# An Interactive Approach for CBIR Using a Network of Radial Basis Functions

Paisarn Muneesawang, *Member, IEEE,* and Ling Guan, *Senior Member, IEEE*

***Abstract*—An important requirement for constructing effective content-based image retrieval (CBIR) systems is accurate characterization of visual information. Conventional nonadaptive models, which are usually adopted for this task in simple CBIR systems, do not adequately capture all aspects of the characteristics of the human visual system. An effective way of addressing this problem is to adopt a "human–computer" interactive approach, where the users directly teach the system about what they regard as being significant image features and their own notions of image similarity. We propose a machine learning approach for this task, which allows users to directly modify query characteristics by specifying their attributes in the form of training examples. Specifically, we apply a radial-basis function (RBF) network for implementing an adaptive metric which progressively models the notion of image similarity through continual relevance feedback from users. Experimental results show that the proposed methods not only outperform conventional CBIR systems in terms of both accuracy and robustness, but also previously proposed interactive systems.**

***Index Terms*—Content-based image retrieval, digital library, relevance feedback, machine learning, radial basis function network, nonlinear human perception.**

## I. Introduction

THERE IS an urgent need for new techniques to access visual data, following the explosion of digital media. This need has occurred in such diverse areas of application such as the entertainment industry, distance education, telemedicine and geographic information systems. Content-based image retrieval (CBIR) is regarded as one of the most effective ways of accessing visual data. It deals with the retrieval of images based on the visual content itself such as color, shape and image structure instead of annotated text. Typical examples of retrieval systems attempting to perform these operations include QBIC [1], Virage [2], Photobook [3], VisualSEEk [4], and Netra [5].

The central problems regarding the retrieval task are concerned with "interpreting" the contents of the images in a collection and ranking them according to the degree of relevance to the user query. This 'interpretation' of image content involves extracting content information from the image and using this information to match the user's need. Knowing how to extract this information is not the only difficulty; another is knowing how to use it to decide *relevance*. The decision of relevance characterizing *user information need* is a complex problem.

Traditional CBIR systems usually adopt "index features" in index and retrieval. In its general form, an index feature is a numerical value which characterizes the color, texture or shape information of individual images. Retrieval based on index features is a simple idea but raises key questions. For instance, it adopts the fundamental principle that the high-level concepts of the images, and the user information need, can be naturally expressed through sets of index features. This assumption is an oversimplification of the problem in hand; most of the semantics in an image or a user request are lost when we replace its content with a set of features. Furthermore, matching between each image and the user request is highly problematic in this very imprecise space of index features.

These difficulties have attracted broad interests in the research and development of retrieval systems and techniques. To be effective in satisfying user information need, a retrieval system must view the retrieval problem as "human-centered" rather than "computer-centered". In a number of recent papers [6]–[9], an alternative to the "computer-centered" predicate was proposed. This new approach is based on a "human–computer" interface, which enhances the system to perform retrieval tasks in line with human capabilities. The activities in this approach consist mainly of analyzing a user's goals from his/her feedback information on the desired images, and adjusting the search strategy accordingly. Under this paradigm, the user manages the retrieval system via the interface through selections of information gathered during each interactive session, to address information needs which are not satisfied by a single retrieved set of images.

The human–computer interface has been less understood than other aspects of image retrieval, partly because humans are more complex than computer systems and their motivations and behaviors are more difficult to measure and characterize. Recent studies have been conducted to simulate human perception of visual contents via the use of the supervised analysis method. Themes are derived from similarity functions through the assignment of numerical *weights* to the pre-extracted features. The weighted Euclidean is typically adopted to characterize the differences between images, so that distinct weights have varying relevance when used in the simulations (see [9]–[11] for examples). This idea can be further generalized by incorporating limited adaptivity in the form of a relevance feedback scheme [6]–[8], [12]–[14]. Here weighting is modified according to the users' preference. As seen from the users' viewpoint, the limited number of adjustable parameters, and the restriction of the distance measure to a quadratic form, may not be adequate for modeling perceptual difference.

Manuscript received July 18, 2002; revised December 18, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Shih-Fu Chang.

P. Muneesawang is with the Department of Electrical and Computer Engineering, Naresuan University, Phisanuloke, Thailand 65000 (e-mail: pmuneesa@ee.ryerson.ca).

L. Guan is with the Department of Electrical and Computer Engineering, Ryerson Polytechnic University, Toronto, ON M5B 2K3 Canada (e-mail: lguan@ee.ryerson.ca).

Digital Object Identifier 10.1109/TMM.2004.834866

In this paper, we attempt to address some of the aforementioned problems by proposing a *nonlinear* approach to simulate human perception. This allows for effectively bridging the gap between the low-level features used in retrieval and the high-level semantics in human perception. We replace the traditional relevance feedback with a specialized radial-basis function (RBF) network [15], [16] for learning the users' notion of similarity between images. In each interactive retrieval session, the user is asked to separate, from the set of retrieved images, those which are more similar to the query image from those which are less similar. The feature vectors extracted from these classified images are then used as training examples to determine the centers and widths of the different RBF's in the network. This concept is adaptively redefined in accordance with different users' preferences and different types of images, instead of relying on any preconceived notion of similarity through the enforcement of a fixed metric. Compared to the conventional quadratic measure and the limited adaptivity allowed by its weighted form, the current approach offers an expanded set of adjustable parameters, in the form of the RBF centers and widths. This allows a more accurate modeling of the notion of similarity from the users' viewpoint.

We then describe two applications of the RBF method and the experimental results. The first domain application is texture retrieval. In this domain, content-based image retrieval is very useful for queries involving texture patterns that represent a region of interest in a large collection of satellite air photos, as demonstrated in [27], [33], [34]. In the second domain, we propose an interactive search engine, iARM: Interactive-based Analysis and Retrieval of Multimedia system to support image searching tools in large image collections over the Internet. This is the ultimate goal in this context since multimedia over the internet is in very high demand. In order to implement a real-time learning application, we require a user-friendly system in the following senses: high accuracy, sufficiently fast (a few user feedbacks), and the capability of learning from a small training size (typically less than 20 per round of interaction). A new architecture of iARM takes into account all of these important features, using the proposed nonlinear analysis, with both positive and negative learning. As reported in Section VI, our proposed system compared favorably with those of other recently proposed interactive systems.

This paper is organized as follows. In Section II, we introduce a nonlinear model to simulate human perception, in comparison with the previous linear approaches. In Section III, we describe the proposed RBF network, and its discriminant function. In Section IV, the corresponding supervised learning strategies are proposed to enable the effectiveness in similarity modeling of the network. A detailed comparison with the performance of the nonadaptive method and other interactive systems is presented in Sections V and Section VI. Conclusions are drawn in Section VII.

## II. General Framework

In this section, we discuss general concepts of the proposed method, pointing out the differences between the assumptions behind the current work and the relevant models proposed in the literature. We also describe the motivations in designing the proposed system architecture.

### A. Human Perceptual Simulation

The most important part in the interactive process is to analyze the role of the users in perceiving image similarity according to their preferred image selections. To perform the analysis, we propose a nonlinear model to establish the link between human perception and distance calculation. In other words, our objective is to perform a nonlinear transformation of the distance.

Many attempts to perform similarity-analyzing have focused on linear models. Thus, a brief introduction to linear-based approaches and their limitations is in order. The approaches can be organized into two categories: 1) an approach based on a query reformulation model ( [7], [12], [14]); and 2) an approach based on an adaptive metric model ([10], [11], [17], [19]). In the first category, the models implement relevance feedback for learning query representation with the goal of conducting *term* weighting to modify the query. In the second category, the models implement relevance feedback for the learning similarity function.

In general, the learning user perception methods implements a mapping $f : \mathcal{R}^P \rightarrow \mathcal{R}$ which is given by

$$y_s = f(\mathbf{x}) \tag{1}$$

where $\mathbf{x} = [x_1, \ldots, x_P]^T \in \mathcal{R}^P$ is the input vector corresponding to an image in the database. The main procedure is to obtain the mapping function $f$ from a *small* set of training images, $\{(\mathbf{x}_1, l_1), (\mathbf{x}_2, l_2), \ldots, (\mathbf{x}_N, l_N)\}$, where the two-class label $l_i$ can be in binary or nonbinary form.

Among the early attempts in the interactive CBIR systems, Multimedia Analysis and Retrieval System, Version 1 (MARS-1) [8], [12] implemented the mapping in the form of the query reformulation model

$$y_s = f_{\text{cosine}}(\mathbf{x}, \mathbf{x}_{\hat{q}}) \tag{2}$$

where

$$\mathbf{x}_{\hat{q}} = \alpha \mathbf{x}_q + \gamma(\underset{l_i=1}{\text{mean}}\{\mathbf{x}_i\}) - \varepsilon(\underset{l_i=0}{\text{mean}}\{\mathbf{x}_i\}). \tag{3}$$

The query model $\mathbf{x}_{\hat{q}}$ is obtained by adjusting the positive and negative weight *terms* of the original query $\mathbf{x}_q$. Although simple, this model has been widely used for adaptive information retrieval (IR) [28], [30] and many image retrieval systems [7], [14]. A chief disadvantage of this integration model is the requirement of an indexing structure to follow term-weighting model used in text retrievals (for effectiveness). More specifically, the model works on the assumption that the query index terms are sparse and are usually binary vector representation. However, in content-based image retrieval, vectors are mostly real vectors.

