An Information Theoretic Approach for In-Situ Underwater Target Classification

Mahmood R. Azimi-Sadjadi and Neil Wachowski

Abstract— **This paper introduces a method for in-situ underwater target classification, based on an image retrieval system, that can be implemented using a simple two-layer kernel-based network. This system incorporates a learning mechanism that captures new information for discriminating between objects in different classes or within the same class from a set of input-output pairs with associated confidence scores. A strategy to select the most informative patterns for optimal parameter adaptation during in-situ learning is also described. The system is then tested on a database of synthetically generated sonar images. The ability of the system to correctly classify images containing objects in different environmental and operating conditions than those used for original training, as well as its ability to incorporate new object types without perturbing the classification performance on other object types are demonstrated.**

I. Introduction

Classification of underwater objects in sonar imagery [1]-[2] is a complicated problem due to various factors such as variations in operating and environmental conditions, presence of spatially varying clutter, variations in target highlight and shadow structures with respect to aspect angle, range and grazing angle, and variations in compositions of the objects. Moreover, bottom features such as coral reefs, sand formations, and vegetation may obscure a target. One way to alleviate the difficulties caused by these variations is through the use of a computer-aided detection and classification (CAD/CAC) system that performs *in-situ* learning, and thus takes into account environmental and operating condition changes as well as the presence of new contacts and man-made or natural clutter. Consequently, such a CAD/CAC system can offer high classification accuracy and flexibility in making reliable decisions in such conditions.

Various methods have been explored for target classification in sonar imagery. The reader is referred to [1]-[2] for more details. One major issue with these previously developed CAD/CAC systems is that they lack the ability to perform in-situ learning and, consequently, do not offer high classification accuracy and flexibility in making reliable decisions in new environmental and operating conditions. The main goal of this paper is therefore to develop a CAD/CAC system with in-situ learning capability for classification of underwater objects in sonar imagery. The proposed system is based on the extension of the method in [3], which is a content-based image retrieval (CBIR) system that can be implemented using a two-layer kernel-based network. This system incorporates a *model-reference* learning mechanism for capturing the class and/or within-class (details that separate images within the same class) information from a set of input-output pairs and associated confidence scores. A novel strategy to select the most informative patterns for optimal parameter adaptation during in-situ learning was also introduced using the Fisher information matrix. This paper discusses extensions made to the system in [3] to allow implementation as a flexible classifier for sonar imagery. Specifically, we discuss (a) proper selection of a set of exemplar patterns used to initialize the system and to represent the entire image database during retrieval, (b) a framework that can be used to perform in-situ learning to correctly classify (or identify) a new image that is drawn from an entirely new environment (e.g. background clutter), and (c) the approach used to incorporate samples that represent new object types encountered during the in-situ learning. This framework was chosen because it offers some distinct advantages over alternatives such as support vector machines [4], Bayesian inference [5], and radial basis functions [6]. That is, the method in [3] does not rely on assumptions about the clustering behavior of positive and negative samples and offers precise control over the relevancy scores.

This paper is organized as follows. Section II discusses the details of the proposed CBIR system for in-situ learning, including the model reference learning mechanism, selective sampling framework, and its implementation as a kernel-based network. In Section III the sonar image databases that are used to perform the experiments conducted in this study are described. Section IV discusses the framework used to apply the proposed CBIR system to the problem of classifying objects in sonar imagery with in-situ learning taking place during testing. The results of applying the proposed system to the sonar image databases are also presented. Finally, Section V provides concluding remarks.

II. Proposed CBIR System with In-Situ Learning

The structure of the proposed system is shown in Figure 1. It contains several components, namely the nonlinear mapping function $\Phi(\cdot) : \mathbb{R}^M \rightarrow \mathbb{R}^{N_f}$ with $N_f \gg M$ that maps the original M-dimensional feature vector $\underline{x}_j$ (extracted from an image) to a high dimensional feature space, a series of adaptable *implicit* feature mapping matrices A_j's that adapt the matching functions in the high dimensional feature space, a retrieval or scoring system that generates labels and confidence scores for the applied pattern, and model-reference learning for situations where class membership and desired scores of samples in a new environment are available. It should be noted that this system is also capable

Mahmood R. Azimi-Sadjadi and Neil Wachowski are with Information System Technologies, Inc., Fort Collins, CO 80521 (email: mo@infsyst.biz).

978-1-4244-8126-2/10/$26.00 ©2010 IEEE

of relevance feedback learning [3] for incorporation of expert operator's high-level concepts as shown in Figure 1; though the experiments conducted here do not make use of this type of learning. The proposed system captures the environmental context and high-level semantic information not only by a series of adaptable mapping matrices A_j but also by proper choice of the high dimensional nonlinear mapping function $\Phi(\cdot)$.

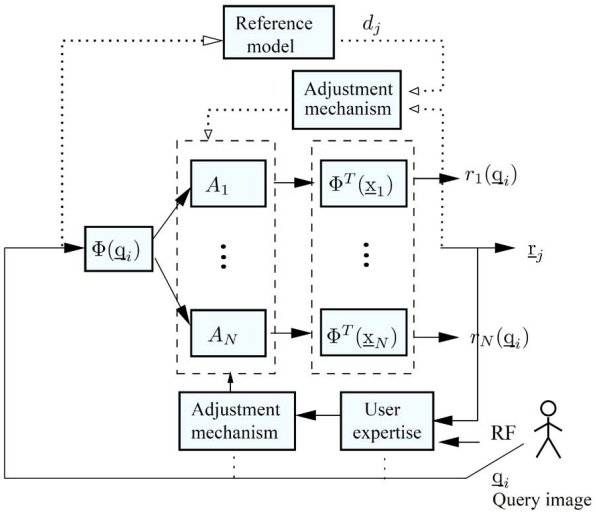

Fig. 1. In-situ adaptable target classification system with model-reference and relevance feedback learning.

