An Adaptable Image Retrieval System With Relevance Feedback Using Kernel Machines and Selective Sampling

Mahmood R. Azimi-Sadjadi, *Senior Member, IEEE*, Jaime Salazar, and Saravanakumar Srinivasan

***Abstract*—This paper presents an adaptable content-based image retrieval (CBIR) system developed using regularization theory, kernel-based machines, and Fisher information measure. The system consists of a retrieval subsystem that carries out similarity matching using image-dependant information, multiple mapping subsystems that adaptively modify the similarity measures, and a relevance feedback mechanism that incorporates user information. The adaptation process drives the retrieval error to zero in order to exactly meet either an existing multiclass classification model or the user high-level concepts using *reference-model* or *relevance feedback* learning, respectively. To facilitate the selection of the most informative query images during relevance feedback learning a new method based upon the Fisher information is introduced. Model-reference and relevance feedback learning mechanisms are thoroughly tested on a domain-specific image database that encompasses a wide range of underwater objects captured using an electro-optical sensor. Benchmarking results with two other relevance feedback learning methods are also provided.**

***Index Terms*—Content-based image retrieval, Fisher information matrix and selective sampling, *in-situ* underwater target identification, kernel machines, regularization, relevance feedback learning.**

I. Introduction

RELEVANCE feedback originally developed for text retrieval systems, has found wide spread acceptance in content-based image (CBIR) systems [1]–[6]. The challenging goal for CBIR is to capture, via the user relevance feedback, high-level semantic user concepts when only low-level image features (e.g., shape, color, and texture) are applied as queries. CBIR systems utilize relevance feedback to dynamically modify either the original image query [4], [5] or the similarity measure [1], [6] in order to meet users' requirements. The systems that rely on query modification for relevance feedback, such as the Multimedia Analysis and Retrieval System (MARS) [5], are inspired from their counterparts in the text retrieval area. In these systems, the original query is modified using both relevant and nonrelevant images to achieve improved retrieval. The query modification process via relevance feedback is typically interactive involving several feedback rounds until the query refinements eventually result in a list of images that most closely carry the required concept. There are many problems with this approach. For CBIR systems, identification of an optimum query image that captures the desired concepts could be a difficult task. Moreover, different users may have different specific concepts of interests even for the same query image. It is also possible that during this tedious and time consuming process the users loose patience in selecting the relevant and nonrelevant images, hence resulting in inaccurate or inconsistent association. More importantly, an acceptable solution may not be reached since there is generally no direct mapping between the low-level feature space and the hidden high-level user concepts. An example of CBIR that modifies the similarity measure is the method in [1], which uses a boosting algorithm to learn the boundary between positive and negative samples using a labeled training set. The results are benchmarked with the boundary region generated by a support vector machine (SVM) [7]. Although this algorithm is found to be useful for some photo collections, its effectiveness remains in doubt in presence of noise and image degradation [8].

New relevance feedback learning methods have recently been proposed among which SVM [9]–[12], Bayesian inference [13], radial basis functions (RBFs) [14] are perhaps the most popular ones. The CBIR system in [9] uses a relevance feedback learning based upon SVM to separate positive feedback samples from the negative ones given that only the positive samples are known. The assumption is that positive samples cluster in a certain way but negative ones do not, as they can potentially belong to any class or group. Although they reported good results using only a small number of positive training samples, the performance of the method relies on proper selection of some parameters. Moreover, there is no control on the relevancy score of the positive samples as all of them are treated equally. The method in [11] addresses the sample poor problems in the SVM-based CBIR. These problems occur since the number of labeled positive samples is typically small comparing to the negative ones and to the dimension of the feature space. To remedy these problems a combination of asymmetric bagging and random subspace SVM relevance feedback learning is adopted. The method in [12] uses the idea that user selected negative samples for relevance feedback form several sub-clusters while the positive ones group only in one cluster. Based upon this idea, kernel marginal convex machines are developed and

Manuscript received October 02, 2008; revised February 20, 2009. First published May 12, 2009; current version published June 12, 2009. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Dan Schonfeld.

The authors are with the Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523 USA (e-mail: azimi@engr.colostate.edu).

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

Digital Object Identifier 10.1109/TIP.2009.2017825

used together with the relevance feedback. The results indicated that this partitioning indeed improves the overall retrieval accuracy. In [15], an orthogonal complement component analysis method was introduced that captures the homogeneous concepts common to all the positive samples. Experimental results are presented to demonstrate superior performance of the proposed method in comparison with those of the linear and kernel principal component analysis method. The CBIR system in [16] exploits marginal Fisher analysis to characterize the *between-class* separability by the margin between positive and negative samples while the *within-class* compactness is characterized only by the distances between each positive sample and its adjacent positive samples. An efficient method for finding the optimal solution for marginal Fisher analysis was also proposed.

Among the wide variety of existing methods that use relevance feedback in CBIR, perhaps the ones most closely related to the present study, are those in [14] and [17]. In [14], the authors introduced an RBF-like scoring using a sum of univariate Gaussian-shaped functions centered at individual features of the query image. The standard deviation for each individual query feature is estimated heuristically by determining the relevance of the feature to the user concept. The input query then goes through a series of modifications in order to drive the query closer to the relevant images and farther from the nonrelevant ones. After convergence, the final query is used in the RBF-like scoring function to compute the score of the images in the database. Although the proposed RBF-based scoring function is easy to implement, some of the desirable properties of RBF as function approximator are not fully exploited owing to the oversimplification of the adopted similarity function. The method in [17] uses the kernel extension of the biased discriminant analysis (BDA) for relevance feedback in CBIR. Although the BDA solves the problems of linear discriminant analysis (LDA) by utilizing the fact that positive and negative samples are clustered differently, it suffers from small sample size data. To overcome this problem, the authors proposed a direct kernel BDA, which solves the singularity and ill-conditioning problems without using regularization. An incremental version of the method was also introduced for more efficient implementation. The results on Corel collection show improved results over the SVM-type relevance feedback.

In this paper, we present a new CBIR system based upon regularization theory, kernel machines and Fisher information measure. The proposed CBIR is developed for applications and databases where it is desired not only to capture expert user concepts and meet exact user-specified scoring requirements but also discover certain class or within-class attributes in the data. The system incorporates user concepts using an online single-round relevance feedback learning or infers the hidden class (or within-class) information using a batch mode model-reference learning. The system consists of multiple *implicit* mapping subsystems which implement similarity function modification in a high-dimensional feature space. A structurally adaptable kernel machine is developed to efficiently implement the similarity function adaptation using the *kernel trick* in response to either the user relevance feedback or multiclass requirements. Similarity function adaptation is localized to only those images that receive relevance feedback, hence requiring limited parameter updating. The proposed single-round relevance feedback learning meets either binary relevance (positive and negative) or exact desired scores for user selected relevant and nonrelevant images. Additionally, the results of model-reference or relevance feedback learning are saved for future usage by other users. Geometric interpretation and stability of the relevance feedback learning are also studied. To select the most informative query image for optimal parameter adaptation during relevance feedback, a new *selective sampling* method based upon the Fisher information matrix is developed. The applicability and effectiveness of the proposed adaptable CBIR is demonstrated on a special-purpose electro-optical imagery database of underwater objects. The goal of the relevance feedback in this specific application is to use expert operator's feedback to provide *in-situ* learning capability essential for maintaining underwater target identification performance in new operating and environmental conditions. The proposed relevance feedback learning is then benchmarked against two other algorithms namely those in [2] and [14].

The organization of this paper is as follows. Section II presents an overview of a typical retrieval system and then introduces the proposed CBIR, its components, generalized similarity function and initialization process. Section III introduces the model-reference learning and its kernel-based implementation. Relevance feedback learning and its kernel implementation, geometric interpretation, stability, and selective sampling method are given in Section IV. Section V presents the results of the proposed learning methods and their benchmarking with those in [14] and [2]. Finally, Section VI presents the conclusions and observations on the proposed adaptable CBIR.

II. Adaptable CBIR and Generalized Similarity Function

In a typical IR system, the retrieval and search process performs, upon the user request or query, a similarity matching (or scoring) function $s(\underline{\mathrm{q}}, \underline{\mathrm{x}}_j)$ between the submitted query image $\underline{\mathrm{q}}$ and every image vector $\underline{\mathrm{x}}_j$, $j \in [1, N]$ in the database represented by matrix $X = [\underline{\mathrm{x}}_1 \underline{\mathrm{x}}_2 \cdots \underline{\mathrm{x}}_N]$ and then delivers $n \leq N$ closest matches (or scores). The M-dimensional vector $\underline{\mathrm{x}}_j$ consists of the extracted features (e.g., shape, texture, color, etc.) of the jth image in the database X. A commonly used similarity function is the generalized Euclidean distance measure $s(\underline{\mathrm{q}}, \underline{\mathrm{x}}_j) = (\underline{\mathrm{q}} - \underline{\mathrm{x}}_j)^T W (\underline{\mathrm{q}} - \underline{\mathrm{x}}_j)$ where W is typically a diagonal matrix that weights each feature in the similarity measure [5], [6]. The search results in an ordered list of retrieved relevant and nonrelevant images arranged according to their distance to the submitted query image. Equivalently, one can use similarity function $s(\underline{\mathrm{q}}, \underline{\mathrm{x}}_j) = \underline{\mathrm{x}}_j^T W \underline{\mathrm{q}}$ in which case similarity is ordered as a decreasing measure between the query and the listed images. Clearly, the relative score of each listed image is a function of the adopted similarity function and the model used by the particular IR system [18]. In the CBIR systems that use similarity measure modification, one may dynamically modify the elements of matrix W in an attempt to meet the user requirements by modifying the importance of each feature to the similarity function [19]. However, this does not necessarily guarantee the capture of user high-level semantic concepts. In this

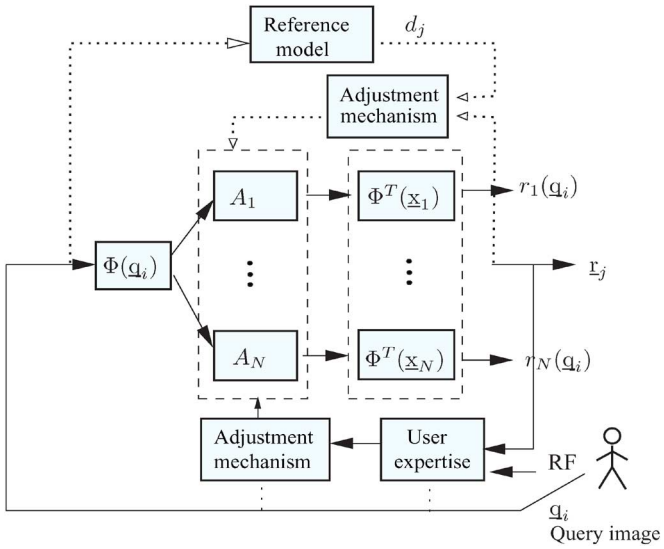


Fig. 1. Proposed adaptable CBIR system and components.

section, we present an adaptive CBIR system that attempts to automatically capture this information in a high-dimensional feature space.

