_j)P(u_j)/P(C_k)$. The conditional density $p(C_k|u_j)$ that represents the confidence in PNN decisions $u_j$ can be generated using a BPNN. That is, once the $i$th agent's BPNN-based probability estimator has received $\mathbf{u}_{ir}$, it estimates $p(C_k|u_j)$ such that

$$y_j(k) \approx p(C_k|u_j), \qquad j \in [1, N], \qquad j \neq i \quad (11)$$

where $y_j(k)$ is the $k$th ($k \in \{0, 1\}$) output of the BPNN for the $u_j$ element of $\mathbf{u}_{ir}$. This is done individually for each element in $\mathbf{u}_{ir}$. The BPNN is trained to capture this mapping between the preliminary decisions in the training data set and the classes. It is well known [14] that, if properly trained, the BPNN performance approximates an optimal Bayesian estimator. Similar to the PNN, the same BPNN is used by every agent. That is, only one BPNN needs to be trained.

Using $y_j(k)$ and its relation to $p(u_j|C_k)$ via the Bayes rule, we can rewrite the threshold $t_{fi}$ in (8) as

$$t_{fi}(\mathbf{u}_{ir}) = \frac{P(C_1)^{N-2} \prod_{j=1, j\neq i}^{N} y_j(0)[J_{10} - J_{00}]}{P(C_0)^{N-2} \prod_{j=1, j\neq i}^{N} y_j(1)[J_{01} - J_{11}]}. \quad (12)$$

Using the PNN outputs, $\Lambda(\mathbf{x}_i)$ in (7) becomes

$$\Lambda(\mathbf{x}_i) = \frac{p(\mathbf{x}_i|C_1)}{p(\mathbf{x}_i|C_0)} = \frac{p(C_1|\mathbf{x}_i)P(C_0)}{p(C_0|\mathbf{x}_i)P(C_1)} = \frac{\hat{O}_1(\mathbf{x}_i)P(C_0)}{\hat{O}_0(\mathbf{x}_i)P(C_1)}. \quad (13)$$

Now using (12) and (13), the final decision rule in (7) can be implemented as

$$\Lambda(\mathbf{x}_i) \underset{u_{fi}=0}{\overset{u_{fi}=1}{\gtrless}} \frac{P(C_1)^{N-2} \prod_{j=1, j\neq i}^{N} y_j(0)[J_{10} - J_{00}]}{P(C_0)^{N-2} \prod_{j=1, j\neq i}^{N} y_j(1)[J_{01} - J_{11}]}. \quad (14)$$

This final decision rule assigns a class label to the feature vector $\mathbf{x}_i$ that is the input to the $i$th agent.

### E. CMAC as a Decision Feedback Classifier

The framework developed for CMAC can also be used to implement a DF system for multiaspect classification [10]. The idea behind DF-based multiaspect classification is to make a final classification decision for an aspect (or sonar ping) $i$ using not only the current feature vector $\mathbf{x}_i$, but also the final decisions made at $M$ previous pings, $u_{f,i-j}$, $j \in [1, M]$. The use of

DF has been exploited in high-performance digital communication and equalization systems [15], [16] to reduce the number of channel states in decision making. In [16], it is shown that applying DF can greatly improve the equalization performance, especially when the channel is time varying and the effects of correlated interference are dominant.

In applying DF, the goal is to generate the probability of class membership of an object given the feature vector $\mathbf{x}_i$ and the set of $M$ previous final decisions $\mathbf{u}_{if} = \{u_{f,i-1}, \cdots, u_{f,i-M}\}$, i.e., $p(C_k|\mathbf{x}_i, \mathbf{u}_{if})$, $k \in [0, 1]$. Assuming that the previous decisions and $\mathbf{x}_i$ are conditionally independent, i.e., $p(\mathbf{x}_i, \mathbf{u}_{if}|C_k) = p(\mathbf{x}_i|C_k)\prod_{j=1}^{M} p(u_{f,i-j}|C_k)$, then the final decision $u_{fi}$ about the class membership of $\mathbf{x}_i$ can be made using the likelihood function

$$\Lambda(\mathbf{x}_i) = \frac{p(C_1|\mathbf{x}_i, \mathbf{u}_{if})}{p(C_0|\mathbf{x}_i, \mathbf{u}_{if})} = \frac{P(C_0)^M p(C_1|\mathbf{x}_i) \prod_{j=1}^{M} p(C_1|u_{f,i-j})}{P(C_1)^M p(C_0|\mathbf{x}_i) \prod_{j=1}^{M} p(C_0|u_{f,i-j})} = u_{fi}. \quad (15)$$

Fig. 2 shows how a DF system can be configured using a CMAC with only one agent. As in the CMAC system, the agent first makes a preliminary decision on the class of a ping using the PNN, hence producing $p(C_k|\mathbf{x}_i)$. The coordinator, in this case, is a tapped-delay line or an accumulator that keeps the previous decisions in $\mathbf{u}_{if}$. The DF process is implemented sequentially given a sequence of aspects or sonar pings. That is, for the first aspect/ping since there are no previous decisions, the preliminary decision made by the PNN is kept as the final decision, i.e., $u_{f,i-M} = u_{i-M}$. For the subsequent aspects, the conditional probabilities $p(C_k|u_{f,i-j}), j \in [1, M]$ that represent the confidence in the previous final decisions $u_{f,i-j}$ are generated one at a time using the single-input BPNN inside the agent (see Fig. 2). Note that the difference here is that this BPNN is trained to find the confidence in the final decisions instead of the preliminary decisions $u_i$. Additionally, though Fig. 2 shows $M$ inputs to the BPNN, each conditional probability $p(C_k|u_{f,i-j})$ is generated separately for input $u_{i-j}, j \in [1, M]$. The decision fusion in the DF-based one-agent CMAC implements the decision rule in (15). Note that a nice feature of this system is that a variable number of previous decisions can be used to form $\Lambda(\mathbf{x}_i)$. To train the BPNN, the actual nonthresholded decisions in (15) are used. Therefore, the same BPNN can be used to generate $p(C_k|u_{f,i-j})$ regardless of how many previous final decisions are used to form each $u_{fi}$. The final thresholded (hard or soft) decision, denoted by $y_{fi}$, is the final class label for ping $i$, as shown in Fig. 2.

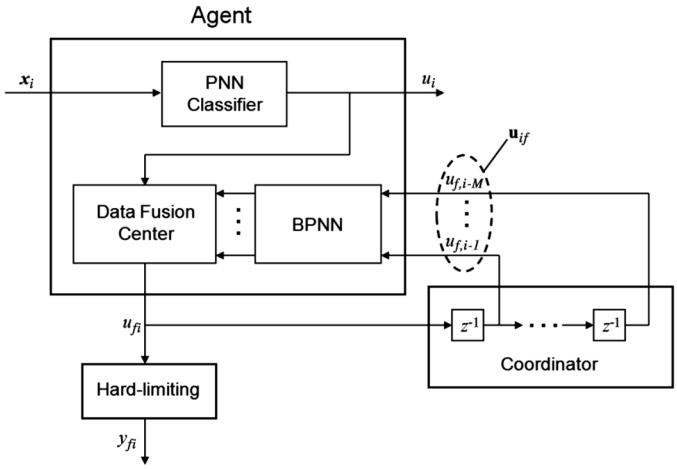


Fig. 2. Implementation of DF-based multiaspect classification using one-agent CMAC.