Upon submitting a pattern (feature) vector $\underline{q}$, the system performs a similarity matching in the high dimensional feature space to generate confidence scores for those *exemplar* patterns that are already captured in the original training database $X = [\underline{x}_1 \underline{x}_2 \cdots \underline{x}_N]$ with known class labels. This is done using

$$r_j(\underline{q}) = s(\underline{q}, \underline{x}_j) = \Phi^T(\underline{x}_j) A_j \Phi(\underline{q}), \quad j \in [1, N] \quad (1)$$

where $\Phi(\underline{q}) = [\phi_1(\underline{q}) \quad \phi_2(\underline{q}) \quad \dots \quad \phi_{N_f}(\underline{q})]^T$ with $\phi_j(\cdot)$'s being scalar functions of the pattern $\underline{q}$, similarly for $\Phi(\underline{x}_j)$, and $s(\underline{q}, \underline{x}_j)$ is the matching (or scoring) function between the submitted pattern $\underline{q}$ and stored exemplar patterns $\underline{x}_j$'s. Thus, in this system the mapping function $\Phi(\cdot)$ first maps both patterns $\underline{q}$ and $\underline{x}_j$ to the higher dimensional space, in which feature mapping matrix A_j then shapes the matching space for pattern $\underline{x}_j$. The mapping to a high dimensional feature space together with multiple function adaptability provides a much better opportunity to capture the information in new operating and environmental conditions.

At first glance it may appear that finding a matrix for every pattern in the database, X, is too inefficient and impractical, especially in the large (possibly infinite) dimensional mapped feature space. However, in our proposed method this is entirely avoided using the *kernel trick* [4]. Additionally, the proposed system in Figure 1 and all the learning rules can be implemented using a simple structurally adaptive kernel machine [3] as will be shown later.

A. Initialization

System initialization is required in any CBIR system to train it based upon unlabeled and/or unstructured images in the database. This one-step training typically involves storing all the images and establishing a similarity measure for initial and crude retrieval and classification. In the proposed system the initial setup of similarity functions in (1) involves choosing initial mapping matrices $A_j,\ j \in [1, N]$ given *only* the initial image database X with N images. Since, except the images $\underline{x}_j$'s, no other information is available in this phase, the mapping matrices can be initialized to either the identity matrix or the projection matrix $A_j = P_{\Phi(\underline{x}_j)} = \Phi(\underline{x}_j)[\Phi^T(\underline{x}_j)\ \Phi(\underline{x}_j)]^{-1}\Phi^T(\underline{x}_j)$, for every image $\underline{x}_j$. Both of these choices for A_j matrices produce the same similarity function $s(\underline{q}, \underline{x}_j) = \Phi^T(\underline{x}_j)\Phi(\underline{q})$ or simply $s(\underline{q}, \underline{x}_j) = k(\underline{x}_j, \underline{q})$ for kernel producing nonlinear function $\Phi(\cdot)$. However, the second choice is preferred because of its nice geometrical properties [3].

Once the CBIR system is initialized, it may undergo further training using the model reference (or relevance feedback) learning mechanism. In the following subsection, we describe the proposed model-reference learning mechanism and some of its features.

B. Model-Reference Learning

The goal of model-reference learning is to capture the input-output mapping of a *reference model* for an ensemble set of patterns (query images) and their corresponding labels and desired scores. The dotted lines in Figure 1 indicate the flow of information and the subsystems involved in the model-reference learning. Model-reference learning may be used to capture some specific *a priori* class and within-class information shared by images in the database. To accomplish this goal, a *model-reference training database* is generated using the specific information of the images in the database and their associated desired scores.

Given a set of L training pairs $\left\{\underline{q}_i, d_j^{(i)}\right\}_{i=1}^{L}$ in the model-reference training database, where $\underline{q}_i$ is the i^{th} new pattern and $d_j^{(i)}$ is the confidence score for exemplar pattern $\underline{x}_j$, the goal of model-reference learning is to incorporate every new pattern, $\underline{q}_i,\ i \in [1, L]$ into either a new or an existing category (class) or subcategory (within-class) and generate the desired confidence score between the new pattern and the exemplar pattern $\underline{x}_j$ representing that category. Note that the score $d_j^{(i)}$ is typically defined either by a user or some confidence assignment method. This procedure is done by finding a new implicit mapping matrix (not explicitly computed) $\hat{A}_j$ and the mapping function $\Phi(\cdot)$ to yield the desired score for the exemplar pattern. Using the proposed generalized similarity function, the problem can be posed as finding $\hat{A}_j$ and $\Phi(\cdot)$ such that

$$\Phi^T(\underline{x}_j)\hat{A}_j\Phi(\underline{q}_i) = d_j^{(i)}, \quad i \in [1, L]. \quad (2)$$