In the later state, weight distance is a common strategy for obtaining the mapping function. This is the case in [9], [10], [13], [19], and in the MARS-2 (Multimedia Analysis and Retrieval

System, Version 2) system [11]. In its general form, the similarity function may be described as

$$f(\mathbf{x}, \mathbf{x}_q) = \sum_{i=1}^{P} h(d_i) \tag{4}$$

$$= (\mathbf{x} - \mathbf{x}_q)^T W (\mathbf{x} - \mathbf{x}_q) \tag{5}$$

where $h(d_i)$ denotes a one-dimensional (1-D) transfer function of distance $d_i \equiv |x_i - x_{qi}|$, and $W$ is a *block-diagonal* matrix with the following structure:

$$W = \text{diag}[w_1, w_2, \ldots, w_P]. \tag{6}$$

The weight parameters $w_i, i = 1, \ldots, P$ are called *relevance weights* applied to the distance $d_i$ with the restriction $w_i > 0, \sum_i w_i = 1$. These can be estimated by the standard deviation criterion as in [11] or a probability method [9].

Based on (4), Rui *et al.* [17] has derived an optimum solution for the similarity function and the query model. This method is referred to as an optimal learning relevance feedback (OPT-RF). Using Lagrange multipliers, an optimum solution for a query model is the weighted average of the training samples:

$$\mathbf{x}_{\hat{q}} = \frac{\mathbf{X}^T \mathbf{v}}{\sum_{i=1}^{N} v_i} \tag{7}$$

where $\mathbf{v} = [v_1, v_2, \ldots, v_N]$ are the similarity scores specified by user, and $\mathbf{X}$ denotes an $N \times P$ matrix $\mathbf{X} = [\mathbf{x}_1 \ldots \mathbf{x}_N]^T$. The optimum solution for the weight matrix $W$ is obtained by

$$W = \begin{cases} (\det(C))^{\frac{1}{K}} C^{-1} & \det(C) \neq 0 \\ \text{diag}\left(\frac{1}{C_{11}}, \frac{1}{C_{22}}, \ldots, \frac{1}{C_{PP}}\right) & \text{otherwise} \end{cases} \tag{8}$$

where $C$ denotes the weighted covariance matrix

$$C_{rs} = \frac{\sum_{i=1}^{N} v_i (x_{ir} - x_{\hat{q}r})(x_{is} - x_{\hat{q}s})}{\sum_{i=1}^{N} v_i}, \quad r, s = 1, \ldots, P. \tag{9}$$

OPT-RF intelligently switches $W$ between a full matrix and a diagonal matrix, to overcome the possible singularity issue when the number of training samples $N$ is smaller than the dimensionality of the feature space $P$. However, this situation does not usually happen in image retrieval—particularly when images are modeled by multiple descriptors and only a *small* set of training samples (from user input) is preferred.

OPT-RF requires the user to specify relative weight parameters on each training image for its effectiveness. This is not an easy task for the user, and practically, we cannot afford performance degradation by human errors. In addition, the query model (7) only takes into account the positive samples (i.e., $v_i = 0$ is set for a negative sample). This yields only a *local* optimum for the small set of samples available at each feedback cycle.

In this paper, we refer to the methods outlined above as a linear-based learning that restricts the mapping function in quadratic form and cannot cope with a complex decision boundary. Although this learning method provides mathematical framework for evaluating image similarity, it is not competent for the nonlinear nature of human perception. For instance, the 1-D distance mapping $h(d_i)$ in (4) takes the following form:

$$h(d_i) = w_i d_i^2 \tag{10}$$

It has no nonlinear capability such as

$$\frac{\partial f(\mathbf{x})}{\partial d_i} = 2 w_i d_i \tag{11}$$

where $w_i$ is fixed to a numerical constant. That is, the linear mapping shows that the degree of similarity between two images is linearly proportional to the magnitude of their distances. In comparison, the assumption for our nonlinear approach is that the same portions of the distances do no always give the same degrees of similarity when judged by humans [20].

The visual section of the human brain uses a nonlinear processing system for tasks such as pattern recognition and classification [15]. We therefore propose using a nonlinear criterion in performing simulation task. The current work is mainly different from MARS-2, and OPT-RF in two aspects. First, we propose a nonlinear kernel for the evaluation of image similarity. Second, we take both positive and negative feedbacks for effectiveness of learning capability. Compared to MARS-1 [cf. (3)], our query model can be generally applied to the feature space without term-weighting transformation. By embedding these two properties, the proposed retrieval system shows a high performance in learning with small user feedback samples, and convergence occurs quickly (results shown in Section V).

### *B. Basic Model*

To simulate human perception, we propose a radial basis function (RBF) network [15], [16] as a nonlinear model for proximity evaluation between images. The nonlinear model is constructed by an input–output mapping function, $f(\mathbf{x})$, that uses feature values of input image $\mathbf{x}$ to evaluate the degree of similarity (according to a given query) by a combination of activation functions associated as a nonlinear transformation.

The estimation of the input–output mapping function, $f(\mathbf{x})$, is performed on the basis of a method called *regularization* [22]. In the context of a mapping problem, the idea of regularization is based on the *a priori* assumption about the form of the solution (i.e., the input–output mapping function $f(\mathbf{x})$). In its most common form, the input–output mapping function is *smooth*, in the sense that similar inputs correspond to similar outputs. In particular, the solution function that satisfies this regularization problem is given by the expansion of the radial basis function [23]. Based on the regularization method, we have utilized a 1-D Gaussian shaped radial basis function to form a basic model:

$$G(x) = \exp\left(-\frac{(x - z)^2}{2\sigma^2}\right) \tag{12}$$

where $z$ denotes the center of the function and $\sigma$ denotes its width. The activity of function $G(x)$ is to perform a Gaussian transformation of the distance $d \equiv |x - z|$, which describes the degree of similarity between the input $x$ and center of the function.

To estimate the input–output mapping function $f(\mathbf{x})$ the Gaussian RBF is expanded through both its center and width, yielding different RBF's which then form as an RBF network. Its expansion is implemented via interactive learning where the expanded RBF's can optimize weighting to capture human perception similarity as discussed in the following section.

## III. RBF Method

RBF networks possess an excellent nonlinear approximation capability [15], [16]. We utilize this property to design a system of locally tuned processing units to approximate the target nonlinear function $f(\mathbf{x})$. In the general solution, an approximation function obtained by the RBF networks takes the following form:

$$f(\mathbf{x}) = \sum_{j=1}^{N} w_j G(\mathbf{x}, \mathbf{z}_j) \tag{13}$$

$$= \sum_{j=1}^{N} w_j \exp\left(-\frac{1}{2\sigma_j^2}\sum_{i=1}^{P}(x_i - z_{ji})^2\right) \tag{14}$$

where $\mathbf{z}_j \in \mathcal{R}^P$ denotes the center of the function $G(\mathbf{x}, \mathbf{z}_j)$ and $\sigma_j$ denotes its width. There are $N$ Gaussian unites in this network. Their sums in the form of a linear superposition define the approximating function $f(\mathbf{x})$. With the *regularization* structure, the RBF network takes a one-to-one correspondence between the training input data $\mathbf{z}_j, j = 1, \ldots, N$ and the function $G(\mathbf{x}, \mathbf{z}_j)$.

A direct application of this network structure to (online learning) image retrieval is, however, considered prohibitively expensive to implement in computational terms for large $N$. It is also sufficient to reduce the network structure into a single unit, since image relevance identification requires only a two-class separation (for a given query). In the current work, with radial-basis function in mind, we associate a 1-D Gaussian-shaped RBF with each component of the feature vector as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{P} G_i(x_i, z_i) \tag{15}$$

$$= \sum_{i=1}^{P} \exp\left(-\frac{(x_i - z_i)^2}{2\sigma_i^2}\right) \tag{16}$$

where $\mathbf{z} = [z_1, \ldots, z_i, \ldots, z_P]^T$ is the adjustable query position or the center of the RBF function, $\sigma_i, i = 1, \ldots, P$ are the tuning parameters in the form of RBF widths, and $\mathbf{x} = [x_1, \ldots, x_i, \ldots, x_P]^T$ is the feature vector associated with an image in the database. Each RBF unit implements a Gaussian transformation which constructs a local approximation to a nonlinear input–output mapping. The magnitude of $f(\mathbf{x})$ represents the similarity between the input vector $\mathbf{x}$ and the query $\mathbf{z}$, where the highest similarity is attained when $\mathbf{x} = \mathbf{z}$. Based on our simulation study, the new single unit RBF network is effective in learning and quickly converges for one-class-relevance classification using small volume of training sets.

In this network structure, each RBF function is characterized by two adjustable parameters: the tuning parameters and the adjustable centers:

$$\{\sigma_i, z_i\}, \quad i = 1, \ldots, P, \tag{17}$$

to form a set of $P$ basis functions:

$$\{G_1(\sigma_1; z_1), G_2(\sigma_2; z_2), \ldots, G_P(\sigma_P; z_P)\}. \tag{18}$$

These parameters are estimated and updated via learning algorithms. The first assumption behind our learning algorithms is that the user's judgment of image similarity can be captured by a small number of pictorial features. This is an unequal bias toward the evaluation of image similarity. That is, given a semantic context, some pictorial features exhibit greater importance or "relevance" than others in the proximity evaluation. This is the same assumption which underlies image matching algorithms in [9], [24]. However, in this case, the weighting process is controlled by an expanded set of tuning parameters, $\sigma_i, i = 1, \ldots, P$, which reflects the relevance of individual features. If a feature is highly relevant, the value of $\sigma_i$ should be small to allow higher sensitivity to any change of the distance $d_i \equiv |x_i - z_i|$. In contrast, a large value of $\sigma_i$ is assigned to the nonrelevant features so that the corresponding vector component can be disregarded when determining its similarity, since the magnitude of $G_i(\cdot)$ is approximately equal to unity regardless of the distance $d_i$. The choice of $\sigma$ according to this criterion will be discussed in Section IV-E.