The structure of the proposed CBIR system is shown in Fig. 1. It contains four main components namely the nonlinear mapping function $\Phi(.) : \mathbb{R}^M \rightarrow \mathbb{R}^{N_f}$ with $N_f \gg M$ that maps the query image to high-dimensional feature space, a series of *implicit* mapping matrices A_j's that define the similarity matching functions, a retrieval system trained using the database X, model-reference learning for situations when class membership may be available, and user relevance feedback learning. The proposed CBIR captures high-level user concepts not only by a series of adaptable mapping matrices A_j (one for every image in the database) but also by proper choice of the nonlinear mapping function $\Phi(.)$.

A. Generalized Similarity Function

As can be seen from Fig. 1, upon submitting query image $\underline{q}$, the system maps this query into the high-dimensional feature space using $\Phi(\underline{q})$, performs a linear mapping in this higher dimensional space, i.e., $\hat{\underline{q}} = A_j\Phi(\underline{q})$, and then generates the retrieval score or similarity measure for the j^{th} image as given by

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

where $\Phi(\underline{q}) = [\phi_1(\underline{q})\ \phi_2(\underline{q})\ \ldots\ \phi_{N_f}(\underline{q})]^T$ with $\phi_j(.)$'s being scalar functions of the query image $\underline{q}$, and similarly for $\Phi(\underline{x}_j)$. Thus, comparing to the standard similarity functions mentioned above or the similarity function modification method in [6], in our system the mapping function $\Phi(.)$ maps both the query image $\underline{q}$ and the j^{th} image $\underline{x}_j$ to the higher dimensional space, in which matrix A_j then shapes the matching space for every image $\underline{x}_j$, $j \in [1, N]$. The mapping to high-dimensional feature space together with multiple function adaptability provides much better opportunity to capture the hidden user concepts. Note that even in absence of the nonlinear mapping, i.e., $\Phi(\underline{x}) = \underline{x}$, the similarity function $r_j(\underline{q}) = \underline{x}_j^T A_j \underline{q}$ is still more general than the standard generalized Euclidean distance measure, $r_j(\underline{q}) = \underline{x}_j^T W \underline{q}$, since for every image, $\underline{x}_j$, in the database a specific nondiagonal matrix A_j is used.

At first glance it may appear that finding a matrix for every image in the database, X, is too inefficient and impractical, especially in the large (possibly infinite) dimensional mapped feature space. However, we show later that this is entirely avoided using the *kernel trick*. Additionally, in Section III, we show how the proposed CBIR system in Fig. 1 and all the learning rules can be implemented using a simple structurally adaptive kernel machine.

B. Initialization Process

Initial setup or system initialization is required in any CBIR to train the system 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. In our system the initial setup of similarity functions in (1) involves choosing initial mapping matrices A_j, $j \in [1, N]$ given *only* the image database X of 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 $A_j = P_{\Phi(\underline{x}_j)}$ where $P_{\Phi(\underline{x}_j)} = \Phi(\underline{x}_j)[\Phi^T(\underline{x}_j)\ \Phi(\underline{x}_j)]^{-1}\Phi^T(\underline{x}_j)$, for every image $\underline{x}_j$. Although, these two choices for A_j matrices produce the same similarity function $s(\underline{q}, \underline{x}_j) = \Phi^T(\underline{x}_j)\Phi(\underline{q})$, the projection matrix is a preferred choice as its energy $\xi = tr(P^T_{\Phi(\underline{x}_j)}P_{\Phi(\underline{x}_j)}) = tr(P^2_{\Phi(\underline{x}_j)}) = 1$, where $tr(.)$ is the trace operator, as opposed to that of the identity matrix $\xi = N_f$. In other words, for the projection matrix the energy is concentrated in only one eigenvector $\Phi(\underline{x}_j)$ with the associated unity eigenvalue, while the energy of the identity matrix is scattered among N_f eigenvectors. For kernel producing functions (see Section III), the initial similarity function becomes simply $s(\underline{q}, \underline{x}_j) = \Phi^T(\underline{x}_j)\Phi(\underline{q}) = k(\underline{x}_j, \underline{q})$, which can easily be computed in the kernel domain. The following section describes how matrices A_j's can be updated using either model-reference or relevance feedback learning.

III. Model-Reference Learning

The goal of the model-reference learning is to learn the input-output mapping of a *reference model* for an ensemble set of query images, the corresponding listed images and their desired scores. The dotted lines in Fig. 1 indicate the flow of information and the subsystems involved in the model-reference learning, which may be used to capture some specific *a priori* class and within-class information shared by the images in the database. The *within-class* information refers to specific details that separate different images within the same class and is typically available in most specific-purpose databases. The goal is to make sure that for a submitted image query, all the images of the same class and within-class are listed at the top according to their desired score. This process could be done prior to the user relevance feedback in order to capture all the prior class or within-class information 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 class labels. Every image in the database $\{\underline{x}_j\}_{j=1}^N$ is first submitted as a query image, i.e $\underline{q} = \underline{x}_j$ for $j \in [1, N]$, retrieved

images are automatically (without user involvement) evaluated against the known class information. If the score of a relevant image needs to be adjusted then a *desired* score (preselected) is set and stored in the training database along with the submitted query. Once all the query images are submitted, the information in the model-reference training database is used to train the CBIR system in a batch mode. This is discussed next.

Given a set of L training pairs $\{\underline{\mathrm{q}}_i, d_j^{(i)}\}_{i=1}^{L}$ in the model-reference database, where $\underline{\mathrm{q}}_i$ is the i^{th} query image and $d_j^{(i)}$ is the desired score for image $\underline{\mathrm{x}}_j$, the goal of the model-reference learning is to find new implicit mapping matrix (not explicitly computed) $\hat{A}_j$ and the mapping function $\Phi(.)$ to yield the desired score for selected image $\underline{\mathrm{x}}_j$, $j \in [1, N]$. Using the proposed generalized similarity function, the problem can be posed as finding $\hat{A}_j$ and $\Phi(.)$ such that

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

The problem is ill-posed because many choices for mapping matrices $\hat{A}_j$ and function $\Phi(.)$ can potentially satisfy this requirement. However, we can restrict the choice of the basis eigenfunctions $\phi_l(.)$'s to those that generate positive and symmetric kernel $k(\underline{\mathrm{s}}, \underline{\mathrm{t}}) = k(\underline{\mathrm{t}}, \underline{\mathrm{s}})$ satisfying Mercer's theorem [7], then $k(\underline{\mathrm{s}}, \underline{\mathrm{t}}) = \sum_{l=1}^{N_f} \phi_l(\underline{\mathrm{s}})\phi_l(\underline{\mathrm{t}}) = \Phi^T(\underline{\mathrm{s}})\Phi(\underline{\mathrm{t}})$. Thus, the problem reduces to that of finding matrix $\hat{A}_j$ using the chosen basis functions. This problem is also ill-posed because if matrix $\hat{A}_j$ is a solution to (2) so is $\hat{A}_j + B_j$, where $\Phi(\underline{\mathrm{x}}_j) \in \mathcal{N}ull(B_j)$, i.e., in null space of B_j. Therefore, a solution $\hat{A}_j$ with minimum Frobenius norm requires that each column be a multiple of $\Phi(\underline{\mathrm{x}}_j)$ leading to a rank-one matrix

$$\hat{A}_j = \Phi(\underline{\mathrm{x}}_j)\underline{\mathrm{w}}_j^T \tag{3}$$

with some weight vector $\underline{\mathrm{w}}_j$. Now, the problem can be converted into a well-posed one by using the regularized least squares (LS) [7] for finding optimal $\underline{\mathrm{w}}_j^*$ such that

$$\underline{\mathrm{w}}_j^* = arg \min_{\underline{\mathrm{w}}_j} \mathcal{T}(\underline{\mathrm{w}}_j, \lambda) \tag{4}$$

where $\mathcal{T}$ is the regularized LS cost function [20], [21] given by

$$\begin{aligned}\mathcal{T}(\underline{\mathrm{w}}_j,\lambda) &= \frac{1}{2}\sum_{i=1}^{L}\left(d_j^{(i)} - \Phi^T(\underline{\mathrm{x}}_j)\Phi(\underline{\mathrm{x}}_j)\underline{\mathrm{w}}_j^T\Phi(\underline{\mathrm{q}}_i)\right)^2 + \frac{1}{2}\lambda\|\underline{\mathrm{w}}_j\|^2 \\ &= \frac{1}{2}\|\underline{\mathrm{d}}_j - c_j\Psi^T\underline{\mathrm{w}}_j\|^2 + \frac{1}{2}\lambda\|\underline{\mathrm{w}}_j\|^2. \end{aligned} \tag{5}$$

Here, $\underline{\mathrm{d}}_j = [d_j^{(1)}\ d_j^{(2)}\ \ldots\ d_j^{(L)}]^T$ is the vector of desired scores for all query images $\underline{\mathrm{q}}_i$, $i \in [1, L]$, $c_j = \Phi^T(\underline{\mathrm{x}}_j)\Phi(\underline{\mathrm{x}}_j) = k(\underline{\mathrm{x}}_j, \underline{\mathrm{x}}_j)$ and $\Psi = [\Phi(\underline{\mathrm{q}}_1)\Phi(\underline{\mathrm{q}}_2)\ \ldots\ \Phi(\underline{\mathrm{q}}_L)]$ is the matrix containing all the mapped queries. The first term in (5) 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 be found as

$$w_j^* = \frac{1}{c_j}\Psi\left(\Psi^T\Psi + c_j^{-2}\lambda I\right)^{-1}\underline{\mathrm{d}}_j. \tag{6}$$

As can be seen from (6), 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 $g_{l,m} = k(\underline{\mathrm{q}}_l, \underline{\mathrm{q}}_m)$. The diagonal loading depends on λ and $c_j^{-2} = 1/k^2(\underline{\mathrm{x}}_j, \underline{\mathrm{x}}_j) \neq 0$ and guarantees the existence of the inverse for matrix $(\Psi^T\Psi + c_j^{-2}\lambda I)$. Note that for symmetric, PD and nondegenerate [22] kernel functions the Gram matrix is nonsingular even without the diagonal loading term. An example of a nondegenerate kernel is the Gaussian kernel $k(\underline{\mathrm{s}}, \underline{\mathrm{t}}) = e^{-a\|\underline{\mathrm{s}}-\underline{\mathrm{t}}\|^2}$, where a is a constant and positive real number. Thus, for nondegenerate kernels the regularization term may be avoided.

Once the optimal solutions $\underline{\mathrm{w}}_j^*$, $j \in [1, N]$'s for all the query images in the model-reference training database are computed the new similarity (or scoring) function, $r_j(\underline{\mathrm{q}})$, for a new query $\underline{\mathrm{q}}$ in (1) can easily be determined by using the optimal vector $\underline{\mathrm{w}}_j^*$ in (6) and (3). This yields

$$\begin{aligned} r_j(\underline{\mathrm{q}}) &= \Phi^T(\underline{\mathrm{q}})\Psi\left(\Psi^T\Psi + c_j^{-2}\lambda I\right)^{-1}\underline{\mathrm{d}}_j \\ &= \sum_{i=1}^{L} b_{ji}k(\underline{\mathrm{q}}, \underline{\mathrm{q}}_i) = \mathcal{K}^T(\underline{\mathrm{q}})\underline{\mathrm{b}}_j \end{aligned} \tag{7}$$

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

This result implies that the new similarity function in (7) for the model-reference learning can be implemented in the kernel domain without the need to explicitly compute $\underline{\mathrm{w}}_j^*$. Note that for most kernels c_j is a constant (e.g., for Gaussian $c_j = 1$). Thus, only one matrix inversion is needed to compute all $\underline{\mathrm{b}}_j$'s in (7). Additionally, there are recursive algorithms for computing the inverse of a Gram matrix (see Appendix I) that don't require any matrix inversion operation.