## III. BOSS Data Collection and Preprocessing

### A. BOSS System and Collected Data Set

The wing BOSS was designed [17] to scan for buried underwater objects using two 1-m hydrophone arrays mounted on the wings of the Bluefin 12 [18] unmanned underwater vehicle (UUV). Each wing contains 20 hydrophone channels, yielding a 40-channel strip array used to collect the sonar returns. This system uses an omnidirectional projector that transmits an FM signal over 3–19 kHz. The wing BOSS is smaller and more mobile than the older generation disk BOSS vehicle [19], as the wing arrays tend to produce less drag than the large circular array. In order to improve the resolution of target imagery, the wing BOSS utilizes time-delay focusing [17] extended to hydrophone data collected over several transmissions. With synthetic aperture processing, the along-track resolution of target imagery improves with distance traveled while forming the synthetic aperture. The use of synthetic aperture processing also allows the along-track length of the BOSS array to be significantly reduced, thereby reducing the hydrophone array drag and surface area and increasing the ease in which BOSS can be deployed on UUVs.

The data set used to test our algorithms was collected off the coast of Panama City, FL, in March 2007, using the wing BOSS system. This data set contains a mixture of mine-like and nonmine-like objects. Fig. 3 shows the exact position and type of each object in the target field. Table I gives the list of the objects, their properties, and burial conditions. As can be seen, objects T1–T10 are buried in sand, while objects M1–M4 are proud on the bottom. Note that there are objects with varying dimensions and compositions within the same class, e.g., a 0.51-m artillery shell, 0.36-m stainless steel sphere, 0.61-m iron cylinder, and 1.83-m concrete pipe that are nonmine-like objects. There are also objects in different classes with similar dimensions and compositions, e.g. mine-like T4 is a 1.83-m iron cylinder [0.46-m outside diameter (OD)], whereas T5 is a 1.52-m iron cylinder (0.18-m OD) and considered to be nonmine-like. Data has been collected for all objects, though some objects have more data than the others. During the data collection phase, the Bluefin 12 UUV operated at a speed of 1.2–1.5 m/s and a ping rate of 25 pings/s. The vehicle altitude was generally between 2.25 and 3.00 m, and the target field bottom was smooth and sandy. The collected sonar data was sampled at 40 043 Hz.

Sonar returns for all objects in the data set were collected over a variety of different runs through the target field. In particular, the Bluefin 12 UUV made five groups of 15–18 runs, where each

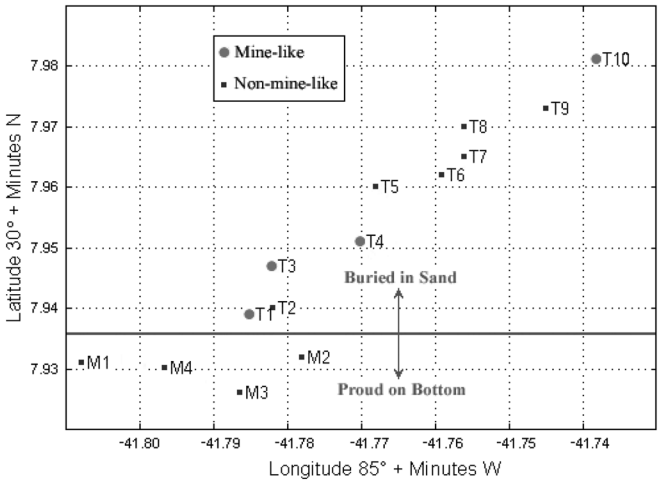


Fig. 3. Target field layout and target information.

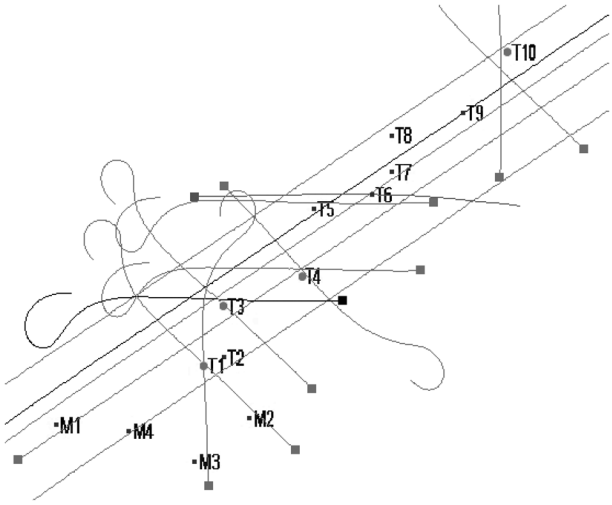


Fig. 4. Runs made by the Bluefin 12 UUV that collected the data used in this study.

TABLE I
DESCRIPTION FOR EACH OBJECT IN THE DATA SET. (ML MEANS MINE-LIKE AND NML MEANS NONMINE-LIKE)

| Label | Description | Class | Status |
|---|---|---|---|
| T1 | 1.83 m Bomb-Shaped Marker (0.46 m OD) | ML | Buried |
| T2 | 0.51 m Artillery Shell | NML | Buried |
| T3 | 1.68 m Bomb-Shaped Target (0.28 m OD) | ML | Buried |
| T4 | 1.83 m Iron Cylinder (0.46 m OD) | ML | Buried |
| T5 | 1.52 m Iron Cylinder (0.18 m OD) | NML | Buried |
| T6 | 0.36 m Stainless Steel Sphere | NML | Buried |
| T7 | 0.89 m Artillery Shell | NML | Buried |
| T8 | 0.36 m Stainless Steel Sphere | NML | Buried |
| T9 | 0.61 m Iron Cylinder (0.15 m OD) | NML | Buried |
| T10 | 1.83 m Bomb-Shaped Marker (0.46 m OD) | ML | Buried |
| M1 | 2.44 m Concrete Pipe (0.46 m OD) | NML | Proud |
| M2 | 1.83 m Concrete Pipe (0.46 m OD) | NML | Proud |
| M3 | 1.83 m Concrete Pipe (0.46 m OD) | NML | Proud |
| M4 | 1.83 m Concrete Pipe (0.46 m OD) | NML | Proud |

group centered around objects T1, T3, T4, T5, or T10. These objects were chosen as the centers of groups mostly because their locations in the target field allow a single run over them to capture sonar returns from a large number of other surrounding objects. In addition, besides T5, all of these objects are considered mine-like, so capturing enough data on them to adequately train classifiers is important. These groups consist of runs that were made in north/south, east/west, and northwest/southeast trajectories at varying distances from the center object. This method of data collection allowed pings from different runs to capture signatures from different aspects of each object. In addition, the Bluefin 12 UUV made 18 diagonal runs in a northeast/southwest trajectory over the entire target field, capturing one to four objects per run. These diagonal runs not only obtained more data on all of the objects covered by the other five groups, except M2 and M3, but were also able to capture sonar returns from objects not covered by the other five groups, namely, those of M1, M4, T7, and T8. Fig. 4 shows the paths of a subset of the runs made by the Bluefin 12 UUV, which are taken from all of the groups of runs mentioned before. Sonar returns obtained during the runs that are shown in Fig. 4 are used to form the training and testing data sets as described in Section IV. As can be seen, in all cases, a run typically captures a large number of sonar pings that are not off an object. For instance, out of typically 500–800 pings for a run, only 8–15 pings on average cover the object of interest.