Our second assumption is about the relationship between the clustering of desired images in the $P$-dimensional feature space and the initial location of the query. For a given query image, its associated feature vector may not be in a position close enough to those stored vectors associated with the relevant images. This initial query may form as a decision region that contains only a local cluster of the desired images in the database. Our goal here is to associate this local cluster as prior information in order to describe a larger cluster of relevant images in the database. The description of this larger cluster of relevant images is built interactively with assistance from the user. This process is implemented by the RBF network through the adjustment of RBF centers, $z_i, i = 1, \ldots, P$, as will be described in the following section.

## IV. Learning Strategies

We propose learning algorithms which enable the RBF network to progressively model the notion of image similarity for effective searching. The image matching process is initiated when the user supplies a query image and the system retrieves the $N_{\mathrm{RT}}$ images in the databases which are closest to the query image. From these images the user selects those as relevant which are most similar to the current query image, while the rest are regarded as nonrelevant. The feature vectors extracted from these images are incorporated as training data for the RBF network to modify the centers and widths. The re-estimated RBF model is then used to evaluate the perceptual similarity in a new search, and the above process is repeated until the user is satisfied with the retrieval results.

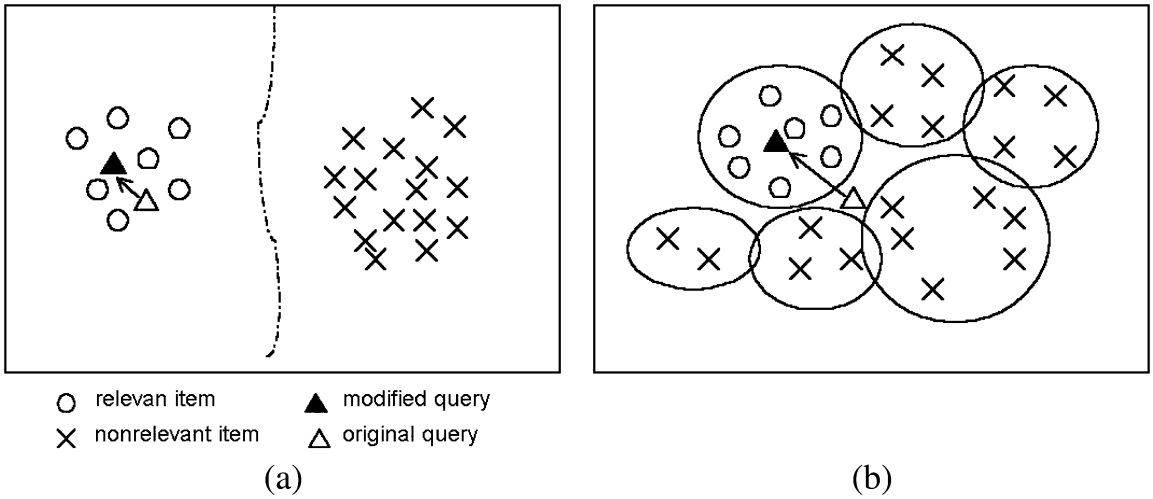


Fig. 1. Query modification. (a) Relevance judgment based on human vision. (b) Relevance clustering in the feature space. In (a), given a set of image collection, the human user may easily distinguish the relevant images from the high-level semantics according to his/her own understanding and expression of information need. In contrast, the low-level feature vector of the query in (b) is likely to be located in a different position in the feature space and may not be a representative sample of the relevant class.

### A. Centers Selection

Given a set of images, the human user may easily distinguish the relevant and nonrelevant images according to their own information needs [Fig. 1(a)]. In contrast, a computer interprets relevance as the distance between low-level image features [Fig. 1(b)] which could be very different from that shown in Fig. 1(a). The low-level vector of the query is likely to be located in a different position in the feature space and may not be a representative sample of the relevant class. To improve computer retrieval performance we modify the low-level query vector via the proposed learning algorithm. This aims at optimizing the current search. The expected effect is that the new query will move toward the relevant items (corresponding to the desired images) and away from the nonrelevant ones, whereas the user's information need remains the same throughout the query modifying process. In the following discussion, we first describe the basic optimization procedure, learning vector quantization (LVQ) [15], and then we propose a modified LVQ to obtain a proper choice for the RBF center associated with the new query vector.

### B. Learning Vector Quantization

LVQ [15] is a supervised learning technique used to optimize vector structures in a *code book* for the purpose of data compression [26]. The initial vectors (in a codebook) referred to as Voronoi vectors are modified in such a way that all points partitioned in the same Voronoi cells have the minimum (overall) encoding distortion. The technique uses the class information provided in a training set to move the Voronoi vectors slightly, so as to improve the accuracy of classification. Let the input vector $\mathbf{x}$ be one of the samples in the training set. If the class labels of the input vector $\mathbf{x}$ and a Voronoi vector $\mathbf{z}$ agree, the Voronoi vector $\mathbf{z}$ is moved in the direction of the input vector $\mathbf{x}$. On the other hand, if the class labels of the input vector $\mathbf{x}$ and the Voronoi vector $\mathbf{z}$ disagree, the Voronoi vector $\mathbf{z}$ is moved away from the input vector $\mathbf{x}$.

The modification of the Voroinoi vectors is usually carried out by an iterative process, where $n = 0, 1, 2, \ldots, n_{\max -1}$ is the step index. Let $\{\mathbf{z}_j\}_{j=1}^{J}$ denote the set of Voronoi vectors. Also, let $\{\mathbf{x}_i\}_{i=1}^{N}$ denote the set of training samples. First, for each input vector $\mathbf{x}_i(n)$, the index $c(\mathbf{x}_i)$ of the best-matching Voroinoi vector $\mathbf{z}_c(n)$ is identified by the condition:

$$c = \arg\min_j \{||\mathbf{x}_i - \mathbf{z}_j||\}. \tag{19}$$

Let $\ell_{\mathbf{z}_c}$ denote the class associated with the Voronoi vector $\mathbf{z}_c$, and $\ell_{\mathbf{x}_i}$ denote the class label of the input vector $\mathbf{x}_i$. The Voroinoi vector $\mathbf{z}_c$ is adjusted as follows:

If $\ell_{\mathbf{z}_c} = \ell_{\mathbf{x}_i}$, then (*reinforced learning*)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]. \tag{20}$$

If, on the other hand, $\ell_{\mathbf{z}_c} \neq \ell_{\mathbf{x}_i}$, then (*anti-reinforced learning*)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)]. \tag{21}$$

Note that, for all $j \neq c, \mathbf{z}_j(n+1) = \mathbf{z}_j(n)$, those Voronoi vectors remain unchanged. Here, the learning constant $\alpha_n$ decreases monotonically with the number of iterations and $0 < \alpha_n < 1$. After several passes through the training data, the Voronoi vectors typically converge, and the training process completes.

Based on the reinforced learning rule, it is clearly shown that the above process tries to move the Voronoi vector $\mathbf{z}_c$ to some points in the input space that are close to those samples which have the same class labels. At the same time the anti-reinforced learning rule moves $\mathbf{z}_c$ away from those samples which are in different classes. This process results in a new set of the Voronoi vectors $\{\tilde{\mathbf{z}}_j\}_{j=1}^{J}$ that minimizes (overall) encoding distortion.

### C. A Modified LVQ

In an interactive retrieval session, it is desirable to reduce the processing time to a minimum without affecting the overall per-

formance. So, we can integrate the LVQ method for query modification, in which we can avoid the implementation of the *iterative* procedure. This minimizes the time complexity of the process $\mathcal{O}(n_{\max})$, where $n_{\max}$ is the total number of iterations.

In image retrieval, we can cluster the database feature space into a number of distinct Voronoi cells with associated Voronoi vectors. Furthermore, the Voronoi vectors may be individually initialized by query vectors. Each Voronoi cell contains a set of feature vectors associated with those retrieved images that are the closest to the corresponding query, according to the nearest-neighbor rule based on the Euclidean metric. Our goal, here, is to optimize these cells by employing the LVQ algorithm. Since only one query is submitted at a particular time, only two partitions are necessary in the space, with one representing the relevant image set. The LVQ algorithm is then adopted to modify this cell from its corresponding training data.

Let the Voronoi vector $\mathbf{z}_q(t)$ denote the submitted query at a retrieval session $t$. Recall that the information input from the user at the interactive cycle is formed as the training set $\mathcal{T}$ that contains training vectors belonging to two separate classes:

$$\mathcal{T}_{(t)} = (\mathbf{x}_i, l_i), \quad i = 1, \ldots, N \tag{22}$$

$$= \{\mathbf{x}'_m, m = 1, \ldots, M | l_m = 1\} \cup \{\mathbf{x}''_q, q = 1, \ldots, Q | l_q = 0\} \tag{23}$$

where $\mathbf{x}_i \in \mathcal{R}^P$ is a feature vector and $l_i \in \{0, 1\}$ is a class label. The set of vectors in (22) represents the set of points closest to the submitted query $\mathbf{z}_q(t)$ according to the distance calculation in the previous search operation. Consequently, each data point can be regarded as the vector $\mathbf{x}_i$ that is "closest" to the Voronoi vector $\mathbf{z}_q(t)$. Therefore, following the LVQ algorithm, we see that all points in this training set are used to modify only the best-matching Voronoi vector, that is, $\mathbf{z}_q(t)$.