Now, we show how the system in Fig. 1 and the associated model-reference and relevance feedback learning mechanisms can be implemented using a two-layer kernel-based network in Fig. 2. This network consists of N pools of neurons, one pool for every image in the training database. Initially, each pool is formed of only one neuron corresponding to an image in this database. That is, when initializing the j^{th} pool, image $\underline{\mathrm{q}}_1 = \underline{\mathrm{x}}_j$ is incorporated into the pool by setting neuron kernel function to $k(\underline{\mathrm{q}}, \underline{\mathrm{q}}_1)$ for any input query $\underline{\mathrm{q}}$. The weight connecting this neuron to the corresponding output neuron is also initially set to $b_{j1} = 1$. Thus, the initial scoring generated by the j^{th} pool for query $\underline{\mathrm{q}}$ is $r_j(\underline{\mathrm{q}}) = k(\underline{\mathrm{q}}, \underline{\mathrm{q}}_1)$, which implements the initialization procedure in Section II-B. Once initially setup, the model-reference (or relevance feedback) learning results in expansion of the pools and updating the weight vector $\underline{\mathrm{b}}_j$ of the second layer. Fig. 2 shows how pool j is expanded by incorporating $L-1$ additional neurons with kernel functions $k(\underline{\mathrm{q}}, \underline{\mathrm{q}}_i), i \in [2, L]$. In essence, all queries $\underline{\mathrm{q}}_i$'s, $i \in [1, L]$, captured by pool j, are now associated with image $\underline{\mathrm{x}}_j$ in the database and they form the basis for representing the scoring function $r_j(\underline{\mathrm{q}}) = \underline{\mathrm{b}}_j^T\mathcal{K}(\underline{\mathrm{q}})$ in (7) at the output of pool j for the submitted query, $\underline{\mathrm{q}}$. The scoring function adaptation also requires updating the weight vector $\underline{\mathrm{b}}_j$ using the desired score information. In this kernel machine, a pool of neurons captures high-level hidden associations among

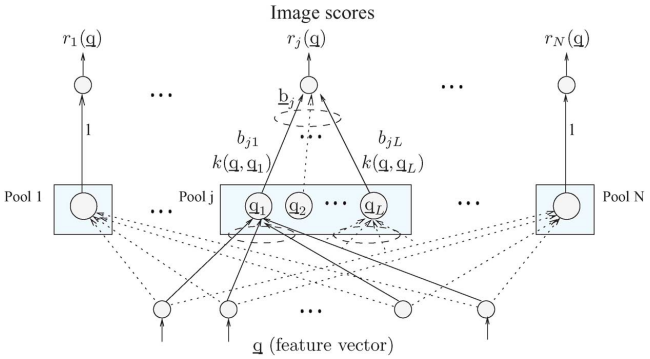


Fig. 2. Practical network for the j^{th} scoring function.

some of images in the database that have certain common characteristics. In the next section, we show how the same network can capture user concepts by adding neurons to the pools corresponding to the user selected images.

IV. Relevance Feedback Learning via Structural Adaptation

As stated before, relevance feedback learning is needed to incorporate users' high-level concepts into the retrieval system using an online learning procedure. For every query image, the retrieved relevant and nonrelevant images are evaluated and the user can either promote one or more images by raising their relative scores (positive feedback) or demote other images by lowering their scores (negative feedback). This visual evaluation of the retrieved images does not usually consider the knowledge of the classes as this could be a difficult task for the user, especially when the categorization of the images is not-well defined, e.g., an image may belong to two or more classes. If the user only decides the desired positions of the selected images then the system computes the desired scores internally. These are then used to generate error signals to update the parameters of the appropriate pools such that if the same (or similar) image query is applied the assigned desired score is reproduced.

In the proposed adaptive CBIR, single-round relevance feedback is implemented using a simple structural adaptation that involves neuron expansion in one or more pool(s) of the network in Fig. 2. Only those pools corresponding to the selected images undergo node expansion. This single-round process applies feedback simultaneously to all the relevant images without requiring the user to engage in reevaluation of the retrieved images. The goal here is to capture an underlying similarity function that assigns desired scores to the user selected images, without perturbing the previous learning, in only one feedback round. Because the relevancy information is only influencing certain similarity functions in the overall retrieval system, the incorporation of information is highly localized. We treat this user feedback learning as an *on-line sequential* process described below.

Assume that at time t_0, queries $\underline{q}_1, \underline{q}_2, \ldots, \underline{q}_L$ have already been learned (e.g., either via model-reference or relevance feedback learning) and stored in pool j, as shown in Fig. 2, and now at time t_1 a new query image $\underline{q} = \underline{q}_{L+1}$ is applied. For this query, suppose the user selects image j as the most relevant image and assigns a desired score $d_j^{(L+1)}$. Let $\underline{w}_j^{(0)}$ and $\underline{d}_j^{(0)}$ be the old weight and desired score vectors for L previously learned queries and $\underline{w}_j^{(1)}$ and $\underline{d}_j^{(1)} = [\underline{d}_j^{(0)^T} d_j^{(L+1)}]^T$ be the new weight vector and desired score vector at time t_1, respectively. It is desirable to generate the new weight vector, $\underline{w}_j^{(1)}$, by updating the old weight vector $\underline{w}_j^{(0)}$, i.e., $\underline{w}_j^{(1)} = \underline{w}_j^{(0)} + \Delta\underline{w}_j$ with the aim of producing the new score and at the same time maintaining the scores of L previously stored images. This problem can also be cast as a regularized LS with the cost function

$$\mathcal{T}(\Delta\underline{w}_j, \lambda) = \frac{1}{2}\left\|\underline{d}_j^{(1)} - c_j \Psi^{(1)^T}\left(\underline{w}_j^{(0)} + \Delta\underline{w}_j\right)\right\|^2 + \frac{1}{2}\lambda\|\Delta\underline{w}_j\|^2 \tag{8}$$

where $\Psi^{(1)} = [\Psi^{(0)} \Phi(\underline{q}_{L+1})]$ is the augmented mapped data matrix. The optimum solution of this regularized LS for $\Delta\underline{w}_j^*$ is given by

$$\Delta\underline{w}_j^* = \frac{1}{c_j}\Psi^{(1)}\left(\Psi^{(1)^T}\Psi^{(1)} + c_j^{-2}\lambda I\right)^{-1}\left[\underline{d}_j^{(1)} - c_j \Psi^{(1)^T}\underline{w}_j^{(0)}\right]. \tag{9}$$

Now, using (7) in $\underline{w}_j^* = \underline{w}_j^{(0)} + \Delta\underline{w}_j^*$ and assuming $\lambda = 0$ (nondegenerate kernels), the new scoring function of the j^{th} image for query $\underline{q}$ becomes

$$r_j^{(1)}(\underline{q}) = r_j^{(0)}(\underline{q}) + \mathcal{K}^{(1)^T}(\underline{q})\left(\Psi^{(1)^T}\Psi^{(1)}\right)^{-1} \times \left[\underline{d}_j^{(1)} - c_j \Psi^{(1)^T}\underline{w}_j^{(0)}\right] \tag{10}$$

where $\mathcal{K}^{(1)}(\underline{q}) = [k(\underline{q},\underline{q}_1) \ \ldots \ k(\underline{q},\underline{q}_{L+1})]^T = [\mathcal{K}^{(0)^T}(\underline{q}) k(\underline{q},\underline{q}_{L+1})]^T$ and $r_j^{(0)}(\underline{q}) = c_j \Phi(\underline{q})^T \underline{w}_j^{(0)}$ is the previous scoring function. It is interesting to note that $\hat{\underline{d}}_j^{(1)} := c_j \Psi^{(1)^T}\underline{w}_j^{(0)}$ in the adjustment term of (10) represents an estimate of the desired score vector $\underline{d}_j^{(1)}$ using the old weight vector $\underline{w}_j^{(0)}$ and the new augmented query data matrix $\Psi^{(1)}$.

In Appendix I, it has been shown that the scoring function in (10) can be reduced to the following recursive updating equation:

$$r_j^{(1)}(\underline{q}) = r_j^{(0)}(\underline{q}) + \delta_j \frac{k(\underline{q},\underline{q}_{L+1}) - \sum_{i=1}^{L} k(\underline{q},\underline{q}_i) z_i}{k(\underline{q}_{L+1},\underline{q}_{L+1}) - \sum_{i=1}^{L} k(\underline{q}_{L+1},\underline{q}_i) z_i} \tag{11}$$

where $\delta_j := d_j^{(L+1)} - r_j^{(0)}(\underline{q}_{L+1})$ is the *prediction error* between the desired score $d_j^{(L+1)}$ assigned by the user and the predicted value $r_j^{(0)}(\underline{q}_{L+1}) = c_j \Phi^T(\underline{q}_{L+1})\underline{w}_j^{(0)}$ by the retrieval system (before updating) for the new query $\underline{q}_{L+1}$, and z_i is the i^{th} coefficient in the expansion

$$k(\underline{q},\underline{q}_{L+1}) \approx \sum_{i=1}^{L} k(\underline{q},\underline{q}_i) z_i \approx \mathcal{K}^{(0)^T}(\underline{q})\underline{z} \tag{12}$$

which shows how $k(\underline{q},\underline{q}_{L+1})$ is approximated by linear combination of kernel functions $\{k(\underline{q},\underline{q}_1)k(\underline{q},\underline{q}_2)\ldots k(\underline{q},\underline{q}_L)\}$. Thus, the adjustment term in (11) is dependent on the function-expansion errors for new query $\underline{q}_{L+1}$ (denominator) and applied query $\underline{q}$ (numerator) as well as the prediction error δ_j. Note that all the operations in this updating equation are carried out in the kernel domain.

A recursive kernel-based updating equation for the new weight vector $\underline{b}_j^{(1)}$ is also derived in Appendix I [see (A5)], which shows how L previous weights in $\underline{b}_j^{(0)}$ are updated and the connection weight to the newly added $(L+1)$th neuron (see Fig. 2), i.e., last element of $\underline{b}_j^{(1)}$, is found. As mentioned before, when the network is initialized, there is only one neuron in each pool with connection weight $b_j^{(0)} = 1$.

Remark 1: For the Gaussian kernel $k(\underline{s},\underline{t}) = e^{-a\|\underline{s}-\underline{t}\|^2}$, if parameter $a \to \infty$, $k(\underline{s},\underline{t}) \to \delta(\underline{s}-\underline{t})$, i.e., reduces to delta function and, hence, $r_j^{(0)}(\underline{q}_{L+1}) \to 0, \delta_j \to d_j^{(L+1)}$, and $\underline{z} \to \underline{0}$. In this case, the scoring function (11) simply becomes

$$r_j^{(1)}(\underline{q}) \approx \sum_{i=1}^{L+1} d_j^{(i)} \delta(\underline{q}-\underline{q}_i) \tag{13}$$

which is a *strict interpolation* scoring function at queries $\underline{q} = \underline{q}_i$, $i \in [1, L+1]$. Although the scoring function (13) is interesting, since it does not require any recursive updating, its low prediction capability makes it of limited usage. Hence, large values of a should be avoided as they lead to poor generalization performance on the novel testing data (see Experiment 3 results in Section V).