The problem is ill-posed because many choices for the mapping matrices $\hat{A}_j$ and function $\Phi(\cdot)$ can potentially satisfy

this requirement. Furthermore, due to the high dimensionality of $\Phi(\cdot)$, the problem remains ill-posed even if a particular class of functions is chosen. However, we can restrict the choice of the basis eigenfunctions $\phi_l(\cdot)$'s to those that generate a positive and symmetric kernel $k(\underline{s},\underline{t}) = k(\underline{t},\underline{s})$ that satisfy Mercer's theorem [4], then $k(\underline{s},\underline{t}) = \sum_{l=1}^{N_f} \phi_l(\underline{s})\phi_l(\underline{t}) = \Phi^T(\underline{s})\Phi(\underline{t})$. Thus, the problem reduces to finding the matrix $\hat{A}_j$ using the chosen basis functions. This problem is ill-posed because if $\hat{A}_j$ is a solution to (2) so is $\hat{A}_j + B_j$, where $\Phi(\underline{x}_j) \in \mathcal{N}ull(B_j)$, i.e. in null space of B_j. Therefore, a solution with minimum Frobenius norm requires that each column be a multiple of $\Phi(\underline{x}_j)$ leading to a rank-one matrix $\hat{A}_j = \Phi(\underline{x}_j)\underline{w}_j^T$ with some weight vector $\underline{w}_j$. This problem may be converted into a well-posed one by using the regularized least squares (LS) [4] for finding optimal $\underline{w}_j^*$ such that

$$\underline{w}_j^* = \arg\min_{\underline{w}_j} \mathcal{T}(\underline{w}_j, \lambda) \tag{3}$$

where $\mathcal{T}$ is the regularized LS cost function [4] given by

$$\mathcal{T}(\underline{w}_j, \lambda) = \frac{1}{2}||\underline{d}_j - c_j\Psi^T\underline{w}_j||^2 + \frac{1}{2}\lambda||\underline{w}_j||^2. \tag{4}$$

Here $\underline{d}_j = [d_j^{(1)}\ d_j^{(2)}\ \ldots\ d_j^{(L)}]^T$ is the vector of desired scores for all patterns $\underline{q}_i$, $i \in [1, L]$, $c_j = \Phi^T(\underline{x}_j)\Phi(\underline{x}_j) = k(\underline{x}_j, \underline{x}_j)$ and $\Psi = [\Phi(\underline{q}_1)\ \Phi(\underline{q}_2)\ \ldots\ \Phi(\underline{q}_L)]$ is the matrix containing all the mapped patterns. The first term in (4) is the sum squared error term induced when trying to reproduce the scores $d_j^{(i)}$, $i \in [1, L]$, while the second term is the regularization term with λ being the regularization parameter. The optimal solution to this regularized LS problem can easily be found as

$$\underline{w}_j^* = \frac{1}{c_j}\Psi(\Psi^T\Psi + c_j^{-2}\lambda I)^{-1}\underline{d}_j. \tag{5}$$

As can be seen from (5), the effect of the regularization term is equivalent to adding a diagonal loading to the $L \times L$ Gram matrix $G = \Psi^T\Psi$, which has elements that are $g_{l,m} = k(\underline{q}_l, \underline{q}_m)$. The diagonal loading depends on λ and $c_j^{-2} = 1/k^2(\underline{x}_j, \underline{x}_j) \neq 0$ and guarantees the existence of the inverse for matrix $(\Psi^T\Psi + c_j^{-2}\lambda I)$. Note that for symmetric, positive definite, and non-degenerate kernel functions the Gram matrix is non-singular even without the diagonal loading term. An example of a non-degenerate kernel is the Gaussian kernel, which is used for the experiments performed in this research, and is defined as $k(\underline{s},\underline{t}) = e^{-a||\underline{s}-\underline{t}||^2}$, where a is a constant and positive real number. Thus, for non-degenerate kernels the regularization term may be avoided.

Once the optimal solutions $\underline{w}_j^*$, $j \in [1, N]$'s for all the patterns in the model-reference training database are computed the new similarity (or scoring) functions, $r_j(\underline{q})$, for a new pattern $\underline{q}$ in (1) can easily be determined by using the optimal vector $\underline{w}_j^*$ in (5). This yields

$$\begin{aligned} r_j(\underline{q}) &= \Phi^T(\underline{q})\Psi(\Psi^T\Psi + c_j^{-2}\lambda I)^{-1}\underline{d}_j \\ &= \sum_{i=1}^{L} b_{ji}k(\underline{q}, \underline{q}_i) = \mathcal{K}^T(\underline{q})\underline{b}_j \end{aligned} \tag{6}$$

where $\underline{b}_j := (\Psi^T\Psi + c_j^{-2}\lambda I)^{-1}\underline{d}_j$ and $\mathcal{K}(\underline{q}) := \Phi^T(\underline{q})\Psi = [k(\underline{q}, \underline{q}_1)\ \ldots\ k(\underline{q}, \underline{q}_L)]^T$.

This result implies that the new similarity function in (6) for the model-reference learning can be implemented in the kernel domain without the need to explicitly compute $\underline{w}_j^*$. Note that for most kernels c_j is a constant (e.g. for Gaussian $c_j = 1$). This implies that only one matrix inversion is needed to compute all $\underline{b}_j$'s in (6). Additionally, there are recursive algorithms for computing the inverse of a Gram matrix that do not require any matrix inversion operation [3]. These recursive algorithms could be used to yield recursive learning equations for both the scoring function $r_j(\underline{q})$ and the weight vector $\underline{b}_j$.

C. Implementation Using A Kernel Machine

Here it is shown how the system in Figure 1 and the associated in-situ model-reference learning mechanism can be implemented using a simple two-layer kernel-based network in Figure 2. This network consists of N pools of neurons, one pool for every pattern in the original training database. Initially, each pool is formed of only one neuron corresponding to the exemplar patterns in the initial image database X. That is, when initializing the j^{th} pool, exemplar pattern $\underline{q}_1 = \underline{x}_j$ is incorporated into the pool by setting the neuron kernel function to $k(\underline{q}, \underline{q}_1)$ for any input pattern $\underline{q}$. The weight connecting this neuron to the corresponding output neuron is also initially set to $b_{j1} = 1$. Thus, the initial confidence score generated by the j^{th} pool for pattern $\underline{q}$ is $r_j(\underline{q}) = k(\underline{q}, \underline{q}_1)$, which implements the initialization procedure mentioned in Section II-A. Note that, if an overabundance of initial images are a available, it may be beneficial to select a subset of X containing images that are representative of the entire database to use for initialization. Selection of this subset is discussed in Section IV. Once initially setup, the model-reference learning results in expansion of the pools and updating the weight vector $\underline{b}_j$ of the second layer.