*Model 1:* According to our previous discussion, after the training process is converted the modified Voronoi vector $\tilde{\mathbf{z}}$ will lie close to the data points that are in the same class and away from those points that are in a different class. Combing these ideas, we now obtain a modified LVQ algorithm, to adjust the query vector $\mathbf{z}_q(t)$, by approximating the modified Voronoi vector $\tilde{\mathbf{z}}_q$ upon convergence:

$$\mathbf{z}_q(t+1) = \mathbf{z}_q(t) + \alpha_R(\bar{\mathbf{x}}' - \mathbf{z}_q(t)) - \alpha_N(\bar{\mathbf{x}}'' - \mathbf{z}_q(t)) \tag{24}$$

$$\bar{\mathbf{x}}' = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}'_m \tag{25}$$

$$\bar{\mathbf{x}}'' = \frac{1}{Q}\sum_{q=1}^{Q} \mathbf{x}''_q \tag{26}$$

where $\mathbf{z}_q(t)$ is the previous query, $\mathbf{x}'_m = [x'_{m1}, \ldots, x'_{mi}, \ldots, x'_{mP}]^T$ is the $m$th feature vector of relevant images, $\mathbf{x}''_q = [x''_{q1}, \ldots, x''_{qi}, \ldots, x''_{qP}]^T$ is the $q$th feature vector of nonrelevant images; $\alpha_R$ and $\alpha_N$ are suitable positive constants; $M$ and $Q$ are, respectively, the number of relevant and nonrelevant images in the training set. The application of the query modification in (24) is to allow the new query, $\mathbf{z}_q(t+1)$, to move toward the new region populated by the relevant images as well as to move away from those regions populated by the nonrelevant images.

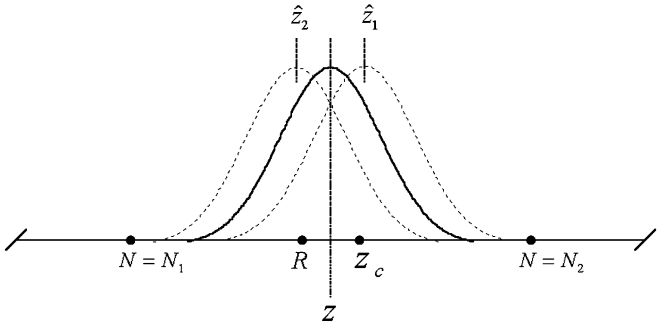


Fig. 2. Illustration of query modification.

Equation (24) can be illustrated in Fig. 2. Let the centers of the relevant image set and nonrelevant image set in the training data, be $R$ and $N$, respectively. Also, let $\mathbf{z}_q(t) = z_c$. As shown in Fig. 2, the effect of the second term on the right hand side of (24) is to allow the new query to move toward $R$. If in $N = N_1 < z_q(t)$, the third term is negative; so, the current query will move to the right, i.e., the position of $z_q(t)$ will shift away from $N_1$ to $\hat{z}_1$. On the other hand, when $N = N_2 > z_q(t)$, the third term is positive, hence $z_q(t)$ will move to the left or $\hat{z}_2$; i.e., away from $N_2$.

In practice, one finds that the relevant image set is more important in determining the modified query than the nonrelevant images. This is because the set of relevant images is usually tightly clustered due to the similarities between its member images, and thus satisfies the modified query with little ambiguity. On the other hand, the set of nonrelevant images is much more heterogeneous, therefore, the centroid of this nonrelevant image set may be located almost anywhere in the feature space. As a result, we have chosen $\alpha_R > \alpha_N$ in (24) to allow a more definite movement toward the set of relevant images, while permitting slight movement away from the nonrelevant regions.

*Model 2:* In order to provide a simpler procedure and a direct movement of the new query toward the relevant set, (24) is reduced to

$$\mathbf{z}_q(t+1) = \bar{\mathbf{x}}' - \alpha_N(\bar{\mathbf{x}}'' - \mathbf{z}_q(t)). \tag{27}$$

The first and the second terms in the right hand side of (24) are replaced by $\bar{\mathbf{x}}'$ (centroid of the relevant vectors). Since the relevant image group indicates the user's preference, the presentation of $\bar{\mathbf{x}}'$ for the new query will give a reasonable representation of the desired image. In particular, the mean value $\bar{x}'_i = (1/M)\sum_{m=1}^{M} x'_{mi}$ is a statistical measure providing a good representation of the $i$th feature component since this is the value which minimizes the average distance $(1/M)\sum_{m=1}^{M}(x'_{mi} - \bar{x}'_i)^2$. Further, the exclusion of the parameter $\alpha_R$ from (24) permits greater flexibility, since only one procedural parameter is necessary for the final fine tuning of a new query.

### *D. Effects of Query Adaptation Process (by Positive and Negative Learning)*

Different users may provide different sets of fed-back images. Thus, a modified query can be characterized by two types of

topic. One is specified by a common interest among users (i.e., a mean value of the relevant samples), and another by a specific topic that depends on the subjectivity of individual perception.

In practice, particularly with a general image collection, retrieval results obtained after the first round usually contain few relevant images, thus users with similar interests rate images similarly. This means that a modified query (obtained from this first round images, $\mathcal{T}_{t=0} = \{(\mathbf{x}_i, l_i)\}_{i=1}^{N}$), is built based on the common interests, as most users would make the same distinction. However, after the next round of retrieval, it is most likely that retrieval system will bring up images that are close to the specific topic of the user interests. Formally, let $\mathrm{Score}(\mathbf{x}_i)$ denotes the user judgment score for retrieved image $\mathbf{x}_i, i \in [1, \ldots, N]$. In most situations, we see that a new retrieval image set should be better than the old one, such that

$$\mathrm{Average}\{\mathrm{Score}(\mathbf{x}_i) \,|\, \mathbf{x}_i \in \mathcal{T}_{t+1}\} > \mathrm{Average}\{\mathrm{Score}(\mathbf{x}_i) \,|\, \mathbf{x}_i \in \mathcal{T}_t\}$$

in particular for negative samples:

$$\mathrm{Average}\{\mathrm{Score}(\mathbf{x}_i) \,|\, (\mathbf{x}_i \in \mathcal{T}_{t+1}) \wedge (l_i = 0)\} > \mathrm{Average}\{\mathrm{Score}(\mathbf{x}_i) \,|\, (\mathbf{x}_i \in \mathcal{T}_t) \wedge (l_i = 0)\}.$$

In this round, new positive and negative samples are tightly-clustered, and discrimination of these images can only be made by user subjectivity. By application of the anti-reinforced learning in (24) and (27), we can decouple centroids of these positive and negative samples. In other words, positive feedback learning tells us about the means values of the user's topic of interest, while negative samples specify the user subjectivity for that particular group of interest.

### *E. Selection of RBF Width*

As we have previously discussed, the nonlinear transformation associated with the output unit(s) of the Gaussian-shaped RBF are adjusted in accordance with different users' preference and different types of images. Through the proximity evaluation, differential biases are assigned to each feature, while features with higher relevance degrees are emphasized, and those with lower degrees are de-emphasized.

Consider that for a particular query location $\mathbf{z} = [z_1, \ldots, z_i, \ldots, z_P]^T$, the training samples can be described by the set of feature vectors $\{\mathbf{x}_i\}_{i=1}^{N}$ as in (22). To estimate the relevance of individual features, only the vectors associated with the set of relevant images in this training set are used to form an $M \times P$ feature matrix $\mathbf{R}$:

$$\mathbf{R} = [\mathbf{x}'_1, \ldots, \mathbf{x}'_m, \ldots, \mathbf{x}'_M]^T = [x'_{mi}] m = 1, \ldots, M, \quad i = 1, \ldots, P \tag{28}$$

where $\mathbf{x}'_m = [x'_{m1}, \ldots, x'_{mi}, \ldots, x'_{mP}]^T$ corresponds to one of the images marked as relevant; $x'_{mi}$ is the $i$th component of the feature vector $\mathbf{x}'_m$; $P$ is the total number of features; and $M$ is the number of relevant images. According to our previous discussion, the tuning parameters $\sigma_i$ should reflect the relevance of individual features. It was proposed, in [9], [24], that given a particular numerical value $z_i$ for a component of the query vector, the length of the interval which completely encloses $z_i$ and a pre-determined number $L$ of the set of values $x'_{mi}$ in the relevant set which falls into its vicinity, is a good indication of the relevancy of the feature. In other words, the relevancy of the $i$th feature is related to the density of $x'_{mi}$ around $z_i$, which is inversely proportional to the length of the interval. A large density usually indicates high relevancy for a particular feature, while a low density implies that the corresponding feature is not critical to the similarity characterization. Setting $L = M$, the set of tuning parameters is thus estimated as follows:

$$\sigma_i = \eta \max_m |x'_{mi} - z_i| \tag{29}$$

The factor $\eta$ guarantees a reasonably large output $G(\cdot)$ for the Gaussian RBF unit, which indicates the degree of similarity, e.g., $\eta = 3$.