A. Stability of Relevance Feedback Learning

The stability in the context of the proposed relevance feedback implies that as learning progresses the information in the new queries can be incorporated into the system without jeopardizing the previously learned information. This is an important issue, especially if the system receives abundant feedback and one needs to maintain the performance on the previously learned query images.

To show stability of our image retrieval system, let us assume that the last query image learned was $\underline{q}_{L+1}$. Using the updated scoring function (11), the score for this query is $r_j^{(1)}(\underline{q}_{L+1}) = r_j^{(0)}(\underline{q}_{L+1}) + \delta_j$. However, since $\delta_j = d_j^{(L+1)} - r_j^{(0)}(\underline{q}_{L+1})$, we get $r_j^{(1)}(\underline{q}_{L+1}) = d_j^{(L+1)}$, i.e., the desired score assigned by the user is exactly met for this newly learned information query. Next, we need to show that the previous information is also retained for all the previous queries $\underline{q}_k$, $k \in [1, L]$. Applying the results in Appendix II and modifying the superscripts in (11) yields

$$r_j^{(l+1)}(\underline{q}) = r_j^{(l)}(\underline{q}) + \delta_j \frac{\left\| P^{\perp}_{\langle \Psi^{(0)} \rangle} \Phi(\underline{q}) \right\| cos\theta}{\left\| P^{\perp}_{\langle \Psi^{(0)} \rangle} \Phi(\underline{q}_{L+1}) \right\|} \tag{14}$$

where $r_j^{(l+1)}(\cdot)$ is the j^{th} scoring function at time $l+1$ and θ is the angle between $\tilde{\Phi}(\underline{q})$ and $\tilde{\Phi}(\underline{q}_{L+1})$. If we query the system with one of the previous queries $\underline{q} = \underline{q}_k k \in [1, L]$, the numerator of the adjustment term becomes zero (since $\Phi(\underline{q}_k) \in \langle \Psi^{(0)} \rangle$), hence $r_j^{(l+1)}(\underline{q}_k) = r_j^{(l)}(\underline{q}_k) = r_j^{(l-1)}(\underline{q}_k), \cdots, = r_j^{(1)}(\underline{q}_k) = d_j(\underline{q}_k)$. This implies that the scoring function for this query remains unchanged (i.e., locally stable) as the first time this query was incorporated with the desired score of $d_j(\underline{q}_k)$ assigned by the user.

B. Selective Sampling Using Fisher Information Matrix

The idea behind selective sampling, statistical active learning or experimental design [23], [24] 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 this search process. In our CBIR, selective sampling also controls the expansion of the neuron pools in Fig. 2 by choosing only those most informative query samples during the relevance feedback learning. The selected query forms a new unit in the pool with a kernel activation defined by that query. As shown in Appendix II, this new query brings old and new information to the pool. While the *old information* (can be deduced from the other already incorporated queries) is not playing a role in relevance feedback learning, the *new information* controls the weight adjustment and scoring function adaptation in the linear kernel space. In this section, we study this issue in the context of Fisher information matrix for sequential selection of training samples, or best basis functions, for updating the scoring function in (11).

For any new query $\underline{q}_i$ in set $\mathcal{Q}$,[1] the generated score, $r_j(\underline{q}_i)$, using the kernel-based scoring function (7) and the desired score, $d_j(\underline{q}_i)$, for this query are related via

$$d_j(\underline{q}_i) = r_j(\underline{q}_i) + \epsilon_i = \underline{b}_j^T \mathcal{K}(\underline{q}_i) + \epsilon_i \tag{15}$$

where $\underline{b}_j$ is the weight vector, $\mathcal{K}(\underline{q}_i)$ is the vector of basis functions (kernels) $k(\underline{q}_i, \underline{q}_l)$'s, and 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 learned L queries) is to be used to minimize the generalization error on this set $\mathcal{Q}$, the best unbiased linear estimator (BLUE) [25] of $\underline{b}_j$ given L observations, $\underline{q}_i \in \mathcal{Q}$, can be expressed as

$$\hat{\underline{b}}_j = M_0^{-1} \underline{y}_j \tag{16}$$

where $M_0 = \sum_{i=1}^{L} (1/\sigma_i^2) \mathcal{K}(\underline{q}_i) \mathcal{K}^T(\underline{q}_i)$ is the Fisher information matrix and $\underline{y}_j = \sum_{i=1}^{L} d_j(\underline{q}_i) \mathcal{K}(\underline{q}_i)/\sigma_i^2$. 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 $M_0 = (1/\sigma^2) \sum_{i=1}^{L} \mathcal{K}(\underline{q}_i) \mathcal{K}^T(\underline{q}_i) = (1/\sigma^2) G_0^T G_0 = (1/\sigma^2) G_0^2$, where $G_0 = \Psi^{(0)^T} \Psi^{(0)}$ is the symmetric Gram matrix or the *sensitivity matrix* [26].

It can easily be shown that error covariance matrix for the unbiased parameter vector estimate is $C_j = E[(\hat{\underline{b}}_j - E[\hat{\underline{b}}_j])(\hat{\underline{b}}_j - E[\hat{\underline{b}}_j])^T] = M_0^{-1}$, i.e., the uncertainty in the parameter estimates on set $\mathcal{Q}$ is proportional to the inverse of Fisher information matrix (Cramer-Rao Bound). Furthermore, the determinant and trace (error variance) of C_j are the smallest for BLUE among all estimators of $\underline{b}_j$ [25].

Now, having learned L queries in pool j, we would like to devise a sequential procedure for choosing the next query $\underline{q}_{L+1} \in \mathcal{Q}$ (or basis function), which is most informative to the pool for updating the weights. The most informative query introduces least amount of parameter uncertainties in the scoring function,

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

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 $p_L = logM_0$, then using $M_0 = (1/\sigma^2)G_0^2$, we have

$$p_L = logM_0 = 2L\log|\sigma| + 2\log det(G_0). \tag{17}$$

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

$$p_{L+1} = \log M_1 = 2(L+1)\log|\sigma| + 2\log det(G_1) \tag{18}$$

where M_1 is the new information matrix and G_1 is defined in Appendix I. The change in the information measure as a result of introducing $\underline{q}_{L+1}$ is given by

$$g_{L+1} = \log det(G_1) - \log det(G_0). \tag{19}$$

Using the definition of G_1 in (A2), it is obvious that $det(G_1) = \zeta det(G_0)$ where $\zeta = \beta - \underline{v}^T G_0^{-1}\underline{v}$. Thus, using the results in the Appendices, information change g_{L+1} after introducing query $\underline{q}_{L+1}$ can be simplified to

$$g_{L+1} = \log|\zeta| = \log\left\| P^{\perp}_{\langle\Psi^{(0)}\rangle}\Phi(\underline{q}_{L+1})\right\|^2. \tag{20}$$

As can be seen, the log of the denominator of adjustment terms in the scoring function updating equation (11) and the weight updating equation (A5) is the change in information g_{L+1}. Also, in Appendix II, it was shown that the *new information* in the query $\underline{q}_{L+1}$ is given by $P^{\perp}_{\langle\Psi^{(0)}\rangle}\Phi(\underline{q}_{L+1})$.

Furthermore, using (A4) and cyclic property of the trace, the change in the error variance (trace of C_j) as a result of introducing the new query image is given by

$$\begin{aligned}\nu_{L+1} &= tr\left(M_1^{-1}\right) - tr\left(M_0^{-1}\right) = \sigma^2\left[tr\left(G_1^{-2}\right) - tr\left(G_0^{-2}\right)\right]\\ &= \frac{\sigma^2}{\zeta^2}\left[2\zeta\underline{z}^T G_0^{-1}\underline{z} + (\underline{z}^T\underline{z}+1)^2\right].\end{aligned} \tag{21}$$

Now, using the definitions of $\underline{z}$ and $\underline{v}$ in Appendix I, we can write the terms in (21) as

$$\begin{aligned}\sigma^2\underline{z}^T\underline{z} &= \underline{v}^T M_0^{-1}\underline{v} = tr\left(M_0^{-1}\underline{v}\underline{v}^T\right)\\ &= \beta\ tr\left[M_0^{-1}\ \Psi^{(0)^T}P_{\left\langle\Phi(\underline{q}_{L+1})\right\rangle}\Psi^{(0)}\right]\end{aligned}$$

and

$$\begin{aligned}\sigma^2\underline{z}^T G_0^{-1}\underline{z} &= tr\left(G_0^{-1}\ M_0^{-1}\underline{v}\underline{v}^T\right)\\ &= \beta\ tr\left[G_0^{-1}\ M_0^{-1}\ \Psi^{(0)^T}P_{\left\langle\Phi(\underline{q}_{L+1})\right\rangle}\Psi^{(0)}\right]\end{aligned} \tag{22}$$

where $P_{\langle\Phi(\underline{q}_{L+1})\rangle} = \Phi(\underline{q}_{L+1})(\Phi^T(\underline{q}_{L+1})\Phi(\underline{q}_{L+1}))^{-1}\Phi^T(\underline{q}_{L+1}) = \Phi(\underline{q}_{L+1})\Phi^T(\underline{q}_{L+1})/\beta$ is the projection matrix associated with 1-D subspace spanned by new query image $\underline{q}_{L+1}$. Since all the terms in the numerator of (21) are positive, the change in the error variance, ν_{L+1}, takes its minimum when the numerator is the smallest, i.e., $P_{\langle\Phi(\underline{q}_{L+1})\rangle}\Psi^{(0)} = \mathbf{0}$, and the denominator ζ, is the largest.

Thus, when the new query image carries little new information, i.e., somewhat redundant query (small g_{L+1} or ζ) more uncertainties (large error variance) are induced to the weight vector of the 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. To avoid these problems, the sequential selective sampling algorithm should involve the following steps: i) select from the training set the unmarked query sample that carries the most new information (largest g_{L+1}, hence smallest ν_{L+1}), ii) submit this most informative query and apply relevance feedback, if needed; iii) mark this query image; and iv) repeat the process starting from step i) until there is no more unmarked query images.

V. Experimental Results

A. Database Description

To assess the performance of the proposed adaptable CBIR several experiments are conducted on a domain-specific image database. This database was collected using an electro-optical (EO) scanning sensor that produces high-resolution 3-D images of underwater objects. The sensor scanned a target field stretched approximately 320 feet long in shallow water (<60 feet deep) in the Gulf of Mexico, Panama City, FL. Several runs were conducted and the raw data was then rendered to produce pairs of contrast (gray-level) and range (depth) maps [27]. The original data set consisted of 117 pairs of contrast and range images. There were 12 mine-like and 28 different non-mine-like objects [28]. The mine-like objects included several bullet, cylindrical, truncated cone, and trapezoidal shaped targets. The non-mine-like objects included some typical man-made and natural objects that could be found on the sea floor. Examples of man-made objects are: tire, crab/fish trap, concrete pipe, pole, and a 55-gallon drum. Additionally, there are several panels (both flat and raised) that are primarily designed and used to quantify certain physical phenomena in the EO imagery and validate the models [28]. Although the database is rich in terms of different object types, the number of samples is limited as far as *within-class* diversity is concerned. Thus, we synthetically generated new images using the geometric and perspective transformation of the contrast and range images to mimic scenarios where the objects are viewed at different grazing angles and elevations of the sensor platform. The expanded data set contained a total of 585 pairs of images where for each object, typically 4–12 pair of images exist. Although the size of this data set is rather small, the within-class diversity in terms of object rotation, grazing angle (14, 27, 37, and 45°) and scaling makes the relevance feedback goals very challenging to achieve, particularly if exact relative position and score of relevant images must be met.