Figure 2 shows how pool j associated with class or within-class j is expanded by incorporating $L-1$ additional neurons with kernel functions $k(\underline{q}, \underline{q}_i)$, $i \in [2, L]$. In essence, all patterns $\underline{q}_i$'s, $i \in [1, L]$, captured by pool j, are now associated with class or within-class j represented by exemplar pattern $\underline{x}_j$ in the original training database and they form the basis for representing the confidence scoring function $r_j(\underline{q}) = \underline{b}_j^T\mathcal{K}(\underline{q})$ in (6) at the output of pool j for the submitted pattern, $\underline{q}$. The scoring function adaptation requires updating the weight vector $\underline{b}_j$ by using the desired score information, $\underline{d}_j$, in $\underline{b}_j = (\Psi^T\Psi)^{-1}\underline{d}_j$. In this kernel machine, a new pool of neurons can be added without impacting the existing pools where the newly encountered pattern will be assigned as the exemplar pattern of this newly generated pool.

D. An Information-Theoretic Selective Sampling Method

The idea behind selective sampling, statistical active learning or experimental design [7] is to selectively search for those most *informative* training samples that minimize the generalization error and avoid over-fitting problems. An information-theoretic approach is typically adopted to guide

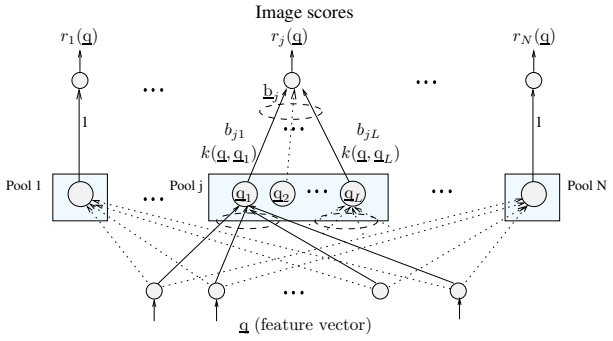

Fig. 2. A structurally adaptable kernel network for in-situ learning implementation.

this search process. The proposed selective sampling method also controls the expansion of the neuron pools in Figure 2 by choosing only those most informative samples during the relevance feedback learning. The selected sample forms a new unit in the pool with a kernel activation defined by that pattern. This new sample brings old and new information to the pool. While the *old information* (can be deduced from the other already incorporated patterns) is not playing a role in learning, the *new information* controls the weight adjustment and scoring function adaptation in the linear kernel space. This issue will be studied in the context of the Fisher information matrix for sequential selection of training samples or best basis functions.

For any new pattern $\underline{\mathfrak{q}}_i$ in the set $\mathcal{Q}$ [1], the score generated by the system, $r_j(\underline{\mathfrak{q}}_i)$ and the desired score, $y_j(\underline{\mathfrak{q}}_i)$, for this pattern are related via

$$y_j(\underline{\mathfrak{q}}_i) = r_j(\underline{\mathfrak{q}}_i) + \epsilon_i = \underline{\mathfrak{b}}_j^T \mathcal{K}(\underline{\mathfrak{q}}_i) + \epsilon_i \tag{7}$$

where ϵ_i is the error in the representation which is assumed to have zero mean and variance σ_i^2. Now, if the same set of basis functions (determined based upon the learnt L patterns) is to be used to minimize the generalization error on this set $\mathcal{Q}$, the best linear unbiased estimator (BLUE) [9] of $\underline{\mathfrak{b}}_j$ given L observations, $\underline{\mathfrak{q}}_i \in \mathcal{Q}$, can be expressed as

$$\hat{\underline{\mathfrak{b}}}_j = M_0^{-1} \left(\sum_{i=1}^{L} y_j(\underline{\mathfrak{q}}_i) \mathcal{K}(\underline{\mathfrak{q}}_i) / \sigma_i^2 \right) \tag{8}$$

where $M_0 = \sum_{i=1}^{L} \frac{1}{\sigma_i^2} \mathcal{K}(\underline{\mathfrak{q}}_i) \mathcal{K}^T(\underline{\mathfrak{q}}_i)$. The Fisher Information is a measure of the information content of the observations relative to the model parameters. For ease of derivations, we assume $\sigma = \sigma_1 = \sigma_2 = \cdots = \sigma_L$. In this case, we have

$$M_0 = \frac{1}{\sigma^2} \sum_{i=1}^{L} \mathcal{K}(\underline{\mathfrak{q}}_i) \mathcal{K}^T(\underline{\mathfrak{q}}_i) = \frac{1}{\sigma^2} G_0^T G_0 = \frac{1}{\sigma^2} G_0^2 \tag{9}$$

where $G_0 = \Psi^{(0)^T} \Psi^{(0)}$ is the symmetric Gram matrix or the *sensitivity matrix* [8] before adding the new pattern $\underline{\mathfrak{q}}_{L+1}$.

[1]The set $\mathcal{Q}$ could be any data set, e.g. cross-validation in the new environment, for which the relevance information and score is known.