### *B. Data Preprocessing and Feature Extraction*

A coherence-based frequency subband feature extraction method using CCA is adopted here [20]. CCA decomposes the linear dependence between two channels into the linear dependence between the canonical coordinates of the channels, where this linear dependence can be determined by the corresponding canonical correlations [1]. In this study, CCA exploits the linear dependence (or coherence) between the same frequency subbands of two sonar pings off an object with a given ping separation. The subbands whose measure of coherence are found to be the most discriminatory are then used to form a feature vector for subsequent classification. However, before coherence-based frequency subband features can be extracted from sonar returns in the data set, each return must undergo preprocessing in order to remove portions of the signal that do not represent the object being captured and form the frequency response of the object.

The sonar signal $x_p[n]$ at ping $p$ captured by a hydrophone element can be expressed as

$$x_p[n] = h_p[n] * s[n] + f_p[n] * s[n] + v_p[n] \qquad (16)$$

where "$*$" stands for the convolution operation, $s[n]$ is the transmit signal, the first term on the right-hand side represents the return signal off the target with impulse response $h_p[n]$, the second term is the collective effect of all correlated clutter with impulse response $f_p[n]$, and $v_p[n]$ represents uncorrelated ambient noise. The correlated noise represents such effects as the direct path return, returns off the hydrophone baffle, secondary reflections, and reverberation.

To apply the coherence-based frequency subband feature extraction, the frequency response of the target $H_p[k]$ must be found as accurately as possible. To this end, the recorded signal $x_p[n]$ is first match filtered to better separate the bottom and

target returns from those of the correlated clutter. In the frequency domain, this process yields

$$X_p[k]S^*[k] = H_p[k]\,|S[k]|^2 + F_p[k]\,|S[k]|^2 + V_p[k]S^*[k] \quad (17)$$

where "*" stands for the complex conjugate operation. Now, the effects of the transmit signal can be removed by an inverse filter using

$$\frac{X_p[k]S^*[k]}{|S[k]|^2+\epsilon} \approx H_p[k] + F_p[k] + \frac{V_p[k]}{S[k]} \quad (18)$$

where $\epsilon$ is a small quantity used to avoid singularity problems. Finally, to remove the correlated noise effects, windowing is done in the time (or frequency) domain. The window $w[n]$ in the time domain is rectangular, 161 samples long, and is placed at a sample number chosen to coincide with the onset of the bottom return. This is because the location of the object return varies with respect to the altitude at which the data was collected. The size of the window is determined based upon the extent of the autocorrelation function of the transmit signal. This operation yields an estimate of the target frequency response $\hat{H}_p[k]$, which is subsequently divided into subbands. The size of each subband is five samples, with each sample corresponding to a frequency bin of 10.5 Hz. This subband size was chosen to provide a small enough window for the CCA process to capture the subtle information in each frequency subband, as well as to avoid data poverty when forming the sample covariance matrices needed in the CCA [2] by averaging over the data from the 40 hydrophone channels.

The two-channel CCA feature extraction method is then applied to the same subbands from two different sonar pings with one ping separation. This ping separation was experimentally determined to be optimal. The CCA method produces as many canonical correlations as there are samples in each subband (i.e., five in this case). The first canonical coordinates are the ones that contribute the most to the coherence between the two channels. In other words, these canonical coordinates of the two subbands of the frequency response capture most of the coherence information between the returns in those subbands. Therefore, the corresponding dominant (largest) canonical correlations are kept as features representing a pair of subbands. This process is then repeated for all 305 subbands (in the frequency range 3–19 kHz) in the frequency response and the 20 dominant canonical correlations with the highest discriminatory power, in the sense of the Fisher distance criterion [21], are used to form a feature vector representing the first ping in the pair. Once a feature vector is extracted, this process is repeated for the next pair of pings until all sonar pings over the object of interest are processed.

Fig. 5 shows the plot of the 20 most discriminatory dominant canonical correlations obtained from pairs of pings off mine-like and nonmine-like objects. As can be seen, the features representing mine-like objects tend to have different patterns than those of the nonmine-like objects. This is because the coherence patterns for a given object are dependent on shape, composition, and aspect angle.

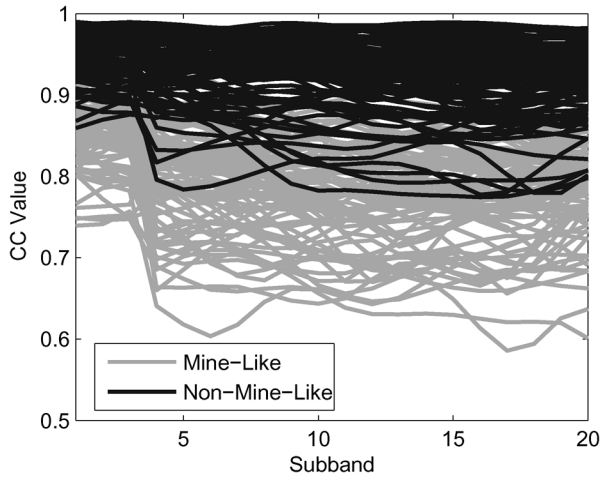


Fig. 5. Plot of 20 dominant canonical correlations for mine-like and nonmine-like objects in the data set.

## IV. Test Results

To train the neural networks used in the CMAC system, three separate subsets of the data are used, one for training/validating the networks, and two for testing. The subset used for training/validating the networks and one for the testing sets is formed from pings off objects captured during runs in the initial five groups that were centered over a different object. The second testing set is formed from pings off objects captured during the group of diagonal runs that traversed the entire target field in a southwest/northeast trajectory. As mentioned in Section III, the diagonal runs capture sonar returns off objects M1, M4, T7, and T8 that are not covered by the other five groups. Therefore, the second testing set contains sonar returns from some objects that are not in the training/validation set or first testing set. Also, the long diagonal runs did not capture sonar returns from objects M2 and M3, while the other five groups of runs did indeed cover them. An equal number of the feature vectors representing mine-like and nonmine-like objects (four to five per object) are used in the training set, and the rest of the subset is used for the validation of the trained networks. This leads to a relatively small PNN structure containing two pools of neurons (one for mine-like and one for nonmine-like objects), each containing 18 pattern neurons (one pattern neuron per training feature vector). Next, the preliminary decisions for each input pattern in the training set along with the corresponding class labels are used to train a two-layer (1–4–2) BPNN, with one input, four hidden-layer neurons, and two outputs. In order to determine the best trained BPNN network, five random initializations are tried. However, because it is a trivial task for the BPNN to accurately classify the scalar $u_i$, performance on the training data set is typically perfect. Thus, the final BPNN structure is decided on after observing the overall system performance on the validation data set.