We also consider a second criterion for estimating the tuning parameters. This is obtained by nonlinear weighting the sample variance in the relevant set as follows:

$$\sigma_i = \exp(\beta \cdot \mathrm{Std}_i) \tag{30}$$

$$\mathrm{Std}_i = \left( \frac{1}{M-1} \sum_{m=1}^{M} (x'_{mi} - \bar{x}'_i)^2 \right)^{\frac{1}{2}} \tag{31}$$

where $\mathrm{Std}_i$ is the standard deviation of the members $x'_{mi}, m = 1, \ldots, M$, which is inversely proportional to their density (Gaussian distribution). The parameter $\beta$ can be chosen to maximize or minimize the influence of $\mathrm{Std}_i$ on $\sigma_i$. For example, when $\beta$ is large, a change in $\mathrm{Std}_i$ will be exponentially reflected in $\sigma_i$. The exponential relationship is more sensitive to changes in relevancy and gives rise to better performance improvement, as we shall see in the experiment.

As a result, (29)–(30) provide a small value of $\sigma_i$ if the $i$th feature is highly relevant (i.e., the sample variance in the relevant set $\{x'_{mi}\}_{m=1}^{M}$, is small). This allows higher sensitivity to any change of the distance $d_i \equiv |x_i - z_i|$. In contrast, a high value of $\sigma_i$ is assigned to the nonrelevant features so that the corresponding vector component can be disregarded when determining the similarity.

TABLE I
AVERAGE RETRIEVAL RATE (%) FOR THE 39 QUERY IMAGES IN MIT DATABASE, USING GABOR TEXTURE FEATURE REPRESENTATION

| Method | t=0 | t=1 | t=2 | t=3 | Parameters |
|---|---|---|---|---|---|
| RBF1 | 74.36 | 90.06 | 92.95 | 93.59 | $\alpha_R$ =1.4, $\alpha_N$ = 0.4, $\beta$ = 2.6 |
| RBF2 | 74.36 | 88.62 | 91.67 | 92.79 | $\alpha_N$ = 0.65 |
| MARS-1 | 64.26 | 77.73 | 79.97 | 80.13 | ($\alpha$, $\gamma$, $\varepsilon$) = (1, 5, 0.5) |

## V. EXPERIMENTAL RESULTS PART 1—TEXTURE RETRIEVAL

In the experiments, we study the retrieval performance of the nonlinear RBF approach to two image retrieval application domains. This section describes the application on texture pattern retrieval, and Section VI describes the application on a large collection of photographs. When evaluating image-retrieval algorithms, there are several factors that determine the choice of a particular algorithm for an application. Central concerns are retrieval accuracy and CPU time. The retrieval accuracy is evaluated by a specific ground truth on a given database. For the

adaptive retrieval algorithms, however, there are additional factors, such as the size of the training set, and the convergence speed. For each domain application, we will evaluate the proposed RBF algorithm, and compare it to other interactive systems, using these factors.

The importance of texture analysis and retrieval has been well demonstrated by the works of Manjunath [27], [33], [34], on the large collection of satellite images and air photos. In this context, content-based retrieval is very useful for such queries as, "Retrieve all LANDSAT images of Santa Barbara which have less than 20% cloud cover," or "Find a vegetation patch that looks like this region." We regard this context as important, and would like to demonstrate our proposed approach to this application.

In the following we compare the RBF's retrieval performance with MARS-1, which was developed early in the texture retrieval domain [8], [12]. The retrieval strategy in MARS-1 has also been extended and used in other works, such as [6], [14]. The comparison methods are summarized as follows:

1) The Radial basis function (RBF) methods: the RBF1 method uses model 1 determining the RBF center [(24)], and (30) for the RBF width. The RBF2 method uses model 2 for determining the RBF center [(27)] and (29) for learning RBF width.
2) The relevance feedback method (RFM) is described in the MARS-1 system [8], [12], is employed by the PicToSeek system [6], and is also used in [7], [13], [14]. In each of these systems, the RFM is implemented differently. We briefly describe these systems in this section.
3) Method 3: simple CBIR using a noninteractive retrieval method, which corresponds to the first iteration of interactive search. This method employs different types of similarity functions, including weighted distance [as in (32)], cosine distance, and the histogram intersection, corresponding to the first iteration found in RBF, MARS, and PicToSeek.

MARS-1 and PicToSeek systems implement relevance feedback based on the query modification model [cf. (3)]. Thus, these systems require an indexing structure as in a term-weighting model [28]. MARS-1 used an integrating version of the well-known TF $\times$ IDF factor for the conversion of image features to a weighted vector model, which can be applied to different types of image feature structures. In contrast, the weighting technique of PicToSeek is the direct application of the TF $\times$ IDF factor, which is only applied for image features that have been treated in histogram form (i.e., sparse vector [6]). In the following experiment, this weighting technique applies only to one of the two feature types. The comparison results in the following section are, therefore, made only with MARS-1, while Section V-D reports a comparison with PicToSeek.

### A. Databases and Ground Truth Classes

Performance evaluations of retrieval in the experiments were carried out using two standard texture databases: 1) the MIT texture collections and 2) the Bordatz database [27]. In the first database, the original test images were obtained from MIT Media Laboratories.[1] There were 39 texture images from different classes, each $512 \times 512$ pixels in size. Each of these images was divided into 16 nonoverlapping subimages, $128 \times 128$ in size, creating a database of 624 texture images. In the second database, texture images and a feature set were created by Ma [27] at UCSB[2]. The Brodatz database contains 1856 patterns obtained from 116 different texture classes. Each class contains 16 similar patterns.

One advantage of using these databases is the corresponding ground truth data, which is known as the set of visually similar images for a given query image. Although, in general, this data is hard to obtain and is very subjective to a particular user, the method of dividing an image into subimages to obtain this ground truth data is a popular method used in other situations. This includes the work found in [9], [12], [27], [33]. It is based on the fact that subimages in the same classes were obtained from one large image and classified according to visual similarity as perceived by the user. This was established through a process of visual inspection and cross-verification by different groups of people apart from the original users.

### B. Texture Feature Representations

Each texture image in the two databases is described by a 48-dimensional vector which characterizes the coefficients after applying Gabor wavelet transformation to the image [27]. The set of basis functions consists of Gabor wavelets spanning four scales ($S = 4$) and six orientations ($K = 6$). The mean and standard deviation of the transform coefficients form the feature vector $\mathbf{f} = [\mu_{00}\sigma_{00}\mu_{01}\cdots\mu_{(S-1)(K-1)}\sigma_{(S-1)(K-1)}]^T$ where $\mu_{mn}$ and $\sigma_{mn}$ are the mean and the standard deviation of the transform coefficients at the $m$th scales and $n$th orientations, respectively.

Since the dynamic range of each feature component is different, a suitable similarity measure for this feature is computed by the following distance measure [27]:

$$d(i,j) = \sum_m \sum_n d_{mn}(i,j) \tag{32}$$

where

$$d_{mn}(i,j) = \left| \frac{\mu_{mn}^{(i)} - \mu_{mn}^{(j)}}{\alpha(\mu_{mn})} \right| + \left| \frac{\alpha_{mn}^{(i)} - \alpha_{mn}^{(j)}}{\alpha(\alpha_{mn})} \right|.$$

$d_{mn}(i,j)$ denotes the distance between the two patterns in the feature space. In addition, $\alpha(\mu_{mn})$ and $\alpha(\sigma_{mn})$ are the standard deviations of the respective features, over all the images in the database. We employ (32) as a base line for similarity measure of the noninteractive approach, to perform a retrieval task based on the Gabor wavelet feature representation.

[1]ftp://whilechapel.media.mit.edu/pub/VisTex/, 2000.

[2]University of Califomia at Santa Barbara, http://vivaldi.ece.ucsb.edu/users/weilcodes.html, 2000. We thank Dr. W. Y. Ma for providing the Brodatz database and the software of the Gabor wavelet features.

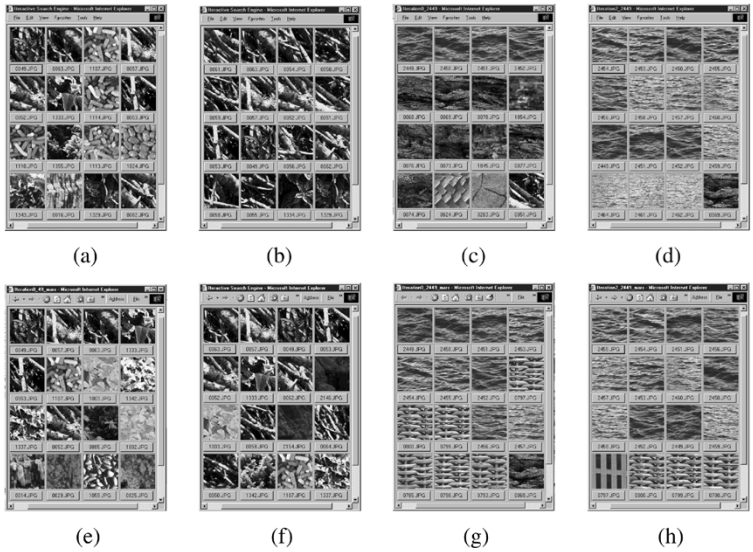

Fig. 3. Pattern retrieval results before and after learning similarity, using MIT database. Results show a comparison between RBF1 and MARS-1 using Gabor wavelet features: (a), (b), (e), and (f) show retrieval results for the query 'Bark.0003'; (c), (d), (g), and (h) show retrieval results for the query 'Water.0000'. In each case, the images are ordered according to decreasing similarity in the 16 best matches from left to right and top to bottom. (a) RBF1($t = 0$). (b) RBF1($t = 2$). (c) RBF1($t = 0$). (d) RBF1($t = 2$). (e) MARS-1($t = 0$). (f) MARS-1($t = 2$). (g) MARS-1($t = 0$). (h) MARS-1($t = 2$).