It can easily be shown that the error covariance matrix for the unbiased parameter vector estimate is $C_j = E[(\hat{\underline{\mathfrak{b}}}_j - E[\hat{\underline{\mathfrak{b}}}_j])(\hat{\underline{\mathfrak{b}}}_j - E[\hat{\underline{\mathfrak{b}}}_j])]^T = M_0^{-1}$, i.e. the uncertainty in the parameter estimates on set $\mathcal{Q}$ is proportional to the inverse of the Fisher information matrix (Cramer-Rao Bound). Furthermore, the determinant of C_j is the smallest for BLUE among all estimators of $\underline{\mathfrak{b}}_j$ [9].

Now, having learnt L patterns in pool j we would like to devise a sequential procedure for choosing the next sample $\underline{\mathfrak{q}}_{L+1} \in \mathcal{Q}$ (or basis function), which is most informative to the pool for updating the parameters/weights. The most informative sample introduces the least amount of parameter uncertainties in the scoring function, hence leading to better generalization. That is, the change in the determinant of the covariance matrix C_j should be the minimum or equivalently the change in the determinant of Fisher matrix should be maximum. It is usually more convenient to work with the logarithm of the determinant of the Fisher information matrix. Let $p_L = \log M_0$, then using $M_0 = \frac{1}{\sigma^2} G_0^2$, we have

$$p_L = \log M_0 = 2L \log |\sigma| + 2 \log det(G_0). \tag{10}$$

Incorporating the new pattern $\underline{\mathfrak{q}}_{L+1}$ in the pool changes this measure to

$$p_{L+1} = \log M_1 = 2(L+1) \log |\sigma| + 2 \log det(G_1) \tag{11}$$

where M_1 is the new information matrix and $G_1 = \Psi^{(1)^T} \Psi^{(1)}$ is the Gram matrix after $\underline{\mathfrak{q}}_{L+1}$ is added. The change in the information measure as a result of introducing the new pattern $\underline{\mathfrak{q}}_{L+1}$ is given by

$$g_{L+1} = \log det(G_1) - \log det(G_0). \tag{12}$$

Using the definition of G_1, it can be shown that $det(G_1) = \zeta det(G_0)$. Thus, the information change g_{L+1} after introducing pattern $\underline{\mathfrak{q}}_{L+1}$ can be simplified to

$$g_{L+1} = \log |\zeta| = \log \parallel P^{\perp}_{<\Psi^{(0)}>} \Phi(\underline{\mathfrak{q}}_{L+1}) \parallel^2 \tag{13}$$

where $P^{\perp}_{<\Psi^{(0)}>}$ is the orthogonal complement of the projection matrix $P_{<\Psi^{(0)}>} = \Psi^{(0)} (\Psi^{(0)^T} \Psi^{(0)})^{-1} \Psi^{(0)^T}$.

As can be seen, the change in the determinant of the Fisher information matrix, g_{L+1}, as a result of introducing a new pattern $\underline{\mathfrak{q}}_{L+1}$ is dependent on the *new information* given by ζ. Thus, when the new pattern carries little new information, i.e. is somewhat redundant (small g_{L+1}), more uncertainties are induced to the weight vector of the associated pool and its scoring function. This, in turn jeopardizes the generalization ability of the system and promotes over-fitting. Additionally, it causes more instability and unnecessary expansion of the pool. A sequential sampling algorithm that is designed to avoid these problems is described in Section IV.

III. Database Description and Test Setup

This section provides a brief description of the sonar image data and the feature extraction method employed in this study. This data consists of one training set and one testing set. The training set is used without restriction to initialize the network and train the system so that each pool has a high

TABLE I

OBJECTS FOUND IN THE TRAINING SET IMAGES.

Label	Description	Images per Background
A	Box	47
B	Pipe	148
C	Cone	108
D	Cylinder	147
E	Background Only	200
F	Tire	18

retrieval score for a certain type of pattern. The images in the training set consist of snippets showing several different types of synthetically generated objects at various ranges and aspect angles embedded in a synthetic background. Specifically, in the training set the image snippets contain one of six different object types, namely: A (box), B (pipe), C (cone), D (cylinder), E (background only), and F (tire). Object types B, C, and D are considered targets, while object types A, E, and F are considered nontargets. The aspect angle for each object ranges from 0 to 360 degrees, and the range varies from 10 m to 106 m. The background noise is generated using the 2-D k-distribution [10] and is changed by a parameter, ν, which takes on the values ν = -0.5, 0, 0.5, 1.0, 1.5, 2.0, 3.0, and 5.0. Smaller values of ν produce more cluttered background. The training set contains a total of 5144 image snippets. A description of each object type and the number of images for each object and background type are shown in Table I.

As with the training set, the testing set that is used to evaluate the performance of the trained system consists of image snippets that contain a synthetically generated object embedded in a synthetic background. Object types in the testing set images include A (box), D (cylinder), and C (cone), as with the training set, but an entirely new object type S (sphere) is also present in 1/4 of the testing images. Furthermore, object types B (pipe) and F (tire) are not present in the testing set. The testing image snippets were extracted from larger images that had the objects embedded in one of three different configurations in different backgrounds. This means that each object type only has one of three different shadow lengths and orientations. In particular, type A objects (box) can have a 25°, 45°, or 65° orientation, while type D objects (cylinder) can have a 25°, 45°, or 60° orientation. The backgrounds for the testing images were also generated using the 2-D k-distribution, but with ν values ranging from -0.75 to 0.75 in increments of 0.5. The image snippets in the testing set have significantly more cluttered backgrounds on average when compared to the training set images. These differences in background clutter distributions in addition to the inclusion of the sphere object type are excellent ways to test the system's ability to perform in-situ learning. Overall, there are 108 testing images for each object type, for a total of 432 images.