In this implementation, a CMAC system with three agents (i.e., $N = 3$) is used. Each agent makes a preliminary decision on a different feature vector (20 canonical correlations) representing one of the three pairs of consecutive pings off an object. The same PNN classifier is used in each agent since this PNN is trained to make preliminary decisions on all feature vectors

extracted from pings in the data set. The same BPNN is used in every agent as well, since the BPNN is trained to find the confidence in the PNN decisions. Therefore, the difference between each agent is the feature vector that each makes a preliminary decision on, and the vector $\mathbf{u}_{ir}$ that each agent receives to estimate the class conditional probabilities. This leads to different values for $\hat{O}_k(\mathbf{x}_i)$ and $y_j(k)$ when generating the final decision at each agent using

$$\frac{\hat{O}_1(\mathbf{x}_i)}{\hat{O}_0(\mathbf{x}_i)} \underset{u_{fi}=0}{\overset{u_{fi}=1}{\gtrless}} \frac{\prod_{j=1,j\neq i}^{3} y_j(0)}{\prod_{j=1,j\neq i}^{3} y_j(1)}. \tag{19}$$

Note here that we assume that classes are equally likely, i.e., $P(C_0) = P(C_1)$ and that the "0–1" cost function ($J_{10} = J_{01} = 1$ and $J_{00} = J_{11} = 0$) is used. When using the CMAC system to classify a sequence of consecutive pings, such as multiple pings off the same object, subsequent iterations of the CMAC process shift the window of pings evaluated by each agent by one ping. That is, for the $i$th iteration of the CMAC, the three agents (in this case) evaluate feature vectors $\mathbf{x}_i$, $\mathbf{x}_{i+1}$, and $\mathbf{x}_{i+2}$ and produce final decisions $u_{fi}^{(3)}$, $u_{f,i+1}^{(2)}$, and $u_{f,i+2}^{(1)}$. For the next iteration, the three agents evaluate feature vectors $\mathbf{x}_{i+1}$, $\mathbf{x}_{i+2}$, and $\mathbf{x}_{i+3}$, and produce final decisions $u_{f,i+1}^{(3)}$, $u_{f,i+2}^{(2)}$, and $u_{f,i+3}^{(1)}$, and so on. This procedure produces three final decisions for each feature vector, e.g., $u_{fi}^{(1)}$, $u_{fi}^{(2)}$, and $u_{fi}^{(3)}$ for the $i$th ping, where $u_{fi}^{(n)} \in \{0, 1\}$, $n \in [1, 3]$, with $u_{fi}^{(n)} = 1$ representing a vote for $C_1$, and $u_{fi}^{(n)} = 0$ corresponding to a vote for $C_0$. In order to make the results of CMAC comparable to other multiple-ping classification systems [4], [5], [10], these final decisions must be combined to produce an overall classification decision for that feature vector. This is accomplished by using a majority voting scheme.

Test results are also obtained by applying the implementation of DF using a one-agent CMAC as discussed in Section II-E. The same PNN classifier is used in this DF system, with the same training feature vectors as with original CMAC. Since the BPNN in the DF system makes decisions based upon the previous final decisions $u_{f,i-m}$, these final decisions are needed to train the BPNN. However, the BPNN outputs are also used to generate each final decision. Thus, the samples used to train the BPNN in the DF system must come from final decisions generated without the use of DF. That is, the training samples are generated using (15) with $M = 0$. This two-layer BPNN has one input, four hidden-layer neurons, and two outputs. Five random weight initializations are tried to determine the best trained BPNN based upon the classification performance on the validation data set. Once the best trained BPNN is obtained, the implementation of the DF system uses two previous decisions ($M = 2$) in forming the final decisions using (15), assuming that the two classes are equally likely.

For benchmarking purposes, the nonlinear DLF system of [4] is also implemented. This system is selected since the previous studies [3], [4] indicated its excellent results in buried underwater target classification. In this system, a two-layer BPNN with 40 neurons in the hidden layer and two neurons in the output layer is used to provide intermediate decisions regarding the class of an object based upon the same CCA features that are used with the CMAC system. When training the BPNN, five random weight initializations are tried in order to find the best network on the validation set. A final decision for a sequence of three pings is then formed by fusing the results of three intermediate decisions using a second two-layer BPNN with six inputs, eight neurons in the hidden layer, and two neurons in the output layer. In order to implement the nonlinear DLF system, an additional training data set needs to be created from the outputs of the BPNN that provides the intermediate decisions for each CCA feature vector. This data set is used to train the second (fusion) BPNN, which captures the unknown mapping in the sets of three intermediate decisions. When training the BPNN used in the nonlinear DLF system, five random weight initializations are also tried in order to find the best network on the validation set.

In the following results, the correct classification rate represents the percentage of pings off objects in each data subset that are assigned the correct class by a given classification system. Table II shows the results obtained on each of the three subsets of the data when using the CMAC, DF, and nonlinear DLF systems. As can be seen, the CMAC system provided about 6% improvement over the nonlinear DLF classifier on both testing sets. The improvements of the CMAC over the nonlinear DLF classifier come from the mine-like objects T3, T4, and T10, as well as nonmine-like objects T2 and M1. For mine-like objects, the improvements often come from pings off the center of the object, since the coherence patterns from pairs of these pings resemble those of nonmine-like objects. This property is because these objects are often symmetrical, and in some cases, have similar shapes except maybe around the ends, hence the center pings exhibit similar ping-to-ping coherence pattern. The CMAC system is able to overcome this problem using its ability to share information about the class membership of each ping. As mentioned before, some objects in this data set are very similar in shape, size, and composition to other objects of the opposite class. In particular, mine-like objects T1, T4, and T10 are similar in size and shape to nonmine-like objects M2, M3, and M4 since they are all 1.83 m $\times$ 0.46 m cylinders. In addition, mine-like object T4 has the same composition and shape as nonmine-like object T5, though they have different sizes. The performance of the DF system is slightly worse than that of the CMAC owing to misclassifications, most notably around the leading edge of the mine-like objects T4 and T1 on the first and second testing sets, respectively. These differences are because final decisions made by the DF system on pings off the leading edge of these objects utilize only previous decisions made on empty bottom returns. The CMAC system avoids this problem by using decisions made on pings located on both sides, thus incorporating information from pings off the object.

The receiver operating characteristic (ROC) curves for the proposed CMAC system, the DF system, and the nonlinear DLF system on the first and second testing sets are shown in Fig. 6(a) and (b), respectively. These ROC curves show the percentage of misclassifications of nonmine-like objects $P_{fa}$ and correct classification of mine-like objects $P_{cc}$ that occur as the decision threshold is modified. For the CMAC, the threshold is

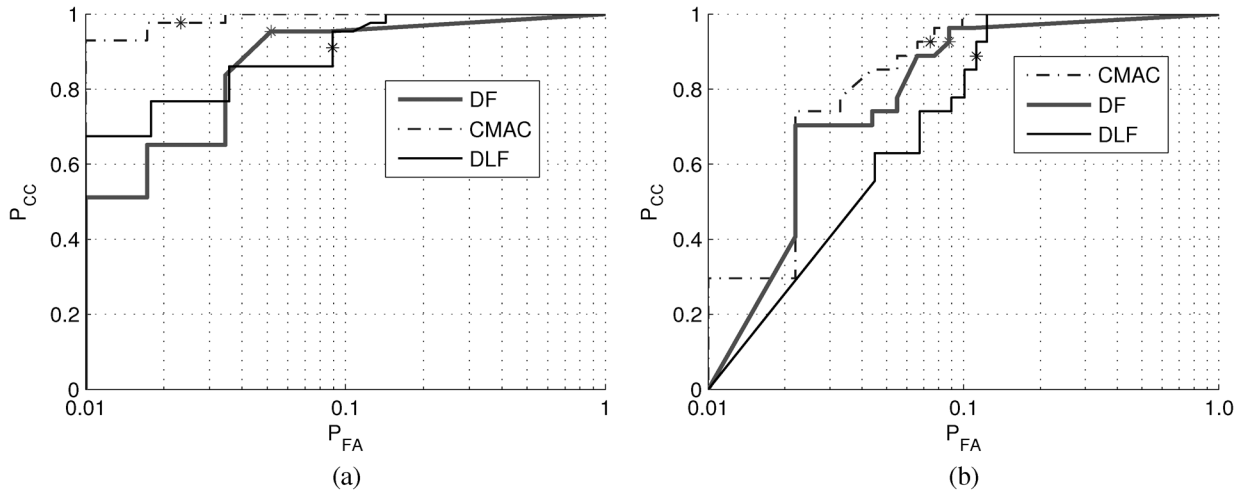


Fig. 6. ROC curves for each classification system on the two testing sets. (a) First testing set. (b) Second testing set.