The feature set consisted of 36 shape and texture attributes extracted from every pair of contrast and range maps. To extract shape dependent features from each segmented object the method of Zernike moments [29] was used. Comparing to the invariant moments, Zernike moments are less sensitive to additive noise and distortion effects in the images. For each object,

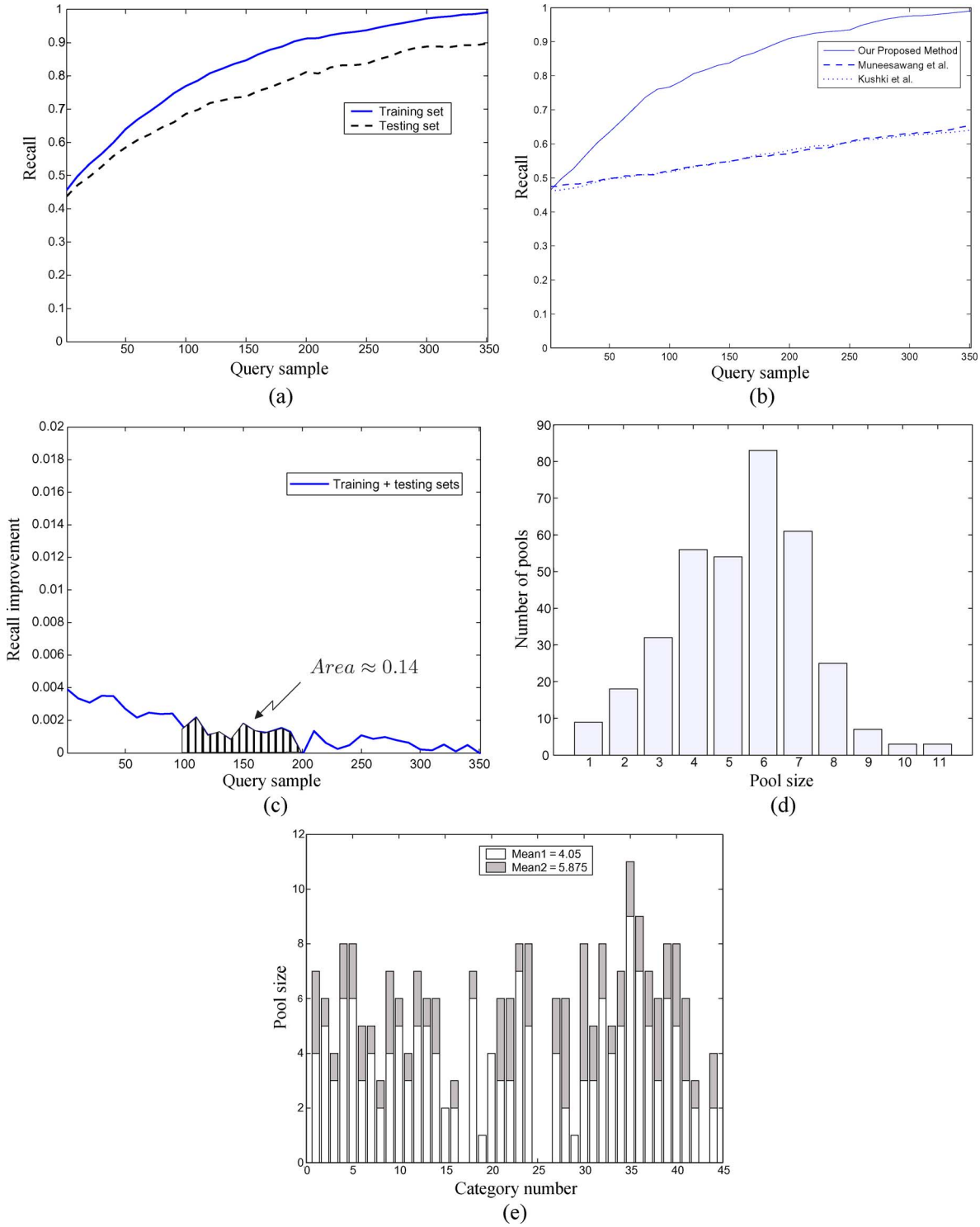


Fig. 3. Experiment 1 results. (a) Recall measure versus the training query sample; (b) recall on the entire database and comparison with other methods; (c) improvement in the recall measure versus the query sample; (d) histogram of pool size; (e) minimum and maximum pool sizes for different object categories.

18 Zernike moments of order 2 to 7 were extracted from the combined (union) contrast and range silhouettes [30]. Our experimental results [30] indicated that though Zernike moments are invariant to rotation, scaling and translation, they do indeed change wrt to grazing angle. Thus, to complement these features with another set that remains invariant to grazing angle, gray level co-occurrence matrices (GLCM) method [31] was applied to extract textural features from the segmented objects in both range and contrast maps. This method computes five statistical features namely contrast, correlation, entropy, energy, and homogeneity for each image. Additionally, we used mean and variance along x and y coordinates for each image of the segmented object.

The training and testing data sets consisted of 351 and 234 randomly chosen feature vectors from different object categories that collectively form the image query database. A file

containing query images and their corresponding scores when applying relevance feedback learning was also generated. In all our experiments except Experiment 3, we used a Gaussian kernel with parameter $a = 1$. The reason for this choice is due to the fact that every component of the 36-dimensional feature vector was normalized based upon its standard deviation computed for the entire database.

B. Relevance Feedback Learning

In this section, the relevance feedback learning methodology developed in Section IV will be tested in conjunction with the selective sampling method described in Section IV-B. The primary objective of the relevance feedback for this specific application is to exploit expert operator's feedback to provide *in-situ* learning necessary for accurate object identification and association in changing operational and environmental conditions. Several experiments are conducted to thoroughly study the properties of the proposed single-round relevance feedback learning.

Experiment 1: The purpose of this experiment is to asses the performance of relevance feedback learning and the complexity of the adaptable CBIR in terms of the size of the pools. The network was first initially setup using the procedure in Section II-B. The relevance feedback was then applied together with the selective sampling method. The desired scores of these images are decided automatically using a simple scheme. The process entails finding, among the retrieved images, the relevant image that has the lowest score, r_1, among all the relevant images that are not misplaced, i.e., there is no nonrelevant image above them. If the score of the nonrelevant image, which is immediately below it, is r_0, the desired scores of those misplaced relevant images are arbitrarily chosen to be between r_0 and r_1. Once the desired scores are found the scoring function and weights of the pools corresponding to the misplaced relevant images are updated using (11) and (A5), respectively. This query is then marked and the process continues until no unmarked query is left.

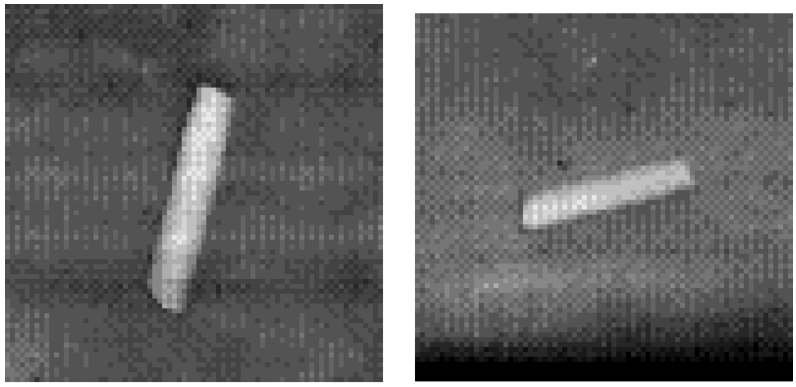
Fig. 3(a) shows the recall plots of the proposed single-round relevance feedback for the training and testing data sets. In our system, every time the learning on a single query, $\underline{q}_k$, from the training set is completed, the updated system is evaluated on the entire training and testing sets and percent correct recalls are computed. A perfect recall of "1" for a particular query image implies that all the top retrieved images after incorporating relevance feedback are most relevant (e.g., same class or within-class) to the query image. As can be seen, after starting from a rather low value around 0.45 corresponding to the initially trained system, the recall for both training and testing sets steadily increased reaching almost "1" and 0.9, respectively. The high final recall value on the unseen testing set, after the relevance feedback learning on all the training set is completed, is indicative of excellent generalization of the proposed system.

Fig. 3(b) shows the average recall measure on the entire query database consisting of 351 training and 234 testing queries, respectively, after the relevance feedback is applied to every query in the training set. This plot is the result of combining the recall measures of the training and testing sets shown in Fig. 3(a). Fig. 3(c) shows the improvement in the recall measure as the training queries are submitted. This plot is obtained by finding the difference between the average recall (averaged over the entire set of queries) before and after relevance feedback on a particular training query is completed. While each point on this curve shows the incremental improvement in the recall measure for one particular query, the area under this curve between two query samples represents the aggregated improvement in the average recall between the two points. As can be observed from Fig. 3(b), the average recall approaches a very high final value close to 1. It is also interesting to note that the improvement in average recall reduces as training progresses. This is due to the selective sampling method for picking the best query samples or best basis functions for scaling function representation that guarantees generalization ability of the adaptive CBIR system.

Next, we study the complexity of our CBIR in terms of the size of the expanded pools as a consequence of relevance feedback. Fig. 3(d) shows the plot of the number of pools versus the pool size. As can be seen, the minimum pool size is 1 while the maximum size is 11. There are 9 pools of size 1 that never received relevance feedback, implying that their predicted scores for the training queries were close to the desired values, and, hence, the pools were not expanded. In contrary, the pools of size 11 (3) were the least reliable ones as they did not predict the desired scores almost every time that a query image of the same category was submitted. As a result, they received multiple relevance feedback for different queries. The average pool size is found to be 5, i.e., on overage pools receive relevance feedback four times during the training process.