### C. Summary of Comparisons

In the simulation study, a total of 39 images, one from each class, were selected as the query images from the MIT database. For each query, the top 16 images were retrieved to provide necessary relevance feedback. Using this method, in the ideal case all the top 16 retrievals are from the same classes. The performance was measured in terms of average retrieval rate (AVR) of the 39 query images, which was defined by [27]:

$$\text{retrieval rate} = \frac{\text{relevant images}}{\text{class size}} \times 100\%.$$

Similarly, a total of 116 images, one from each class, were selected as the query images from Brodatz database. The performance was measured in terms of average retrieval rate of the 116 query images.

Table I summarizes average retrieval rate of the 39 query images, using the MIT database, where $t$ denotes the number of iterations. The following observations are made from the results.

*First*, for all methods, the performance with the interactive learning method after three iterations $(t = 3)$ was substantially better than the noninteractive cases $(t = 0)$. The improvements are quite striking. *Second*, after three rounds of interactive learning, RBF1 method gave the best performance: on average 93.59% of the correct images are in the top 16 retrieved images (i.e., more than 14 of the 16 correct images are present). This is closely followed by RBF2 at 92.79% of correct retrieval. These results show that the RBF methods perform substantially better than MARS-1, which provides a retrieval performance of 80.13%. It is also observed that the RBF methods provide much better results after one iteration (88.62%) than MARS-1 after three iterations (80.13%). *Third*, for all the three interactive methods, convergence is achieved within a few iterations.

Fig. 3 shows two examples of retrieval sessions performed by RBF1 in comparison with MARS-1. They clearly illustrate the superiority of the proposed method. We observed that RBF1 considerably enhanced retrieval performance, both visually and statistically. In addition, given the small number of training samples (e.g., 16 retrieved images used in training), the RBF approach can effectively learn and capture user input on image similarity.

TABLE II
AVERAGE RETRIEVAL RATE (%) OBTAINED BY RETRIEVING 116 QUERY IMAGES IN THE BRODATZ DATABASE, USING GABOR WAVELET REPRESENTATION

| Method | t=0 | t=1 | t=2 | t=3 | Parameters |
|---|---|---|---|---|---|
| RBF1 | 73.71 | 85.18 | 88.31 | 90.52 | $\alpha_R$ =0.8, $\alpha_N$ = 0.7, $\beta$ = 4.6 |
| RBF2 | 73.71 | 83.90 | 86.43 | 87.62 | $\alpha_N$ = 0.5 |
| MARS-1 | 67.10 | 75.74 | 77.76 | 78.46 | ($\alpha$, $\gamma$, $\varepsilon$) = (1, 11, 2.5) |

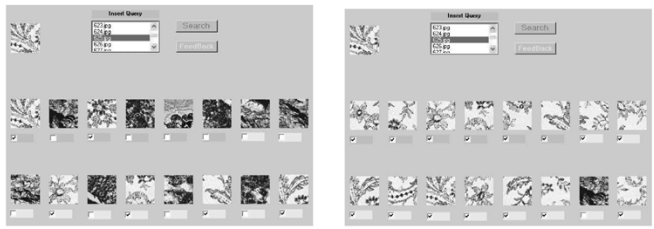

Fig. 4. Top 16 retrievals obtained by retrieving texture "D625" from Brodatz database, using RBF1; (a) results before learning and (b) results after learning.

Table II summarizes AVR (%) obtained by retrieving 116 query images, using Brodatz database. It can be seen that all interactive methods demonstrate significant performance improvement across the tasks. In the final results after learning, RBF1 gave the best performance at 90.5% of correct retrieval, followed by RBF2 (87.6%), with MARS-1 (78.5%) a distant third. We observed that characteristics of retrieval results obtained from the Brodatz database are very similar to those obtained from the MIT database. This implies that RBF1 consistently displays superior performance over MARS-1.

Fig. 4 illustrates retrieval examples with and without learning similarity. It shows some of the difficult patterns to analyze, which clearly illustrate the superiority of the RBF method.

TABLE III
AVERAGE RETRIEVAL RATE (%) OBTAINED BY RETRIEVING 39 QUERY IMAGES IN THE MIT DATABASE, USING MHI FEATURES REPRESENTATION

| Method | t=0 | t=1 | t=2 | t=3 | Parameters |
|---|---|---|---|---|---|
| RBF1 | 63.46 | 85.10 | 90.87 | 92.47 | $\alpha_R$ =1.4, $\alpha_N$ = 0.6, $\beta$ = 0.8 |
| RBF2 | 63.46 | 82.85 | 88.62 | 90.55 | $\alpha_N$ = 0.7 |
| MARS-1 | 63.94 | 80.77 | 84.46 | 85.42 | ($\alpha$, $\gamma$, $\varepsilon$) = (1, 8, 3) |
| PicToSeek | 62.18 | 77.08 | 83.97 | 84.87 | ($\alpha$, $\gamma$, $\varepsilon$) = (1, 8, 3) |

Further, it was observed that retrieval performance of the interactive method is sensitive to the choice of the associated parameters used in the learning algorithms. With proper selection of parameters, interactive methods can achieve the best retrieval performance. Those parameters that gave the optimum retrieval results are listed in the last column of Tables I and II. The procedural parameters $(\alpha, \gamma, \epsilon)$ of MARS-1 system in (3) were determined according to the standard formula studied by Ricchio [30]. The constant $\alpha$ is fixed to 1 and the constants $\gamma, \epsilon$ are varied to obtain the best retrieval results.

### D. Using Compressed Domain Features for Retrieval

In this experiment, we apply "relevance feedback" learning to a compressed domain image retrieval system, using images compressed by wavelet transform (WT) and vector quantization (VQ) coders [32]. Specifically, image indexing and retrieval are directly performed on the compressed data. This is advantageous in terms of computational efficiency. We use the multiresolution-histogram indexing (MHI) proposed in [18] to describe the texture images. A two-level wavelet decomposition scheme with the 15-coefficient biorthogonal filter is adopted for the analysis of the image. The five subbands containing detail coefficients are then vector quantized by a *multiresolution codebook*, at a total bit rate of 1 bit/pixel. The coding labels are used to construct a feature vector by computing the labels histograms. MHI features make use of the fact that the usage of codewords in the subcodebook reflects the content of the input subimage that has been encoded.

An important reason for including the MHI features in the experiment is that this representation scheme is comparable to the *term-weighting model* of the information retrieval (IR) systems [28]. An image is decomposed into a set of VQ codewords, and the MHI vectors reflect the frequencies with which the codewords appear. Similarly, the basis for the *term-weighting model* is the representation of a document as a set of terms. This numeric vector is based on the commonsense understanding that the more often a term (or codeword) is mentioned, the more likely it is to be central to the subject matter. This *term-weighting model* is favored by the RFM method, particularly the query refinement formula (3), in which it has been effectively employed [14], [28], [29]. As a result, we can study how well the RFM method works with the term vector model compared to the proposed approach.

We now present experimental results with interactive approaches on MHI compressed domain features, using the MIT database. The experiments compare retrieval accuracy between RBF1, RBF2, MARS-1, and PicToSeek, for all 39 queries used in the previous experiments reported in Section V-C.

Results are shown in Table III, which presents the average retrieval rate as a function of the iteration. The following observations have been made:

1) With the application of interactive learning, each method shows considerable improvement of retrieval, indicating the effectiveness of learning strategies. This also implies that MHI features provide a very good representation in retrieving the compressed images. A combination of MHI features and interactive learning can achieve an average retrieval rate of more than 92% on this database.
2) RBF1 gave the best overall performance at 92.5% correct retrieval, which is closely followed by RBF2 at 90%. MARS-1 performed slightly better (85.4%) than PicToSeek (84.5%). The initial results of all methods were quite similar: the cosine distance performed marginally better (63.9%) than the normalized distance of (32) (63.5%), and the histogram intersection (62.2%).
3) Comparison of retrieval accuracy of the two feature representations, MHI and Gabor wavelet, as reported in the previous session, indicates that the MARS-1 worked well with MHI features (85.4%), and more efficiently than with Gabor wavelet features (80.1%). This is because MHI feature structure is closer to the term-weighting model, consequently, the query reformulation strategy (3) is more effective.[3] However, this also suggests that the learning strategy implemented by MARS-1 is not very robust, due to the requirements of this underlying feature structure.

## VI. EXPERIMENTAL RESULTS PART 2—APPLICATION TO A LARGE IMAGE COLLECTION

To address the challenging research issues involved in CBIR, an Interactive based Analysis and Retrieval of Multimedia (iARM) system has been conducted at Ryerson University [18], [21]. The interactive retrieval architecture proposed in this paper has been implemented on the Java 2 Enterprise Edition (J2EE) platform, which is available from http://www.ee.ryerson.ca/~pmuneesa.

We have used images from Corel Gallery [25] in these evaluations. Corel Gallery has a database which contains 40 000 real-life photographs, in two groups, each of which has either $384 \times 256$ or $256 \times 384$ pixels in size. It is organized into 400 categories by Corel professionals. These categories were used as a ground truth in our evaluation[4]. For indexing purposes, each image is characterized by visual descriptors using multiple types of features, $F = \{F_{\text{color}}, F_{\text{texture}}, F_{\text{shape}}\}$, where the representations are color histograms and color moments for color descriptors; GW transform for the texture descriptors [27]; and Fourier descriptor for the shape descriptors [35]. The algorithms for obtaining these descriptors are summarized in Table IV. Note that the resulting feature database (which is a matrix of size;

[3]The application of query reformulation strategy to the term-weighting model has been shown to have information theoretic motivation [14], [28], [29].