TABLE II
CORRECT CLASSIFICATION RATES OBTAINED USING EACH SYSTEM ON THE THREE SUBSETS OF THE DATA

| | Validation | Testing 1 | Testing 2 |
|---|---|---|---|
| CMAC | 97.7% | 97.0% | 93.2% |
| DF | 93.0% | 96.0% | 92.4% |
| DLF | 94.0% | 90.9% | 87.1% |

compared to the average of the three soft final decisions made on each feature vector. However, for the DF and nonlinear DLF systems, the threshold is compared to the overall soft final decisions made on each feature vector. For the testing data set 1, Fig. 6(a) indicates that at the knee point of the ROC curve (i.e., where $P_{cc} + P_{fa} = 1$), we have $P_{cc} = 97.7\%$ and $P_{fa} = 2.3\%$ for the CMAC system, $P_{cc} = 94.8\%$ and $P_{fa} = 5.2\%$ for the DF system, and $P_{cc} = 91.1\%$ and $P_{fa} = 8.9\%$ for the nonlinear DLF system. On the other hand, for the testing data set 2, Fig. 6(b) illustrates that at the knee point of the ROC curve, we have $P_{cc} = 92.6\%$ and $P_{fa} = 7.4\%$ for the CMAC system, $P_{cc} = 91.2\%$ and $P_{fa} = 8.8\%$ for the DF system, and $P_{cc} = 88.8\%$ and $P_{fa} = 11.2\%$ for the nonlinear DLF system. These results show 6.6% and 3.8% improvements in performance of the CMAC system over the nonlinear DLF system for the first and second testing sets, respectively. Overall, CMAC reduced the number of false alarms in this data set while maintaining a high degree of classification accuracy for the mine-like objects in each of the runs compared to the nonlinear DLF classifier. These improvements are achieved while at the same time overcoming all the shortcomings of the nonlinear DLF system. The DF system performed nearly as well as the CMAC on all of the data subsets, but suffered a slight decrease in performance for the same reasons mentioned earlier.

### A. *Detection and Classification Results on the Entire Runs*

The results in the previous section show classification accuracy on pings off individual objects in the data set, assuming that the objects in a given run have already been detected before being assigned a class label. The detection process that must be applied to an entire run in order to first detect potential mine-like objects before classification can be avoided by using a classifier trained using the coherence-based frequency subband features. This is due to this property that, in the CCA, the level of coherence between the same frequency subband from two different pings can be used as a measure to detect potential mine-like objects. On the other hand, the pattern of coherence between these frequency subbands, captured by the dominant canonical correlations, would allow the discrimination of the detected objects [2]. This is accomplished by first extracting coherence-based frequency subband features for all pings in a given run through a target field and then applying them to the appropriate classification system in order to determine the location and type of the objects.

Figs. 7–9 show the results obtained by applying the nonlinear DLF system, the DF system, and the proposed CMAC system to all extracted feature vectors on six separate runs. Shown at the top of these figures are "detection/classification strips" that illustrate which sonar pings were detected and classified as mine-like by each method. In these classification strips, the black-colored portions represent the pings at which the classification system has declared a mine-like object. Beneath the classification strips in each figure is the match-filtered image generated using the receiver channel 1 of the wing BOSS array. The match-filtered image is generated solely for the purpose of providing a visual representation of the effects of the objects in the target field on the sonar return. From this image, one can visually verify the locations of the various mine-like and non-mine-like objects in the target field, which appear as hyperbolas. At the bottom of each figure is the synthetic aperture sonar (SAS) image for each run generated by the BOSS as it traversed the target field. These SAS images were generated using a window of 71 pings during the SAS processing. Note that typically SAS images cannot be used for classification since in these images targets and nontargets appear similar. However, as can be seen, the relative locations of the objects in each of the SAS images correspond to the same spatial locations in the corresponding match-filtered images and in the corresponding classification strips. Thus, a spatial relationship is determined and shown between the objects in each image. Mine-like objects are

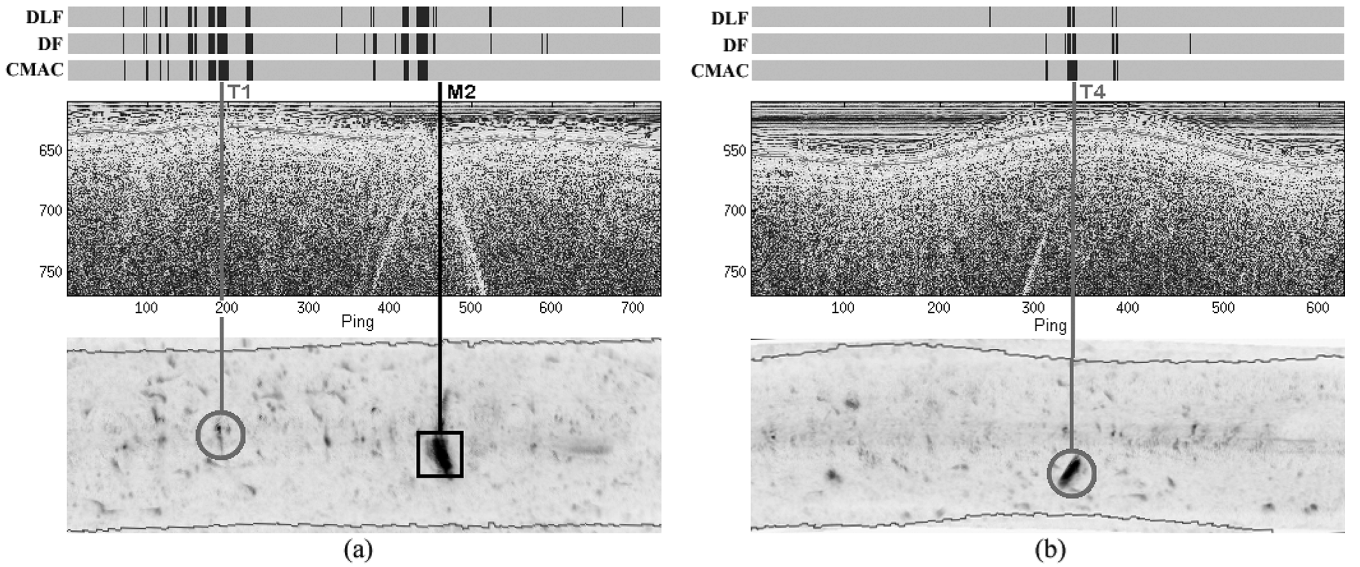


Fig. 7. Classification results on entire runs from the training/validation set. (a) Run 1. (b) Run 2. At the top are detection/classification strips, in the middle is the match-filtered image (match-filtered results versus ping), and at the bottom is the SAS image (distance across track versus ping).