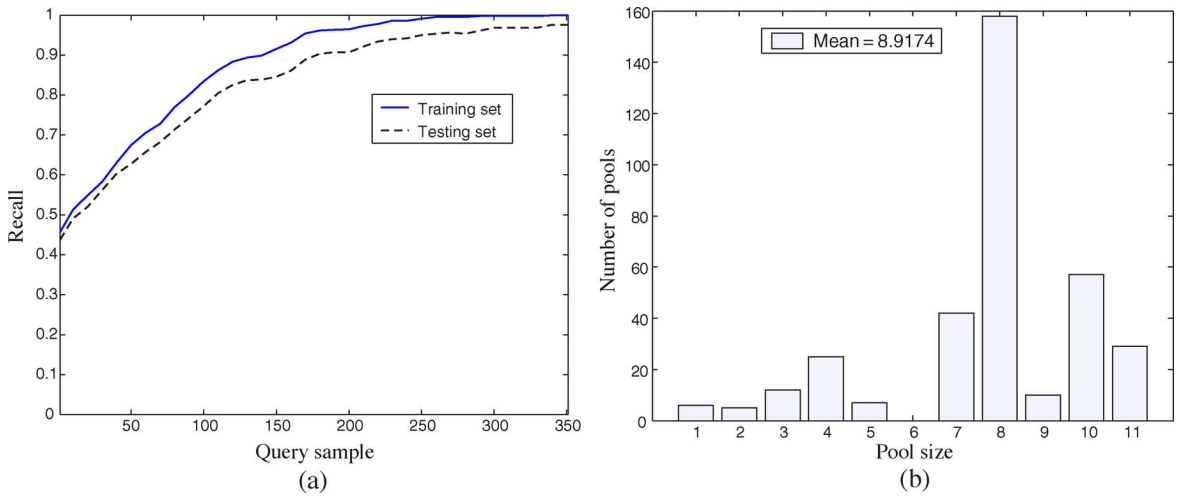
It must be pointed out that the size of the expanded pools directly relates to the similarity of the images in different categories. The pools associated with the objects that are similar in shape, size and texture but belong to different categories receive more relevance feedback than those without feature similarity. Fig. 3(e) shows the bar chart of minimum and maximum pool sizes versus the object category index. Mine-like objects comprise categories 28–39 while the rest are non-mine-like objects. As can be seen, the pools corresponding to Categories 19 and 29 are very reliable as they do not receive any relevance feedback. Clearly, the extracted features from these objects were capable of successfully representing and distinguishing these objects from the rest. In contrast, categories 36 and 37 have received large number of relevance feedback. These two particular categories correspond to cylindrical mine-like objects having slight differences at their tapered ends, which are hard to distinguish even for expert operators. Examples of two images in these categories are shown in Fig. 4(a) and (b). Clearly, this explains the reason why the final (after training) pool sizes for Categories 36 and 37 are 9 and 7, respectively.

Experiment 2: The purpose of this experiment is to show that using a more stringent score assignment the recall can further be improved at the price of increasing the pool sizes. Relevance feedback learning is applied whenever the score of a relevant image is below a prespecified value (e.g., 0.9). The difference between this preselected desired score and the actual score is used to update the appropriate pool by using (11) and (A5). All other steps are the same as those in Experiment 1.



(a) (b)

Fig. 4. Two object categories with substantial feature similarity. (a) Category 36; (b) category 37.



(a) (b)

Fig. 5. Experiment 2 results. (a) Recall measure versus the training query sample; (b) histogram of pool size.

Fig. 5(a) shows the recall plot during the relevance feedback learning. Comparing to the plot in Fig. 3(a) one can notice three major differences. First, the recall for the testing test reaches a higher final value of approximately 0.98 when the learning on the training set is completed. Second, the recall plot on the testing set continues increasing even after the recall on the training set has reached its maximum value of "1" around query image 250. The reason being unlike in Experiment 1, here relevance feedback is applied based upon the score of the relevant images rather than the relative positions with respect to the nonrelevant images. Moreover, the recall at the initial stages of learning shows a higher trend than that of Experiment 1, implying that the prediction capabilities of the pools of neurons are much better in this case. This improved performance comes at the price of increasing the size of the pools. Fig. 5(b) shows the plot of the number of pools versus their size after the relevance feedback training is completed. Comparing to Fig. 3(b), it can easily be observed that the average size of the pools is increased to almost 9 neurons per pool, which is the average number of objects in each category. Clearly, there are still pools with only few neurons, e.g., 6 pools of size 1, 5 pools of size 2, and 11 pools of size 3.

Experiment 3: In this experiment, effects of parameter a of the Gaussian kernel function is studied on the stability and generalization ability of our adaptive CBIR. The purpose is to highlight the importance of choosing an appropriate value for this parameter for a given application. Experiment 2 $(a = 1)$ was repeated for much larger values of this parameter, namely $a =$ 400, 500, 1000, and 5000. Fig. 6(a) and (b) shows the recall plots for $a = 500$ and 5000 cases. The results indicate that, in all cases, the recall on the training set increased almost linearly reaching perfect recall value of "1" with probability 1 at the end of the training. This, experimentally validates the strict interpolation result in Remark 1 and proves the stability of our CBIR even for very large values of parameter a, e.g., 5000. Additionally, the initial recall at the start up for either the training or testing set is deteriorated as parameter a increases. For $a = 5000$, the initial recall for the testing set is around 0.034 and the recall measure remains almost at that level throughout the entire relevance feedback training, hence confirming the statement made in Remark 1 about the low prediction capability of the system for large values of a.

C. Comparison With Other Relevance Feedback Learning Algorithms

Method 1: Here, we compare the relevance feedback learning algorithm proposed in this paper to the multiround query feedback learning algorithm in [2]. In this algorithm, upon submitting a query $\underline{q}^{(t)}$ at time t, the initial similarity score vector $\underline{s}(\underline{q}^{(t)})$ is computed leading to the ranked list of images. During

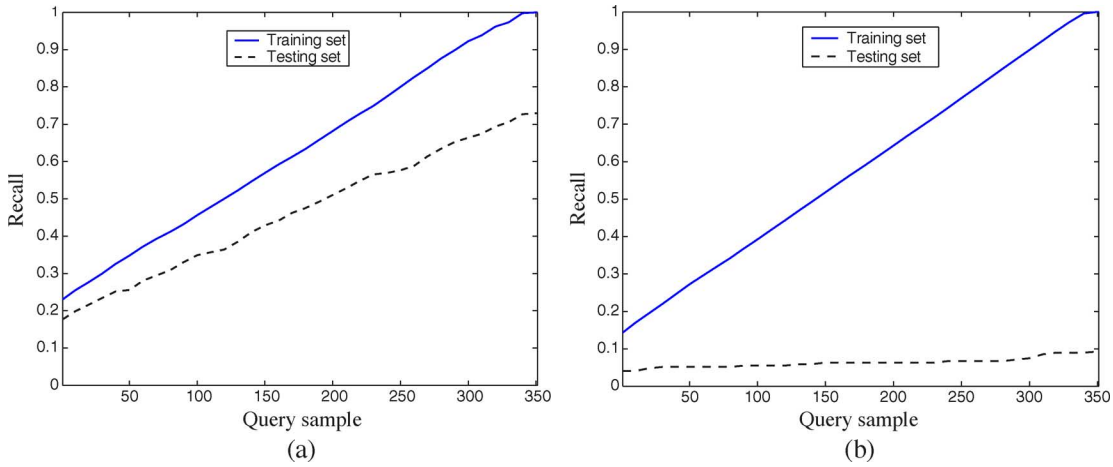


Fig. 6. Recall measure for large values of kernel's parameter a. (a) $a = 500$; (b) $a = 5000$.

the relevance feedback, new score vector $\underline{\mathrm{s}}'(\underline{\mathrm{q}}^{(t)})$ is computed from $\underline{\mathrm{s}}(\underline{\mathrm{q}}^{(t)})$ by forcing the scores of the nonrelevant images to be zero and keeping the scores of the relevant images the same. Also, the scores of the unlabeled (or undecided) images are multiplied by a forgetting factor $\alpha \in (0,1)$ to ensure convergence of the learning algorithm. During each round of feedback, involving query selection and score update, the relevant image that is listed last in the retrieved list for the original query image is chosen as the new query image $\underline{\mathrm{q}}_{\mathrm{new}}^{(t)}$. This ensures that maximum information is captured during the relevance feedback score update. Using $\underline{\mathrm{s}}(\underline{\mathrm{q}}_{\mathrm{new}}^{(t)})$ obtained for the new query image $\underline{\mathrm{q}}_{\mathrm{new}}^{(t)}$ and $\underline{\mathrm{s}}'(\underline{\mathrm{q}}^{(t)})$, the ranked list at the next iteration $t+1$ is computed using

$$\underline{\mathrm{s}}\left(\underline{\mathrm{q}}^{(t+1)}\right) = \min\left(\underline{\mathrm{s}}'\left(\underline{\mathrm{q}}^{(t)}\right), \underline{\mathrm{s}}\left(\underline{\mathrm{q}}_{\mathrm{new}}^{(t)}\right)\right)^{\beta} \times \max\left(\underline{\mathrm{s}}'\left(\underline{\mathrm{q}}^{(t)}\right), \underline{\mathrm{s}}\left(\underline{\mathrm{q}}_{\mathrm{new}}^{(t)}\right)\right)^{(1-\beta)} \quad (23)$$

where the operations are done element-wise. In our experiment, the parameters α and β used in this similarity score update process are chosen to be 0.25 and 0.5, respectively, as suggested in [2]. The performance of the algorithm is evaluated by training the system over the 351 query images in the training data set. For consistency purpose, the maximum number of rounds of feedback applied for each query image is set to be equal to the number of relevant images for that query image. Fig. 3(b) shows the plot of the average recall computed over the 351 and 234 query images contained in the training and testing data sets, respectively. Each point in this plot is obtained by averaging the recall over all the query images in the training and testing data sets combined, after applying the relevance feedback for the given query image in the training set. The average recall before any user feedback is 0.46 while the average recall after applying relevance feedback to all the query images in the training data becomes 0.64.

Method 2: In [14], a relevance feedback learning algorithm is developed that updates an RBF-based similarity function. The scoring function is the sum of several univariate Gaussian centered at each query image feature vector element. The standard deviation of each Gaussian is heuristically updated based upon the selected relevant images. The query model is also progressively updated based on the relevance feedback from the users. When a query image $\underline{\mathrm{q}}^{(t)}$ is submitted at time t, the similarity scores are computed using

$$s\left(\underline{\mathrm{q}}^{(t)}, \underline{\mathrm{x}}_i\right) = \sum_{j=1}^{L} e^{-\left(x_{ij} - q_j^{(t)}\right)^2 / 2\sigma_j^2} \quad (24)$$

where x_{ij} and $q_j^{(t)}$ are the j^{th} elements of $\underline{\mathrm{x}}_i$ and $\underline{\mathrm{q}}^{(t)}$, respectively, σ_j is the RBF width of the j^{th} feature component. The user evaluates the retrieved list of images by selecting the images that are relevant and nonrelevant to the submitted query image. The query vector at time $t+1$ is then updated using a modified learning vector quantization (LVQ) learning [32] as

$$\underline{\mathrm{q}}^{(t+1)} = \underline{\mathrm{q}}^{(t)} + \alpha_r\left(\underline{\mathrm{x}}'^{(r)} - \underline{\mathrm{q}}^{(t)}\right) - \alpha_n\left(\underline{\mathrm{x}}'^{(n)} - \underline{\mathrm{q}}^{(t)}\right) \quad (25)$$

where $\underline{\mathrm{x}}'^{(r)}$ and $\underline{\mathrm{x}}'^{(n)}$ are the centroids of the relevant and nonrelevant image feature vectors at iteration t and α_r and α_n are chosen [14] to be 1.4 and 0.4, respectively. The RBF width is updated using $\sigma_j = e^{\beta s_j}$, where s_j is the standard deviation of the j^{th} feature component of the selected relevant images and β is a constant chosen [14] to be 2.6.

Fig. 3(b) also shows the plot of the average recall versus the query images in the training data for this method. The average recall on the entire data set at the startup and after relevance feedback learning on all 351 queries in the training data is completed are 0.47 and 0.65, respectively. Although both methods exhibit an improvement of $\sim$0.2 in average recall after applying several rounds of relevance feedback for every query, the performance is still inferior to that of the proposed CBIR system which yields an almost perfect recall [see Fig. 3(b)] on the entire data set using only a single round of relevance feedback. Moreover, in contrast to the proposed system, in these methods the user concepts are not saved for future use after querying rounds are completed.