The proposed CBIR system operates on 43-dimensional feature vectors that were extracted using the contourlet framework discussed in [10]. This method first decomposes an image into a series of multi-resolution, multi-directional shell images. Various statistical measures are then calculated using each of the resulting 18 shell images, which become the different elements of the feature vector used represent the original image. In order to ensure that the feature vectors were compatible with the proposed CBIR system, they were whitened. Whitening is performed on all feature vectors (training and testing) by using only the data from the training set to calculate the necessary mean vector and whitening matrix.

IV. EXPERIMENTAL RESULTS

This section presents the test results obtained by applying the proposed CBIR system to the test image database mentioned in the previous section. More specifically, a network is first initialized using a subset of the images in the training set as exemplar patterns, and the system is then trained by adding the rest of the images to existing pools that meet certain criteria. The classification capability of the system is then tested as it is trained in-situ using a portion (one half) of the testing set. Additionally, the receiver operating characteristic (ROC) curves of system generated after in-situ training concludes are given. Before the results are presented, the testing procedure is described along with several extensions that were made to the system to allow it to function as a classifier for objects in sonar imagery.

A. Training and Testing Procedures

Choosing the proper samples (exemplar patterns) for initializing the pools of neurons in the CBIR network can greatly impact the performance of the system. The exemplar patterns should be representative of the large variations in range, orientation, and type of the objects observed within the database. The network should also be initialized using the same number of exemplar patterns for each object type so that there is no bias of a particular type of image appearing in the top images more often (i.e. considered more relevant due to a higher retrieved score) owing to this type having more representation. This is important for this particular experiment due to the method used for classifying submitted images, which discussed shortly. To facilitate this type of initialization, k-means clustering [11] is performed on groups of patterns representing each object type separately, with the exception of background only images (type E), which are excluded from the initialization and training procedures. The patterns that are closest to the resulting 16 cluster centers are then chosen as exemplar patterns to initialize the pools representing that class of objects. This ensures that, for every training feature vector of a given object type, there is at least one similar pattern of the same type that was used to initialize the network. In other words, the entire training feature space for a given object type is well-represented by the exemplar patterns, hence reducing the risk that the feature vector for a given pattern is closer to those of exemplar patterns of other types than those of the same type. There are 16 pools initialized for each object type since one of them (type F) has only 16 associated images available, and having equal

representation for all object types ensures there is no bias for a certain type during one retrieval.

Once the CBIR network is initialized, model-reference learning is used to train the system on the feature vectors extracted from images in the training set. The input-output relationships of the training samples are captured in the model-reference database that contains the class labels of the training samples. Although the system has access to the entire training image database for batch model-reference training, only one pattern is presented to the network at a time so that the selective sampling method in Section II-D can be used. Note that, due to the use of this selective sampling, not all patterns are added. To adapt the network to a single pattern, it is first submitted to the network to retrieve scores for all the pools in the network. The subset of pools the selected pattern may be added to is restricted to those pools represented by an exemplar pattern that corresponds to the same type as the selected pattern but are not in the top 16 pools (according to their retrieved scores) when the pattern is submitted. Finally, the information content of the selected pattern is evaluated with respect to the low-scoring pools, and the pattern is added to those pools (with a desired score of 0.9) that have a corresponding selective sampling measure that exceeds a predetermined threshold.

To test the proposed CBIR system and determine its effectiveness in a realistic environment which requires in-situ learning, the following testing procedure is adopted. Once learning on all data in the training set is complete, the system is tested on the testing set and the baseline performance is recorded. Next, one pattern is drawn randomly and blindly (without any prior knowledge) from the testing set to acquire its label and submit it to the network to retrieve images. This method of selection simulates a testing environment where a deployed system would not be able to choose an image to use for in-situ learning, but instead would evaluate a newly encountered one. After the image is applied, the network is adapted with its corresponding labeled pattern, if necessary, by using the same criteria as was used in the training phase and the system is retested on the entire testing set. Finally, this process is repeated until either no performance gain is witnessed or until half the samples from the testing set have been used to train the system in-situ.

Since a portion of the images in the testing set contain an object type (sphere) that was not included in the initialization and training of the system, the initially trained network does not have any pools associated with this object type. To accommodate this new object type (type S), a procedure for adding new pools must be used. In this procedure, the first randomly selected pattern that corresponds to an image containing a sphere (or any new object type) will always be used to initialize a new pool. Every subsequent time a sphere pattern is selected the system evaluates the amount of *new information* it would potentially add to existing type S pools, i.e. the selective sampling measure is evaluated for every subsequent pattern. If this measure is sufficiently high for all the existing pools, then a new pool is added. This process continues up until the number of added pools for type S reaches the number of pools associated with every other object type, i.e. 16. It must be noted that, in a practical in-situ learning setting, there may be some new samples that are well-represented by existing pools, and hence are easier to classify than others that are not. Samples that are not well-represented are likely either outliers or represent new object types, which are two circumstances that may be difficult to distinguish. For this reason it is highly desirable to develop an unsupervised version of the selective sampling method that would require an expert operator's involvement for class label assignment only on need basis. This is a topic for future research.

For this study, the classification performance of the in-situ trained CBIR system is evaluated after every in-situ training iteration, i.e. after the network has the opportunity to be trained on one new pattern randomly picked from the testing set. To decide on the class label for a single pattern in the test set using this CBIR system, the pattern is first submitted to the trained system to obtain a retrieved score for every pool in the network. The scores of all the retrieved images corresponding to the target class (representing the different target pools) are then added together and the scores of all the retrieved images corresponding to the nontarget class are added separately. The sum of target scores is then divided by the number of target pools and the same is done for the sum of the nontarget scores. This must be done since, prior to in-situ learning, there are 48 target pools (16 each for object types B, C, and D), while there are only 32 nontarget pools (16 each for object types A and F). Furthermore, as training progresses, new type S pools (nontarget) are added to the network. This weighting of the sum of scores ensures there is no bias for classifying the applied pattern as a target. Finally, the weighted sum of target scores is compared to the weighted sum of nontarget scores and a class label is assigned to the applied pattern that corresponds to the larger score.