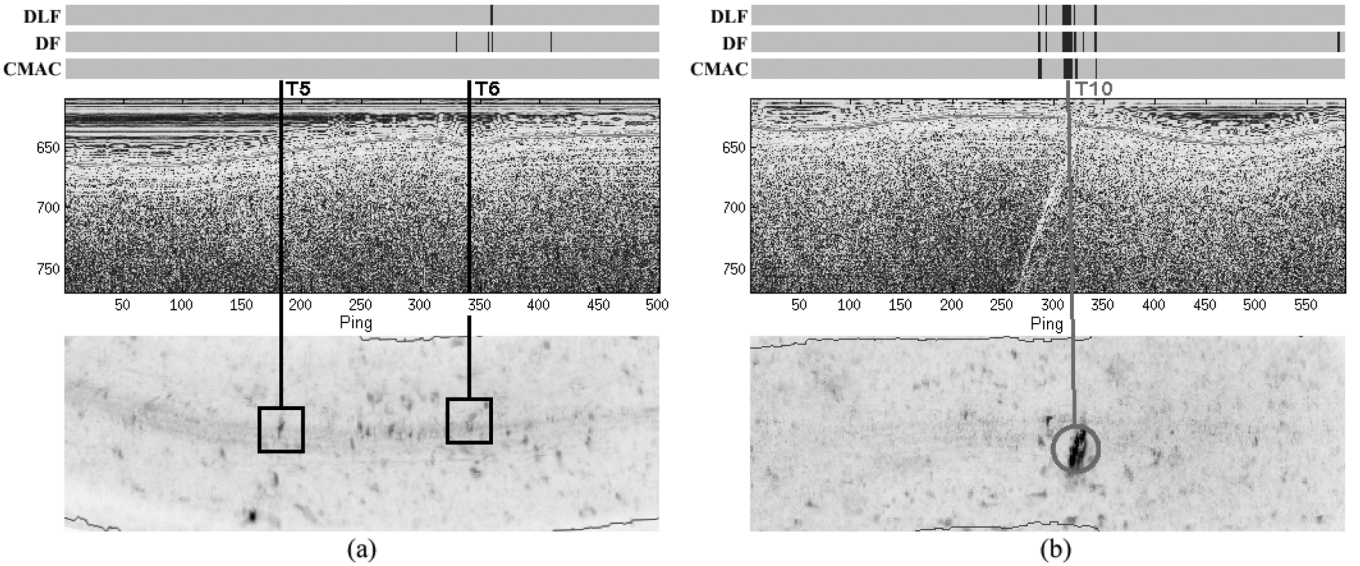


Fig. 8. Classification results on entire runs from the first testing set. (a) Run 3. (b) Run 4. At the top are detection/classification strips, in the middle is the match-filtered image (match-filtered results versus ping), and at the bottom is the SAS image (distance across track versus ping).

circled in the SAS image, and nonmine-like objects are outlined with squares.

These strips illustrate that all three systems are capable of providing simultaneous detection and classification performance on the mine-like objects in the target field. All systems generate relatively few false alarms, though the CMAC system shows significant improvement over the nonlinear DLF system in this regard. This is more noticeable in the runs shown in Figs. 7(a), 8(a), and 9(b). The performance of the DF system is similar to that of CMAC, only it produces a few more false alarms in some areas. The CMAC system performs particularly well on nontargets T5 and T6 [Fig. 8(a)], as well as on nontarget M1 [Fig. 9(b)] as no false alarm is generated for all pings of these objects. Most importantly, the CMAC system correctly classifies all relevant pings off mine-like objects, including mine-like object T4, which is misclassified by the nonlinear DLF system for a few pings in the center and sides. Overall, the CMAC and DF systems perform well on this data set compared to the nonlinear DLF system. In all cases and for all runs, the CMAC system was able to consistently detect and classify the mine-like objects while simultaneously reducing false alarms.

## V. Conclusion

The proposed CMAC in this paper provides a mechanism for multiaspect/ping classification while allowing collaboration and information sharing among individual decision-making agents. CMAC framework can also be used to implement a simple DF classification system. The CMAC system overcomes many of

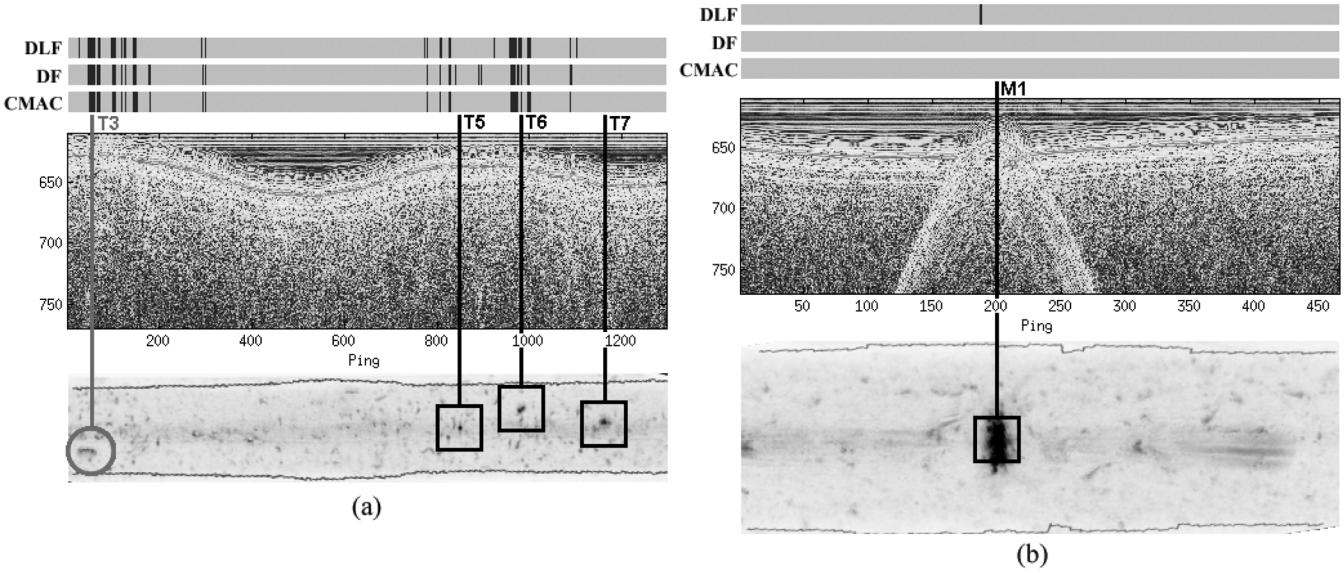


Fig. 9. Classification results on entire runs from the second testing set. (a) Run 5. (b) Run 6. At the top are detection/classification strips, in the middle is the match-filtered image (match-filtered results versus ping), and at the bottom is the SAS image (distance across track versus ping).

the shortcomings of the previous multiple-ping classification systems while simultaneously combining their benefits, e.g., the ability to process a variable number of pings with possibly variable ping separation. When used to classify a mixture of mine-like and nonmine-like objects, it was shown that the proposed CMAC system offers around 6% improvements on both testing sets when compared to a nonlinear DLF system [4]. High correct classification and low false alarm rates were even achieved on the testing set 2, which contained several new objects not included in the training of the classifier in the CMAC. When used to perform simultaneous detection and classification on data from an entire run, it was shown that the CMAC system was able to correctly detect and classify all of the mine-like objects and substantially reduce the number of false alarms not removed by the nonlinear DLF system. The adaptability and ease of implementation of this system coupled with its superior performance on all the BOSS tested data sets [3], [22], [23] makes it a valuable tool for underwater buried object classification.