[4]The ground truth classes are subjected to the Corel professionals, and obtained with a high degree of semantic concepts. In addition, one may find some overlapping between the image classes.

TABLE IV
CONTENT DESCRIPTIONS USED FOR IMAGE INDEXING OF THE COREL DATABASE

| | |
|---|---|
| **Color Descriptors** | |
| *Color Histogram* ($d$=48)<br>*Bins=48* | The descriptor is a 48-bin color histogram in HSV color space, where $H$ and $S$ are uniformly quantized into 16 and 3 regions respectively. |
| *Color Moments* ($d$=9) | From the RGB color image, we extract mean, standard deviation, and skew from the three color channels and therefore have a color feature vector of length $3 \times 3 = 9$. |
| **Texture Descriptors** | |
| *GW transform* ($d$=48) | The image is resized into $128 \times 128$ pixels in size, and converted to the gray scale level. Gabor wavelet (GW) filters spanning four scales and six orientations are then applied to the gray scale image. The mean and standard deviation of the GW coefficients form the 48-dimension feature vector. |
| **Shape Descriptors** | |
| *Fourier Descriptors* ($d$=9) | The Sobel edge detection algorithm is applied to each color channel of the RGB color image. The resulting contour edge is characterized on polar coordinates. Fast Fourier transform (FFT) is then applied to the contour edge and the coefficients in the low frequency range are truncated to form a 9-dimension feature vector. |

40 000 by 114) was scaled by feature means and standard deviations to remove unequal dynamic range of each feature variable.

In the following simulation studies, we obtained the performance comparisons between the nonlinear RBF method, MARS-2 [11] and OPT-RF [17] systems (described in Section II). MARS-2 is relatively newer than MARS-1, and has been intensively tested on the large Corel image collection in [11]. This has become a popular benchmark for image retrieval. In [17], OPT-RF has been proven to be the most optimized framework currently used in interactive CBIR systems. The major differences between these two systems are that the learning algorithm in OPT-RF has both an optimum query and a switching option of the weight matrix $W$ [cf. (8)] between a full matrix and a diagonal matrix. Particularly in this practical application, as we have very high feature dimensions $(P = 114)$, we implement OPT-RF with $W$ in a diagonal matrix form. In the RBF case, relevance feedback learning is processed based on the Gaussian kernel, having a nonlinear decision criterion. In addition, RBF obtains automatic weighting to capture user perception, whereas OPT-RF requires users to specify weighting in the form of a slider bar [cf. Figs. 6(d) and 7(d)]. Moreover, RBF method uses both positive and negative samples to track the optimum query model. Neither OPT-RF nor MARS-2 support these features.

The average precision rates[5] (APR) and CPU time required are summarized in Table V. These are obtained using RBF model [(27) and (30)], MARS-2 system [(4)–(6)], and OPT-RF system [(7)–(9)]. Notice that all methods employ norm-1 metric distance to obtain initial retrieval results at $t = 0$. We have selected 35 queries each from different categories, which are shown in Fig. 5. The performances were measured from the top 16 retrievals, and averaged over all 35 queries.

Evidently, the nonlinear RBF method exhibits significant retrieval effectiveness, while offering more flexibility than

[5]Precision is defined by number of relevant images over the top sixteen retrievals.

TABLE V
AVERAGE PRECISION RATE (%) OBTAINED BY RETRIEVING 35 QUERIES SELECTED FROM DIFFERENT CATEGORIES, USING THE COREL DATABASE. AVERAGE CPU TIME (SECOND) OBTAINED BY RETRIEVING A SINGLE QUERY, NOT INCLUDING THE TIME FOR DISPLAY THE RETRIEVED IMAGES, USING A 1.8 GHz PENTIUM IV PROCESSOR AND A MATLAB IMPLEMENTATION

| Method | t=0 | t=1 | t=2 | t=3 | CPU time (second per iteration) |
|---|---|---|---|---|---|
| RBF | 44.82 | 79.82 | 88.75 | 91.79 | 2.34 |
| MARS-2 | 44.82 | 60.18 | 61.61 | 61.96 | 1.26 |
| OPT-RF | 44.82 | 72.14 | 79.64 | 80.54 | 1.27 |
| Simple CBIR | 44.82 | - | - | - | 0.90 |

Fig. 5. The 35 test query images chosen from different categories from the Corel database.

MARS-2 and OPT-RF. With this large, heterogeneous image collection, an initial result obtained by the simple CBIR system has less than 50% precision. With the application of the RBF interactive learning, we can improve the performance to more than 90% precision. Due to the limitation in the degrees of


Fig. 6. Top 16 retrieved images obtained by retrieving the "Yacht" query, from the Corel Database, using (a) simple CBIR, (b) RBF, (c) MARS-2, and (d) OPT-RF.

adaptivity, MARS-2 provides the lowest performance gains and converges at about 62% precision. We observe that the learning capability of the RBF is more robust than that of OPT-RF, not only with respect to retrieving accurately, but also learning speed. As evidenced by Table V, results after *one* round of the RBF is similar to results after *three* rounds of the OPT-RF. This quick learning is highly desirable, since the user workload can be minimized. This robustness follows from imposing nonlinear discriminant capability in combination with the positive and negative learning strategies. Notice that OPT-RF requires users to specify weight parameters in the form of a slider bar for learning, whereas RBF automatically evaluates these weight parameters from the feedback images.

In regard to CPU time for the retrievals, the RBF approach is longer, at 2.34 s per iteration for a single query. However, the RBF method gains about 80% precision within only the first iteration, i.e., in only 2.34 s. By contrast, though faster, the OPT-RF needs three iterations to reach this underlined performance, i.e., taking $1.27 \times 3 = 3.81$ seconds. In other words, we see that RBF can reach the best performance within a shorter CPU time than the other methods discussed. This also means that OPT-RF users are required to go through two more rounds of feedback in order to achieve equivalent performance. Furthermore, we can observe that, when subject to three iterations, RBF reaches a 91% precision level that cannot be achieved by any other method.

Typical retrieval sessions are shown in Figs. 6 and 7. Fig. 6 shows retrieval results of the "Yacht" query. Fig. 6(a) shows the 16 best-matched images before applying any feedback, with the query image display in the top-left corner. It was observed that some retrieved images are similar to the query in terms of color composition. In this set, three retrieved images were marked as relevant subject to the ground truth classes. Fig. 6(b) shows the improvement of retrieval after three rounds of RBF interactive learning. This is superior to the results obtained by MARS-2 [cf. 6(c)] and OPT-RF [cf. Fig. 6(d)]. The outstanding performance of the RBF method can also be seen from Figs. 7(a)–(d), showing the retrieval results in answering the "Tiger" query. As evidenced by the results of Figs. 6(b) and 7(b), we observed that nonlinear analysis obtained by RBF can effectively capture high-level concepts in few retrieval sessions.

## VII. Conclusion

In the past, a number of attempts have been made to describe visual contents with "index features" for operating content-based image retrieval. The evidence shows that semantics and user request are more essential than the "index features". This has directed a number of researchers to suggest that such a retrieval problem must be interpreted as human-centered, rather than computer-centered [6], [7]. We have shown in this paper that the user information needs in a visual-seeking environment are well addressed by user-interface methodologies. User interface allows the retrieval system to solve the problem of fuzzy understanding of user's goals and thus aid in the expression of information needs. There are two main points that have been demonstrated by our method: 1) learning-based systems can adjust their strategy in accordance with user input; and 2) user in-

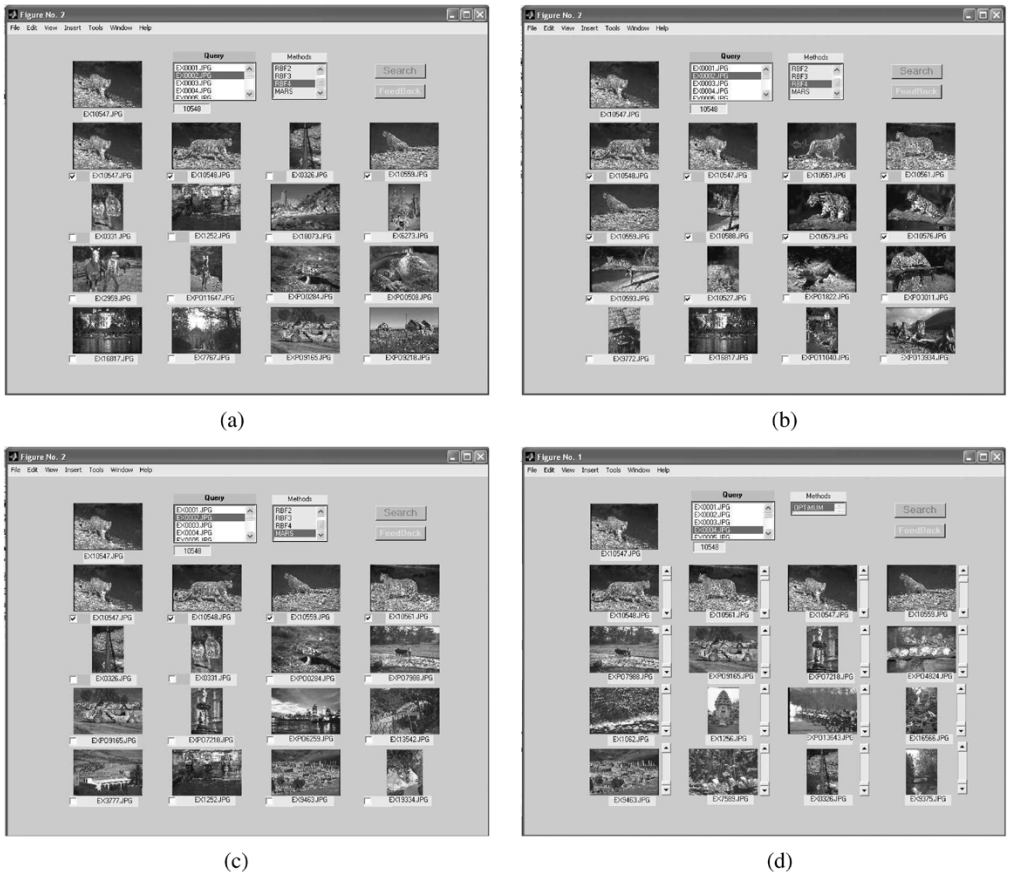


Fig. 7. Top sixteen retrieved images obtained by retrieving the "Tiger" query, from the Corel Database, using (a) simple CBIR, (b) RBF, (c) MARS-2, and (d) OPT-RF.

formation needs are satisfied by a series of selections of information.