B. Classification Results

Figure 3 shows how the correct classification rate (obtained by applying the entire testing set to the CBIR network) changes as in-situ training progresses. As can be seen, after starting from a low value around 0.491 corresponding to the initially trained system, the correct classification rate reaches a value of 0.778 after in-situ learning on only half the samples in the testing data. The Learn++ method in [12], which is an incremental training algorithm for neural networks, was also implemented on the same data set. This method was selected because of its ability to learn from new training samples without losing much of the previously acquired knowledge as well as its ability to accommodate new classes that may be introduced with new training data. However, this method was only able to achieve a correct classification rate of 0.597 on the testing data after in-situ learning (compared to 0.778 for the proposed CBIR system) due to its lack of generalization ability. This classification rate was achieved while adding 10 classifiers to the system. Due to its significantly lower correct classification rates the

remaining results obtained with this Learn++ classifier are omitted.

It is interesting to note that, after in-situ learning, the CBIR system is able to correctly classify the sphere nearly as often as other object types it was initially trained on. This is important in light of the fact that all other objects had a large number of patterns in the original training set. Thus, the main cause of misclassifications on this testing set was the fact that the testing images contained vastly more cluttered background in general than those in the training set. The highly cluttered backgrounds likely caused a large shift in the feature space for the testing images, making them difficult to classify. Figure 3 also shows that, occasionally, the classification performance on the testing set decreases for a period of iterations. This behavior is due to the fact that the correct classification rate for images that the system was not yet trained on temporarily decreased at a rate slightly higher than the rate of increase of the correct classification rate for the images the system was being trained on.

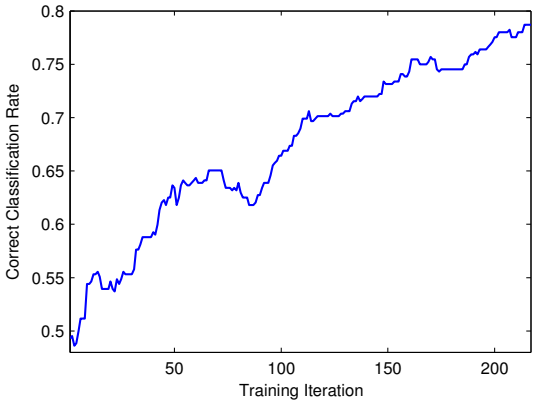

Fig. 3. Correct classification rate during in-situ learning.

The classification results obtained by applying the CBIR system to all the patterns in the testing set after each in-situ training iteration are now presented in terms of the ROC curves, i.e. the plot of the probability of correct classification rate (P_{cc}) versus the probability of false alarm rate (P_{fa}) as a decision threshold is modified. For this study, ROC curves are generated using a procedure that is similar to the one described in the previous subsection to assign class labels to each applied pattern. The only modification to the procedure is that the difference of the sum of target scores and sum of nontarget scores is compared to a variable threshold. By repeating this procedure for each image in the database, P_{cc} and P_{fa} can be calculated for each threshold level.

The evaluation metrics considered here are the area under the ROC curve (AUC) generated using the baseline network (before in-situ learning), the asymptotic AUC value, the rate of AUC increase during in-situ learning, and the "knee-point" of the ROC curve (where P_{cc} + P_{fa} = 1) generated by the system after in-situ learning. The AUC is important since it is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

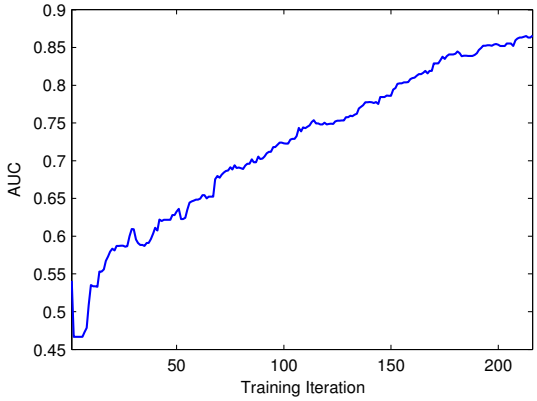

Fig. 4. AUC during in-situ learning.

Figure 4 shows how the AUC changes as in-situ learning progresses. As can be seen the baseline AUC is 0.541 and the asymptotic AUC is 0.866. Figure 5 shows the ROC curve generated using a network that completed in-situ training on half of the testing set, where the knee-point (marked with a '$*$') is at $P_{cc} = 75.9\%$ and $P_{fa} = 24.1\%$. The results of the ROC evaluation are consistent with those of the correct classification rate results presented before in that the AUC increases steadily as training progresses, eventually reaching a satisfactory value. Note that higher correct classification rates could have been obtained by using a larger testing set with more representative features. The fact that the testing set contained a new object type, different target orientations and aspect angles, and significantly more cluttered backgrounds all contributed to making this a very challenging classification problem. Overall, these results demonstrate the ability of the CBIR system to quickly learn the distribution of the features associated with the new object type (sphere) so that images containing them can be correctly classified without initializing or retraining a new network or perturbing the classification performance on other object types.