## APPENDIX

In this appendix, we provide the proof of the threshold equation in (8). Consider the decision strategy at the data fusion center of the $i$th agent. The objective is to obtain the optimum decision rule $\gamma_{fi}$ so as to minimize the expected cost of making an incorrect decision over all possible decision rules, i.e., $E[J\{\gamma_{fi}(\mathbf{x}_i, \mathbf{u}_{ir}), C_k\}]$, where $u_{fi} = \gamma_{fi}(\mathbf{x}_i, \mathbf{u}_{ir})$. The average cost or risk function for the $i$th agent is given by

$$R_{fi} = E\left[J(u_{fi}, C_k)\right] = \sum_{C_k, u_{fi}} \int_{\mathbf{u}_{ir}} \int_{\boldsymbol{\chi}} P(C_k) p(u_{fi}, \mathbf{u}_{ir}, \boldsymbol{\chi} | C_k) \times J_{ik} d\boldsymbol{\chi} d\mathbf{u}_{ir}. \quad (20)$$

Here, $\boldsymbol{\chi}$ is a vector that denotes the augmented set of patterns under consideration, i.e., $\boldsymbol{\chi} = [\mathbf{x}_1^T, \mathbf{x}_2^T, \ldots, \mathbf{x}_N^T]^T$, $\mathbf{u}_{ir}$ denotes the vector of intermediate decisions made by the other agents regarding patterns $\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \ldots, \mathbf{x}_N$, and $J_{ik} = J(u_{fi}, C_k)$ is the cost incurred in choosing class $i$ given that the true class is $k$.

Making the assumption that $u_{fi}$ does not depend on $\mathbf{x}_j$, $j \in [1, N]$, $j \neq i$, and by summing over values of $u_{fi} \in \{0, 1\}$, we get

$$\begin{aligned} R_{fi} = & \sum_{C_k} \int_{\mathbf{u}_{ir}} \int_{\boldsymbol{\chi}} P(C_k) p(u_{fi} = 0 | \mathbf{u}_{ir}, \mathbf{x}_i, C_k) \\ & \times p(\mathbf{u}_{ir}, \boldsymbol{\chi} | C_k) J_{0k} d\boldsymbol{\chi} d\mathbf{u}_{ir} \\ & + \sum_{C_k} \int_{\mathbf{u}_{ir}} \int_{\boldsymbol{\chi}} P(C_k) p(u_{fi} = 1 | \mathbf{u}_{ir}, \mathbf{x}_i, C_k) \\ & \times p(\mathbf{u}_{ir}, \boldsymbol{\chi} | C_k) J_{1k} d\boldsymbol{\chi} d\mathbf{u}_{ir}. \end{aligned} \quad (21)$$

Since the final decision $u_{fi}, i \in [1, N]$ is made based on the decision rule $\gamma_{fi}(\mathbf{x}_i, \mathbf{u}_{ir})$ and is not dependent on $C_k$, we can consolidate like parts in (21) and ignore the constant term

$$\sum_{C_k} \int_{\mathbf{u}_{ir}} \int_{\boldsymbol{\chi}} P(C_k) p(\mathbf{u}_{ir}, \boldsymbol{\chi} | C_k) J_{1k} d\boldsymbol{\chi} d\mathbf{u}_{ir} \quad (22)$$

as it does not depend on $u_{fi}$. This simplifies (21) to

$$\begin{aligned} R_{fi} = & \int_{\mathbf{u}_{ir}} \int_{\mathbf{x}_i} p(u_{fi} = 0 | \mathbf{u}_{ir}, \mathbf{x}_i) d\mathbf{x}_i \\ & \times \sum_{C_k} \int_{\boldsymbol{\chi}^i} P(C_k) p(\mathbf{u}_{ir}, \boldsymbol{\chi} | C_k) [J_{0k} - J_{1k}] d\boldsymbol{\chi}^i d\mathbf{u}_{ir} \end{aligned} \quad (23)$$

where $\boldsymbol{\chi}^i$ is obtained by removing the $i$th pattern from $\boldsymbol{\chi}$, i.e., $\boldsymbol{\chi}^i = [\mathbf{x}_1^T, \ldots, \mathbf{x}_{i-1}^T, \mathbf{x}_{i+1}^T, \ldots, \mathbf{x}_N^T]^T$.

From (23), we see that the risk is minimized if

$$p(u_{fi} = 0 | \mathbf{u}_{ir}, \mathbf{x}_i) = \begin{cases} 0, & \text{if } \sum_{C_k} \int_{\boldsymbol{\chi}^i} P(C_k) p(\mathbf{u}_{ir}, \boldsymbol{\chi} | C_k) [J_{0k} - J_{1k}] d\boldsymbol{\chi}^i \geq 0 \\ 1, & \text{otherwise.} \end{cases} \quad (24)$$

Hence, the final decision rule should be such that

$$\sum_{C_k} P(C_k) \int_{\boldsymbol{\chi}^i} p(\mathbf{u}_{ir}, \boldsymbol{\chi}|C_k)[J_{0k} - J_{1k}] d\boldsymbol{\chi}^i \underset{u_{fi}=0}{\overset{u_{fi}=1}{\gtrless}} 0. \quad (25)$$

Keeping in mind that $u_l$ represents the intermediate decision made by agent $l$, we can say that $u_l$ depends only on the $l$th input pattern $\mathbf{x}_l$, $l \in [1, N]$, $l \neq i$, i.e.,

$$\begin{aligned} \int_{\boldsymbol{\chi}^i} p(\mathbf{u}_{ir}, \boldsymbol{\chi}|C_k) d\boldsymbol{\chi}^i &= p(\mathbf{x}_i|C_k) \prod_{j=1, j\neq i}^{N} \int_{\mathbf{x}_j} p(u_j, \mathbf{x}_j|C_k) d\mathbf{x}_j \\ &= p(\mathbf{x}_i|C_k) \prod_{j=1, j\neq i}^{N} p(u_j|C_k). \end{aligned} \quad (26)$$

This result is obtained based on the assumption that the preliminary decisions $u_j$, $j \in [1, N]$, $j \neq i$, contained in the vector $\mathbf{u}_{ir}$ are conditionally independent and using the marginal density, i.e., $\int_{\mathbf{x}_j} p(u_j, \mathbf{x}_j|C_k) d\mathbf{x}_j = p(u_j|C_k)$. Now substituting (26) into (25), we get an expression for the final decision rule $u_{fi}$ as follows:

$$\sum_{C_k} P(C_k) p(\mathbf{x}_i|C_k) \prod_{j=1, j\neq i}^{N} p(u_j|C_k)[J_{0k} - J_{1k}] \underset{u_{fi}=0}{\overset{u_{fi}=1}{\gtrless}} 0. \quad (27)$$

Expanding over $C_k$, we get

$$\begin{aligned} &P(C_1) p(\mathbf{x}_i|C_1) \prod_{j=1, j\neq i}^{N} p(u_j|C_1)[J_{01} - J_{11}] \\ &\quad \underset{u_{fi}=0}{\overset{u_{fi}=1}{\gtrless}} P(C_0) p(\mathbf{x}_i|C_0) \prod_{j=1, j\neq i}^{N} p(u_j|C_0)[J_{10} - J_{00}]. \end{aligned} \quad (28)$$

Further simplifying the expression leads to an equation identical to (7) when (8) is invoked.