D. Model Reference Learning

In this section we conduct an experiment to evaluate the effectiveness of the model-reference learning described in Section III. As mentioned before, model reference learning

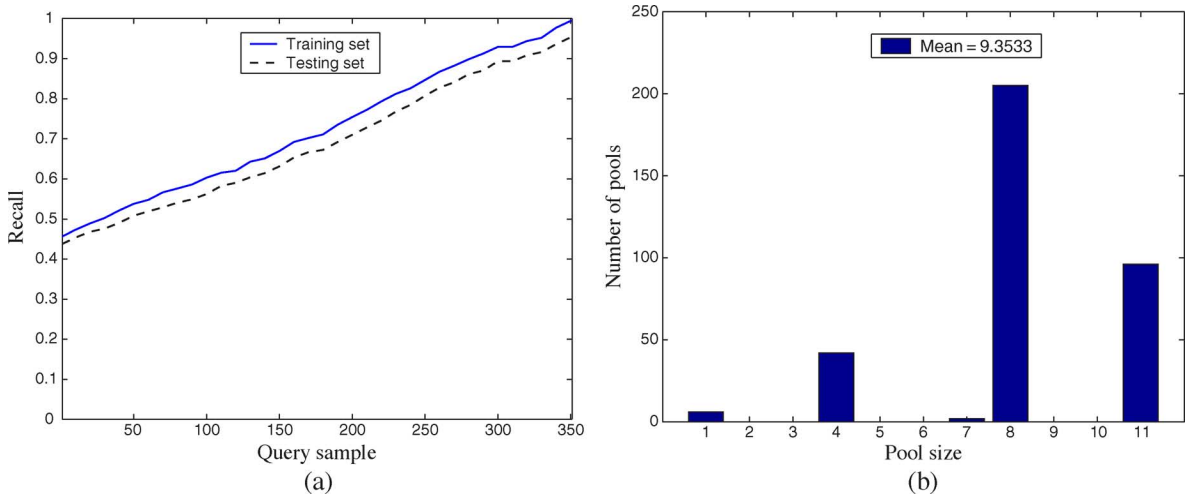


Fig. 7. Experiment 4 results. (a) Recall measure versus the training query sample; (b) histogram of pool size.

can be used prior to relevance feedback learning to capture prior known class-dependent information in the database. The training and testing sets are the same as those in previous experiments. However, unlike the relevance feedback learning model-reference learning is implemented in batch mode after all the input-output relationships of the training queries are captured in the model-reference database. Clearly, the selective sampling process cannot be applied here owing to the batch nature of the learning.

Experiment 4: Here, the class labels of the training queries are assumed to be known and used to estimate the desired score. Every pool is trained independently by first collecting all the input-output relationships for a set of training queries. To this aim, all the training queries are submitted to the CBIR operating in a search and retrieve mode. When a query image is submitted, if the score of a relevant image is less than the threshold 0.9, then its new score is set to 0.9; otherwise the score is kept unchanged. These input-output relationships are saved in a model-reference database for all the training queries, which is subsequently used to compute the weights of the pools using (7).

The recall plot for the model-reference learning mode is shown in Fig. 7(a). As can be observed, the recall measure for the training set reaches a perfect "1" while the recall for the testing set reaches a high value of 0.95. This final value is slightly less than that of Experiment 2 mainly owing to the selective sampling method used during the relevance feedback. Additionally, the rate of learning is somewhat fixed throughout the entire training. The histogram of size of the pools, after the model-reference learning is completed, is shown in Fig. 7(b). As can be seen, the average size is higher than that of the Experiment 2.

VI. Conclusion

This paper introduced an adaptable CBIR system that attempts to capture high-level semantic user concepts by modifying a set of similarity functions in the kernel domain. The learning can be implemented in two modes namely model-reference and relevance feedback. The former is used when the system is implemented as an adjustable multiclass classification system while the latter is applied to capture expert user concepts. The incorporation of user concepts in the relevance feedback learning is carried out in an online fashion while the incorporation of the model-reference information is performed in a batch mode. A simple kernel machine for the implementation of these two learning mechanisms was also proposed. This structurally dynamic two-layer network consists of several pools of neurons in the first layer where initially each pool contains only one neuron. As learning progress pools that receive feedback expand in order to perform similarity function adaptation for meeting the user requirements. The geometric interpretation and stability of the proposed relevance feedback learning were also carefully studied. To control the expansion of the pools during the relevance feedback learning and provide better generalization on the testing data an information-theoretic method was developed. Using this method, the most informative query image is selected and introduced into the pools.

To assess the usefulness of the proposed adaptive CBIR several experiments were conducted. These experiments reveal that a) the proposed single-round relevance feedback learning when used in conjunction with the selective sampling method provided excellent generalization on the new testing data set while controlling the expansion of the pools by picking the best basis for the scoring function representation; b) for the Gaussian kernel large values of the smoothing parameter, a, destroy the prediction and generalization ability of the system; and c) the model-reference learning assumes the knowledge of the class membership and provides average recall around 0.95 though the results are still inferior to those of the relevance feedback with a stringent scoring requirement. The proposed relevance feedback learning was then benchmarked against the methods in [2] and [14]. These results indicate that the average recall measures of these multiple-round relevance feedback learning methods are significantly lower than that of the proposed method. This is mainly attributed to the fact that these methods are not designed to meet exact scoring requirements and capture the subtle within-class information

in the database and/or the hidden user concepts. Moreover, the relevance feedback information for each query and user is not saved for future use by other users.

APPENDIX A
DERIVATIONS OF SCORING FUNCTION AND WEIGHT VECTOR UPDATING EQUATIONS

In this Appendix, we show how (10) can be simplified to yield the kernel-based recursive updating equation (11) for the scoring function $r_j^{(1)}(\underline{q})$. Let us start by simplifying the error vector $[\underline{d}_j^{(1)} - c_j \Psi^{(1)^T} \underline{w}_j^{(0)}]$ in (10) by using $\underline{d}_j^{(1)} = [\underline{d}_j^{(0)^T} \; d_j^{(L+1)}]^T$, where $d_j^{(L+1)}$ is the desired score of image j for the newest query image $\underline{q}_{L+1}$ and expressing $c_j \Psi^{(1)^T} \underline{w}_j^{(0)} = [c_j (\Psi^{(0)^T} \underline{w}_j^{(0)})^T \; c_j \Phi^T(\underline{q}_{L+1}) \underline{w}_j^{(0)}]^T$ where $\underline{d}_j^{(0)} = c_j \Psi^{(0)^T} \underline{w}_j^{(0)}$. Consequently, the error term in (10) can be rewritten as

$$\begin{aligned}\left[\underline{d}_j^{(1)} - c_j \Psi^{(1)^T} \underline{w}_j^{(0)}\right] &= \left[\underline{0}^T \left(d_j^{(L+1)} - c_j \Phi^T(\underline{q}_{L+1}) \underline{w}_j^{(0)}\right)\right]^T \\ &= [\underline{0}^T \delta_j]^T \end{aligned} \tag{A1}$$

where $\delta_j := d_j^{(L+1)} - r_j^{(0)}(\underline{q}_{L+1})$ is the prediction error. Next, the inverse of the new Gram matrix $G_1 = \Psi^{(1)^T} \Psi^{(1)}$ (after adding query image $\underline{q}_{L+1}$) can also be represented in terms of inverse of the old Gram matrix $G_0 = \Psi^{(0)^T} \Psi^{(0)}$ (before introducing query image $\underline{q}_{L+1}$) as follows:

$$\begin{aligned} G_1^{-1} &= \left(\Psi^{(1)^T} \Psi^{(1)}\right)^{-1} \\ &= \begin{bmatrix} \Psi^{(0)^T} \Psi^{(0)} & \Psi^{(0)^T} \Phi(\underline{q}_{L+1}) \\ \Phi^T(\underline{q}_{L+1}) \Psi^{(0)} & \Phi^T(\underline{q}_{L+1}) \Phi(\underline{q}_{L+1}) \end{bmatrix}^{-1} \\ &= \begin{bmatrix} G_0 & \underline{v} \\ \underline{v}^T & \beta \end{bmatrix}^{-1} \end{aligned} \tag{A2}$$

where vector $\underline{v}$ is defined as

$$\begin{aligned} \underline{v} &= \Psi^{(0)^T} \Phi(\underline{q}_{L+1}) \\ &= \left[k(\underline{q}_1, \underline{q}_{L+1}) k(\underline{q}_2, \underline{q}_{L+1}) \ldots k(\underline{q}_L, \underline{q}_{L+1})\right]^T \\ &= \mathcal{K}^{(0)}(\underline{q}_{L+1}) \end{aligned} \tag{A3}$$

and scalar $\beta = \Phi^T(\underline{q}_{L+1}) \Phi(\underline{q}_{L+1}) = k(\underline{q}_{L+1}, \underline{q}_{L+1})$. Now, the inverse of the symmetric matrix in (A2) can be expressed as

$$\begin{bmatrix} G_0 & \underline{v} \\ \underline{v}^T & \beta \end{bmatrix}^{-1} = \frac{1}{\zeta} \begin{bmatrix} G_0^{-1} \left(\zeta I + \underline{v} \underline{v}^T G_0^{-1}\right) & -G_0^{-1} \underline{v} \\ -\underline{v}^T G_0^{-1} & 1 \end{bmatrix} \tag{A4}$$

where $\zeta := \beta - \underline{v}^T G_0^{-1} \underline{v}$. Now, plugging (A4) and (A1) into (10) and making the necessary simplifications yields updating equation (11).

To find a recursive updating equation for weight vector $\underline{b}_j$, we substitute (A4) in the expression for $\underline{b}_j^{(1)} = (\Psi^{(1)^T} \Psi^{(1)})^{-1} \underline{d}_j^{(1)}$. This gives

$$\underline{b}_j^{(1)} = \begin{bmatrix} \underline{b}_j^{(0)} - \frac{1}{\zeta} \delta_j \underline{z} \\ \frac{1}{\zeta} \delta_j \end{bmatrix}. \tag{A5}$$

Note that $\zeta = k(\underline{q}_{L+1}, \underline{q}_{L+1}) - \sum_{i=1}^{L} k(\underline{q}_{L+1}, \underline{q}_i) z_i$. Also, it can easily be verified that z_i's, i.e., the element of $\underline{z}$ can be found by using (12) and solving for $\underline{z}$'s when $\underline{q}$ is set to $\underline{q}_1, \ldots \underline{q}_L$, which yields $\underline{z} = G_0^{-1} \underline{v}$.

As learning progresses, the inverse of the new Gram matrix can be computed using (A4) without actually performing any matrix inversion. Thus, (11) and (A5) can be carried out in the kernel domain without any computationally demanding matrix operations. The computational complexity of the scoring function updating equation (11) and weight updating equation (A5)) is only $O(L^2)$ primarily due to the computation of $\underline{z}$.