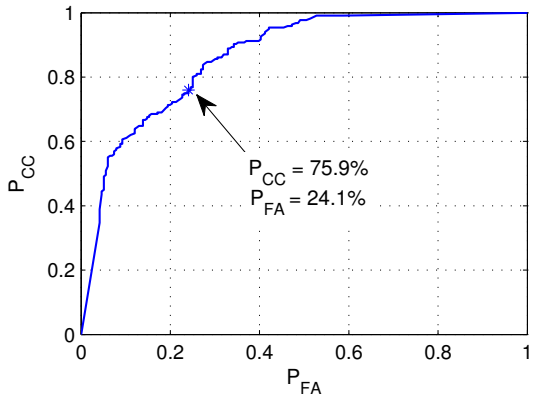

Fig. 5. ROC curve after in-situ learning.

An important aspect of training the CBIR system is the trade-off between obtaining a network that achieves a higher

correct classification rate and one that does not incorporate an overabundance of samples (neurons) in each pool. Some drawbacks of including too many samples include decreased generalization ability and slower response time for the system to retrieve images for a submitted pattern. This trade-off is managed by choosing an appropriate threshold for the selective sampling measure. In particular, if this threshold is increased then there is an increased chance that little to no samples will be added to a given pool, which can potentially lead to less accurate scoring, albeit faster calculation of this score. For this study an appropriate threshold was selected experimentally, based on training data. Alternatively, development and implementation of a scheme for adapting this threshold could provide a more powerful means for managing this trade-off.

In reference to limiting network growth, it must be noted that, despite the fact that a fairly large image database was used to initially train the CBIR system (5144 images total) less than half of these images were actually added to the network. This is due to the use of the selective sampling measure, which ensured that only the most informative samples were added to each pool, thereby allowing the growth of the network to be controlled while at the same time incorporating enough samples such that acceptable classification performance could be achieved. It is also important to note that the drastic increase in classification performance observed during the in-situ learning process was achieved by adding a little more than half the available testing samples (i.e. only 1/4 of the total samples in the testing set since up to half could be used) to the network. It was observed that lowering the selective sampling threshold to include more samples did not increase the overall classification performance of the system when it was applied to the entire testing set. The reason being while incorporating additional samples in the network improved performance slightly on the samples that were randomly selected to be used for in-situ learning, this increase was offset by a decrease in performance on samples the network was never trained on. In other words, lowering the selective sampling measure had a negative impact on the generalization ability of the system, and thus stymies its ability to perform well in real mine-hunting scenarios.

V. Conclusions

This paper discussed a CBIR system used as a classifier for sonar imagery that has the ability to learn the patterns of new objects and adapt to new environmental and operating conditions in-situ by modifying a set of similarity functions in the kernel domain. Learning is implemented using a model-reference mechanism that captures the class and within-class information shared by images in a training database. A structurally dynamic two-layer network is used to implement the retrieval and classification. As learning progresses, pools that receive feedback expand in order to perform similarity function adaptation. To control the expansion of the pools during learning and provide better generalization ability an information-theoretic method was incorporated that only allows the most informative images to be introduced into the pools. The proposed system was then applied to a sonar image classification problem where performance was measured during the in-situ learning process. It was observed that the performance of the system improved dramatically during in-situ learning not only for known object types in different environments, but also for a new object type not used during the initial training. This new object type was incorporated into the system without perturbing the classification performance of the system on existing object types. This increase in classification performance was obtained while at the same time using the selective sampling feature of the system to limit network growth, hence preserving the network's generalization ability.

Acknowledgment

This work was supported by the Office of Naval Research, Small Business Innovative Research Phase I Program under contract #091-066

References

[1] G. J. Dobeck, J. Hyland and L. Smedley, "Automated detection/classification of sea mines in sonar imagery", *Proc. SPIE*, vol. 3079, pp. 90-110, April 1997.

[2] C. Ciany and J. Huang, "Data fusion of VSW CAD/CAC algorithms", *Proc. SPIE*, vol. 4038, pp. 413-420, April, 2000.

[3] M.R. Azimi-Sadjadi, J. Salazar, and S. Srinivasan, "An adaptable image retrieval system with relevance feedback using kernel machines and selective sampling", *IEEE Trans. on Image Processing*, vol. 18, No. 7, pp. 1645-1659, July 2009.

[4] B. Scholkopf and S. Smola, *Learning with kernels*, The MIT press, 2002.

[5] N. Vasconcelos and A. Lippman, "A Bayesian framework for content-based indexing and retrieval," *Proc. Data Compression Conf.*, pp. 580, 1998.

[6] P. Muneesawang and L. Guan, "An interactive approach for CBIR using a network of radial basis functions," *IEEE Trans. Multimedia*, vol. 6, no. 10, pp. 703716, Oct. 2004.

[7] E. Dura, Y. Zhang, X. Liao, G.J. Dobeck and L. Carin, "Active learning for detection of mine-like objects in side-scan sonar imagery", *IEEE Journal of Oceanic Engr.*, vol. 30, no. 2, pp. 360-371, April 2005.

[8] L.L. Scharf and L. T. McWhorter, "Geometry of the Cramer-Rao bound", *Signal Processing*, vol. 31, no. 3, pp. 301-311, April 1993.

[9] V. V. Fedorov, *Theory of optimal experiments*, Academic Press, Inc, 1972.

[10] J.-E.Wilbur, R.J. McDonald, and J. Stack, "Contourlet detection and feature extraction for automatic target recognition", *IEEE International Conf. on Systems, Man and Cybernetics*, pp. 2734-2738, Oct 2009.

[11] D. MacKay, "Chapter 20. An example inference task: clustering", *Information Theory, Inference and Learning Algorithms*, Cambridge University Press.

[12] R. Polikar, L. Udpa, and S. Udpa, "Learn++: an incremental learning algorithm for supervised neural networks", *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 31, pp. 497-508, November 2001.