## ACKNOWLEDGMENT

The data used in this work was supplied by the Naval Surface Warfare Center (NSWC), Panama City, FL. The authors would like to thank Dr. S. G. Schock at Florida Atlantic University, Boca Raton, for providing the technical support on this data set.

## REFERENCES

[1] L. L. Scharf and C. T. Mullis, "Canonical coordinates and the geometry of inference, rate, and capacity," *IEEE Trans. Signal Process.*, vol. 48, no. 3, pp. 824–831, Mar. 2000.

[2] A. Pezeshki, M. R. Azimi-Sadjadi, and L. L. Scharf, "Undersea target classification using canonical correlation analysis," *IEEE J. Ocean. Eng.*, vol. 32, no. 4, pp. 948–955, Oct. 2007.

[3] M. Yamada, J. Cartmill, and M. R. Azimi-Sadjadi, "Buried underwater target classification using the new BOSS and canonical correlation decomposition feature extraction," in *Proc. MTS/IEEE OCEANS Conf.*, Aug. 2005, vol. 1, pp. 589–596.

[4] M. R. Azimi-Sadjadi, D. Yao, and G. J. Dobeck, "Underwater target classification using wavelet packets and neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 3, pp. 784–794, May 2000.

[5] M. Robinson, M. R. Azimi-Sadjadi, and J. Salazar, "Multiaspect target discrimination using hidden Markov models and neural networks," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 447–459, Mar. 2005.

[6] P. Runkle, B. Bharadwaj, L. Couchman, and L. Carin, "Hidden Markov models for multi-aspect target classification," *IEEE Trans. Signal Process.*, vol. 47, no. 7, pp. 2035–2040, Jul. 1999.

[7] P. Bharadwaj, P. Runkle, and L. Carin, "Target identification with wave-based matched pursuits and hidden Markov models," *IEEE Trans. Antennas Propag.*, vol. 47, no. 10, pp. 1543–1554, Oct. 1999.

[8] N. Dasgupta, P. Runkle, L. Couchman, and L. Carin, "Dual hidden Markov model for characterizing wavelet coefficients from multi-aspect scattering data," *Signal Process.*, vol. 81, pp. 1303–1316, Jun. 2001.

[9] J. Shihao, L. Xuejun, and L. Carin, "Adaptive multiaspect target classification and detection with hidden Markov models," *IEEE Sensors J.*, vol. 5, no. 5, pp. 1035–1042, Oct. 2005.

[10] M. R. Azimi-Sadjadi, D. Yao, A. Jamshidi, and G. J. Dobeck, "A biologically inspired adaptive underwater target classification using a multi-aspect decision feedback unit," in *Proc. MTS/IEEE OCEANS Conf.*, Oct. 2002, vol. 1, pp. 38–45.

[11] Z. Chair and P. K. Varshney, "Distributed Bayesian hypothesis testing with distributed data fusion," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-18, pp. 695–699, Sep./Oct. 1988.

[12] D. F. Specht, "Probabilistic neural network," *Neural Netw.*, vol. 3, pp. 109–118, 1990.

[13] P. K. Varshney, *Distributed Detection and Data Fusion*, 1st ed. New York: Springer-Verlag, 1997.

[14] S. Haykin, *Neural Networks*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1999.

[15] S. Chen, B. Mulgrew, and S. McLaughlin, "Adaptive Bayesian equalizer with decision feedback," *IEEE Trans. Signal Process.*, vol. 41, no. 9, pp. 2918–2927, Sep. 1993.

[16] S. Chen, B. Mulgrew, and P. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 4, no. 4, pp. 570–579, Jul. 1993.

[17] S. G. Schock and J. Wulf, "Imaging performance of BOSS using SAS processing," in *Proc. MTS/IEEE OCEANS Conf.*, Sep. 2006, pp. 1–5.

[18] G. Allen, G. Sulzberger, J. Bono, J. Pray, and T. Clem, "Initial evaluation of the new real-time tracking gradiometer designed for small unmanned underwater vehicles," in *Proc. MTS/IEEE OCEANS Conf.*, 2005, vol. 3, pp. 1956–1962.

[19] S. G. Schock, A. Tellier, J. Wulf, J. Sara, and M. Ericksen, "Buried object scanning sonar," *IEEE J. Ocean. Eng.*, vol. 26, no. 4, pp. 677–689, Oct. 2001.

[20] N. Wachowski and M. R. Azimi-Sadjadi, "Buried underwater target classification using frequency subband coherence analysis," in *Proc. MTS/IEEE OCEANS Conf.*, Quebec City, QC, Canada, Sep. 15–18, 2008, ISBN: 978-1-4244-2620-1, Library of Congress: 2008904879, IEEE Catalog Number: CFP08OCE-CDR.

[21] R. Duda, P. Hart, and D. G. Storke, *Pattern Classification*. New York: Wiley, 2001.

[22] N. Wachowski, J. Cartmill, and M. R. Azimi-Sadjadi, "Underwater target classification using the wing BOSS and multi-channel decision fusion," in *Proc. SPIE Defense Security*, Apr. 2007, vol. 6553, pp. Q1–Q10.

[23] J. Cartmill, M. R. Azimi-Sadjadi, and N. Wachowski, "Buried underwater object classification using a collaborative multi-aspect classifier," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2007, pp. 1807–1812.

**Jered Cartmill** (M'01) received the B.S. degree in engineering science from Trinity University, San Antonio, TX, in 2003 and the M.S. degree in electrical engineering from Colorado State University, Fort Collins, in 2006.

His interests include pattern recognition and classification, signal and image processing, communication systems, and neural networks. Currently, he is a Software Engineer at Argon ST, Fairfax, VA.

**Neil Wachowski** received the B.S. degree in electrical engineering from Michigan Technological University, Houghton, in 2002. Currently, he is working towards the M.S. degree at the Electrical and Computer Engineering Department, Colorado State University, Fort Collins.

He was a Field Applications Engineer at Texas Instruments Inc. from 2002 to 2006. His research interests include target detection and classification, digital signal processing, and digital image processing.

**Mahmood R. Azimi-Sadjadi** (S'81–M'81–SM'89) received the M.S. and Ph.D. degrees from the Imperial College of Science and Technology, University of London, London, U.K., in 1978 and 1982, respectively, both in electrical engineering with specialization in digital signal/image processing.

Currently, he is a Full Professor at the Electrical and Computer Engineering Department, Colorado State University (CSU), Fort Collins. He is also the Director of the Digital Signal/Image Laboratory, CSU. His main areas of interest include digital signal and image processing, wireless sensor networks, target detection, classification and tracking, adaptive filtering, system identification, and neural networks. His research efforts in these areas resulted in over 250 journal and referenced conference publications. He is the coauthor of the book *Digital Filtering in One and Two Dimensions* (New York: Plenum Press, 1989).

Prof. Azimi-Sadjadi served an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON NEURAL NETWORKS.