APPENDIX B
GEOMETRIC INTERPRETATION OF RELEVANCE FEEDBACK LEARNING

In this Appendix, we show the geometric interpretation of the relevance feedback learning in (11) in terms of the orthogonal projection matrix, $P_{\langle \Psi^{(0)} \rangle} = \Psi^{(0)} (\Psi^{(0)^T} \Psi^{(0)})^{-1} \Psi^{(0)^T}$ corresponding to linear subspace $\langle \Psi^{(0)} \rangle$. To this end, let us first consider $\hat{\Phi}(\underline{q}_{L+1}) = P_{\langle \Psi^{(0)} \rangle} \Phi(\underline{q}_{L+1})$, which is the projection (or estimate) of the new query $\underline{q}_{L+1}$ onto the subspace spanned by columns of $\Psi^{(0)}$, i.e., the *old information* in this new query. Using the definition of $\underline{z}$ in Appendix I, it can easily be verified that

$$\hat{\Phi}(\underline{q}_{L+1}) = P_{\langle \Psi^{(0)} \rangle} \Phi(\underline{q}_{L+1}) = \Psi^{(0)} G_0^{-1} \underline{v} = \sum_{i=1}^{L} \Phi(\underline{q}_i) z_i. \tag{B6}$$

The *new information* in the query $\underline{q}_{L+1}$, which cannot be estimated from the old (previously captured) queries, $\underline{q}_1, \ldots \underline{q}_L$, is contained in $\tilde{\Phi}(\underline{q}_{L+1}) := \Phi(\underline{q}_{L+1}) - \hat{\Phi}(\underline{q}_{L+1}) = P_{\langle \Psi^{(0)} \rangle}^{\perp} \Phi(\underline{q}_{L+1})$ where $P_{\langle \Psi^{(0)} \rangle}^{\perp} = I - P_{\langle \Psi^{(0)} \rangle}$ is the orthogonal complement of $P_{\langle \Psi^{(0)} \rangle}$. That is, this new information is the projection of $\Phi(\underline{q}_{L+1})$ onto the subspace orthogonal to $\langle \Psi^{(0)} \rangle$. Clearly, we have $\hat{\Phi}^T(\underline{q}_{L+1}) \tilde{\Phi}(\underline{q}_{L+1}) = 0$. Now, premultiplying both sides of (B6) by $\Phi^T(\underline{q}_{L+1})$ and using kernels yields

$$\Phi^T(\underline{q}_{L+1}) \hat{\Phi}(\underline{q}_{L+1}) = \left\| \hat{\Phi}(\underline{q}_{L+1}) \right\|^2 = \sum_{i=1}^{L} k(\underline{q}_{L+1}, \underline{q}_i) z_i. \tag{B7}$$

Similarly, we have

$$\begin{aligned} \Phi^T(\underline{q}_{L+1}) \tilde{\Phi}(\underline{q}_{L+1}) &= \left\| \tilde{\Phi}(\underline{q}_{L+1}) \right\|^2 \\ &= k(\underline{q}_{L+1}, \underline{q}_{L+1}) - \sum_{i=1}^{L} k(\underline{q}_{L+1}, \underline{q}_i) z_i \\ \text{and} \quad \Phi^T(\underline{q}) \tilde{\Phi}(\underline{q}_{L+1}) &= \tilde{\Phi}^T(\underline{q}) \tilde{\Phi}(\underline{q}_{L+1}) \\ &= k(\underline{q}, \underline{q}_{L+1}) - \sum_{i=1}^{L} k(\underline{q}, \underline{q}_i) z_i. \end{aligned} \tag{B8}$$

where $\tilde{\Phi}(\underline{q}) = P_{\langle \Psi^{(0)} \rangle}^{\perp} \Phi(\underline{q})$.

Thus, the denominator of the adjustment term in the scoring function (11) is the norm of $\tilde{\Phi}(\underline{q}_{L+1})$ while the numerator is dependent on the norm of $\tilde{\Phi}(\underline{q})$ and the cosine of the angle between this component and $\tilde{\Phi}(\underline{q}_{L+1})$. Note that if $\Phi(\underline{q}) \in \langle \Psi^{(0)} \rangle$, then the numerator becomes zero and no updating takes place as the

information of the query $\underline{q}$ in the mapped space is already captured by L previous stored queries $\underline{q}_k$, $k \in [1, L]$'s that form the subspace $\langle \Psi^{(0)} \rangle$.

ACKNOWLEDGMENT

The authors would like to thank NSWC-Panama City, FL, for providing the electro-optical imagery database for this study.

REFERENCES

[1] G. Guo-Dong, A. Jain, M. Wei-Ying, and Z. Hong-Jiang, "Learning similarity measure for natural image retrieval with relevance feedback," *IEEE Trans. Neural Netw.*, vol. 13, no. 7, pp. 811–820, Jul. 2002.
[2] A. Kushki, P. Androutsos, K. Plataniotis, and A. Venetsanopoulos, "Query feedback for interactive image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 644–655, May 2004.
[3] R. Zhang and Z. Zhang, "Effective image retrieval based on hidden concept discovery in image database," *IEEE Trans. Image Process.*, vol. 16, no. 2, pp. 562–572, Feb. 2007.
[4] G. Aggarwal, S. Ghosal, and P. Dubey, "Efficient query modification for image retrieval," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2000, vol. 2, pp. 255–262.
[5] K. Porkaew, M. Ortega, and S. Mehrota, "Query reformulation for content based multimedia retrieval in MARS," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, Jun. 1999, vol. 2, pp. 747–751.
[6] A. D. Doulamis and N. Doulamis, "Generalized nonlinear relevance feedback for interactive content-based retrieval and organization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 5, pp. 656–671, May 2004.
[7] B. Scholkopf and A. J. Somola, *Learning With Kernels*. Cambridge, MA: MIT, 2002.
[8] E. Bauer, "An empirical comparison of voting classification algorithms: Bagging, boosting and variants," *Mach. Learn.*, vol. 36, pp. 105–139, Aug. 1999.
[9] Y. Chen, X. S. Zhou, and T. Huang, "One-class SVM for learning in image retrieval," in *Proc. Int. Conf. Image Processing*, Thessaloniki, Greece, 2001, vol. 1, pp. 34–37.
[10] J. Li, N. Allinson, D. Tao, and X. Li, "Multitraining support vector machine for image retrieval," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3597–3601, Nov. 2004.
[11] D. Tao, X. Tang, X. Li, and X. Wu, "Asymmetric bagging and random subspace for support vector machnies-based relevance feedback in image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1088–1099, Jul. 2006.
[12] D. Tao, X. Li, and S. Maybank, "Negative samples analysis in relevance feedback," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 4, pp. 568–580, Apr. 2007.
[13] N. Vasconcelos and A. Lippman, "A Bayesian framework for content-based indexing and retrieval," in *Proc. Data Compression Conf.*, 1998, p. 580.
[14] 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. 703–716, Oct. 2004.
[15] D. Tao, X. Tang, and X. Li, "Which components are important for interactive image searching," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 1, pp. 3–11, Jan. 2008.
[16] D. Xu, S. Yan, D. Tao, S. Lin, and H. Zhang, "Marginal fisher analysis and its variants for human gait recognition and content-based image retrieval," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2811–2821, Nov. 2007.
[17] D. Tao, X. Tang, X. Li, and Y. Rui, "Direct Kernel biased discriminant analysis: A new content-based image retrieval relevance feedback algorithm," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 716–727, Apr. 2006.
[18] R. Manmatha, T. Rath, and F. Feng, "Modeling score distributions for combining the outputs of search engines," in *Proc. 24th Annu. Int. ACM SIGIR Conf. Research and Development in Information Retrieval*, 2001, pp. 267–275.
[19] A. D. Doulamis, N. Doulamis, and S. Kollias, "Interactive content-based retrieval in video databases using fuzzy classification and relevance feedback," in *Proc. IEEE Int. Conf. Multimedia Computing and Systems*, 1999, vol. 2, pp. 954–958.
[20] A. N. Tikhonov, A. V. Goncharsky, V. V. Stepanov, and A. G. Yagola, *Numerical Methods for the Solution of Ill-Posed Problems*. Norwell, MA: Kluwer, Aug. 1995, vol. 328.
[21] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, Sep. 1990.
[22] R. Courant and D. Hilbert, *Methods of Mathematical Physics*, 3 ed. New York: Interscience, 1961.
[23] K. Fukumizu, "Statistical active learning in multilayer perceptrons," *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 17–26, Jan. 2000.
[24] E. Dura, Y. Zhang, X. Liao, G. Dobeck, and L. Carin, "Active learning for detection of mine-like objects in side-scan sonar imagery," *IEEE J. Ocean. Eng.*, vol. 30, no. 2, pp. 360–371, Feb. 2005.
[25] V. Fedorov, *Theory of Optimal Experiments*. New York: Academic, 1972.
[26] L. Scharf and L. T. McWhorter, "Geometry of the Cramer-Rao bound," *Signal Process.*, vol. 31, no. 3, pp. 301–311, 1993.
[27] A. J. Nevis, "Image enhancement for mine identification with laser line scan sensors," presented at the 3rd Int. Symp. Technology and the Mine Problem, Monterey, CA, Apr. 1998.
[28] J. S. Taylor and M. C. Hulgan, "Electo-optic identification research program," in *Proc. MTS/IEEE Oceans Conf.*, Biloxi, MS, Oct. 2002, vol. 2, pp. 994–1002.
[29] S. Liao and M. Pawlak, "On the accuracy of Zernike moments for image analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1358–1364, Dec. 1998.
[30] J. Salazar, "Adaptive Text and Image Retrieval Systems Using Relevance Feedbck," Ph.D. dissertation, Colorado State Univ., Fort Collins, 2006.
[31] R. Haralick *et al.*, "Textural features for image classification," *IEEE Trans. Syst. Man Cybern.*, vol. 3, pp. 610–621, Mar. 1973.
[32] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

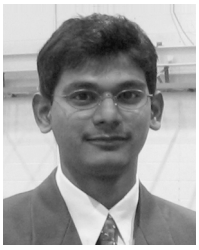
Mahmood R. Azimi-Sadjadi (M'81–SM'89) received the M.S. and Ph.D. degrees from the Imperial College of Science and Technology, University of London, London, U.K., in 1978 and 1982, respectively, both in electrical engineering with specialization in digital signal/image processing.

He is currently a Full Professor with the Electrical and Computer Engineering Department, Colorado State University (CSU), Fort Collins. He is also serving as the director of the Digital Signal/Image Laboratory at CSU. His main areas of interest include digital signal and image processing, target detection, classification and tracking using sonar, acoustics, radar and IR sensors, sensor array processing and distributed sensor networks. He is the co-author of the book *Digital Filtering in One and Two Dimensions* (Plenum, 1989).

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

Jaime Salazar received the B.S. degree in physics in 1981, the M.S. degree in computer science from the Instituto Teconológico y de Estudios Superiores de Monterrey (ITESM), Mexico, in 1991, and the Ph.D. degree in electrical and computer engineering from Colorado State University, Fort Collins, in 2006.

Currently, he is an Associate Professor at the Computer Science Department, ITESM, Toluca Campus, Mexico. His research areas of interest include: pattern recognition, text and image retrieval systems, and multimedia systems.



Saravanakumar Srinivasan received the B.E. degree in electronics and computer engineering from Coimbatore Institute of Technology, India, in 2000, and the M.S. in electrical engineering from Colorado State University, Fort Collins, in 2004.

From 2000 to 2002, he was with Larsen and Toubro Infotech as a Software Engineer. Since 2005, he has been with Information System Technologies, Inc., Fort Collins, where he is a Technical Research Staff involved in several distributed acoustic sensing projects. His research interests include: text and image retrieval systems, digital, detection, estimation and classification, and neural networks.