The most difficult task in the interactive process is to analyze the role of the users in perceiving image similarity. We have emphasized the importance of 'mapping' human perception onto the image-matching process. Our RBF model incorporates and emphasizes many new features not found in earlier interactive retrieval systems. Many of these features are imparted by *nonlinear* discriminant analysis with a high degree of adaptivity from learning through negative and positive samples. This results in a high performance learning machine that learns effectively and quickly from a small set of feedback data. We have suggested that through a learning-based approach it is possible to relate the behavior of human perception to low-level feature processing in visual retrieval systems. The learning-based approach takes into account the complexities of individual human perception and in fact uses individual user choices to decide relevance. This learning machine combines state-of-the-art retrieval performance, with a very rich set of features, which may help to usher in a new generation of multimedia applications.

## REFERENCES

[1] M. Flickner, H. Sawhney, and W. Niblack, "Query by image and video content: The QBIC system," *IEEE Computer*, vol. 28, pp. 23–31, Sept. 1995.

[2] J. R. Back, F. Charles, A. Gupta, A. Hampapur, B. Horowitz, R. Humphrey, R. Jain, and C.-F. Shu. (2000) Virage Image Search Engine: An Open Framework for Image Management. [Online]. Available: http://www.virage.com

[3] A. Pentland, R. W. Picard, and S. Sclaroff, "Photobook: Content-based manipulation of image databases," *Int. J. Comput. Vis.*, vol. 8, no. 3, pp. 233–254, 1996.

[4] J. R. Smith and S.-F. Chang, "VisualSEEk: A fully automated content-based image query system," in *Proc. ACM Multimedia Conf.*, New York, 1996, pp. 87–98.

[5] W. Y. Ma and B. S. Manjunath, "Netra: A toolbox for navigating large image databases," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, 1997, pp. 568–571.

[6] T. Gevers and W. M. Smeulders, "PicToSeek: Combining color and shape invariant features for image retrieval," *IEEE Trans. Image Processing*, vol. 9, pp. 102–119, Jan. 2000.

[7] A. Celentano and E. D. Sciasicio, "Feature interation and relevance feedback analysis in image similarity evaluation," *J. Electron. Imag.*, vol. 7, no. 2, pp. 308–317, 1998.

[8] Y. Rui, T. S. Hang, S. Mehrotra, and M. Ortega, "A relevance feedback architecture for content-based multimedia information retrieval systems," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997, pp. 82–89.

[9] J. Peng, B. Bhanu, and S. Qing, "Probabilistic feature relevance learning for content-based image retrieval," *Comput. Vis. Image Understand.*, vol. 75, no. 1/2, pp. 150–164, 1999.

[10] G. Ciocca and R. Schettini, "Using a relevance feedback mechanism to improve content-based image retrieval," in *Visual Information and Information Systems*, P. Huijsmans and W. M. Smeulders, Eds. Berlin, Germany: Springer, 1999, pp. 105–114.

[11] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: A power tool for interactive content-based image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 8, no. 5, pp. 644–655, 1998.

[12] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. IEEE Int. Conf. on Image Processing*, Washington, DC, 1997, pp. 815–818.

[13] E. Di Sciascio and M. Mongiello, "DrawSearch: A tool for interactive content-based image retrieval over the net," in *Proc. SPIE*, vol. 3656, 1999, pp. 561–572.

[14] H. Müller, W. Müller, S. Marchand-Maillet, and D. McG Squire, "Strategies for positive and negative relevance feedback in image retrieval," in *Int. Conf. on Pattern Recognition*, Barcelona, Spain, Sept. 2000.
[15] S. Haykin, *Neural Networks: A Comprehensive Foundation*. New York: Macmillan, 1994.
[16] J. Moody and C. J. Darken, "Fast learning in networks of locally-tuned processing units," *Neural Computat.*, vol. 1, no. 2, pp. 281–294, 1989.
[17] Y. Rui and T. S. Huang, "Optimizing learning in image retrieval," in *Proc. of IEEE Int Conf. on Computer Vision and Pattern Recognition (CVPR)*, Hilton Head, SC, June 2000.
[18] P. Muneesawang and L. Guan, "Multiresolution-histogram indexing for wavelet-compressed images and relevant feedback learning for image retrieval," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, Vancouver, BC, Canada, 2000, pp. 526–529.
[19] S. Sclaroff, L. Taycher, and M. L. Cascia, "ImageRover: A content-based image browser for the world wide web," in *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997, pp. 2–9.
[20] R. L. De Valois and K. K. De Valois, *Spatial Vision*. New York, NY: Oxford, 1988.
[21] P. Muneesawang and L. Guan, "A nonlinear RBF model for interactive content-based image retrieval," in *First IEEE Pacific-Rim Conference on Multimedia*, Sydney, Australia, 2000, pp. 188–191.
[22] A. N. Tikhonov, "On solving incorrectly posed problems and method of regularization," in *Neural Networks: A Comprehensive Foundation*, S. Haykin, Ed: Prentice Hall, 1999.
[23] T. Poggio and F. Girosi, "Networks for approximation and learning," *Proc. IEEE*, vol. 78, pp. 1481–1497, 1990.
[24] J. H. Friedman, "Flexible Metric Nearest Neighbor Classification," Dept. Statistics, Stanford Univ., Stanford, CA, Tech. Rep., 1994.
[25] Corel Gallery Magic 65 000 (1999). [Online]. Available: www.corel.com
[26] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1992.
[27] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 8, pp. 837–842, 1996.
[28] G. Salton and M. J. McGill, *Intoduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.
[29] S. J. Cunningham, G. Holmes, J. Littin, R. Beale, and I. H. Witten, "Applying connectionist models to information retrieval," *Blain-Link Computing and Intelligent Information Systems*, 1998.
[30] J. J. Rocchio, "Relevance feedback in information retrieval," in *The SMART Retrieval System Experiments in Automatic Document Processing*, G. Salton, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[31] *MPEG-7 Visual Part of the eXperimentation Model (Version 9.0)*, Jan. 2001. ISO/IEC JTC1/SC29/WG11 N3914.
[32] N. B. Karayiannis, P.-I. Pai, and N. Zervos, "Image compression based on fuzzy algorithms for learning vector quantization and wavelet image decomposition," *IEEE Trans. Image Processing*, vol. 7, pp. 1223–1230, Aug. 1998.
[33] B. S. Manjunath, P. Wu, S. Newsam, and H. D. Shin, "A texture descriptor for browsing and similarity retrieval," *Signal Process. Image Commun.*, vol. 16, no. 1–2, pp. 33–43, Sept. 2000.
[34] P. Wu, B. S. Manjunath, S. D. Newsam, and H. D. Shin, "A texture descriptor for image retrieval and browsing," in *Computer Vision and Pattern Recognition Workshop*, Fort Collins, CO, June 1999.
[35] M. Safar, C. Shahabi, and X. Sun, "Image retrieval by shape: A comparative study," in *IEEE International Conference on Multimedia and Expo (I)*, 2000, pp. 141–144.

**Paisarn Muneesawang** (M'02) received the B.Eng. degree from the Department of Telecommunication Engineering, Mahanakorn University of Technology, Thailand, in 1996. In 1998, he received the Royal Thai Government Scholarship for studying postgraduate programs in Australia. In 1999, he received the M.Eng.Sc. degree in electrical engineering for his work on application of multiresolution analysis methods to image processing, from the University of New South Wales, Australia. From 1999 to 2002, he was a Research Student in the Signal and Multimedia Processing Laboratory in the School of Electrical and Information Engineering, University of Sydney, Australia, where he submitted the Ph.D. thesis on the application of computational intelligence techniques to the problem of multimedia processing, particularly the problem of image and video indexing/retrieval and the development of a human–computer interaction model for multimedia retrieval.

He has been a Lecturer at Naresuan University, Thailand, since 1996. From March to August 1997, he was a Researcher in the Optical Fiber Laboratory, School of Electrical Engineering, University of New South Wales. He is currently a Lecturer in the Department of Electrical and Computer Engineering, Naresuan University, Thailand. His research interests include video, image, multimedia signal processing, computer vision, and neural network.

**Ling Guan** (S'88–M'90–SM'96) received the Ph.D. degree in electrical engineering from the University of British Columbia, Vancouver, BC, Canada, in 1989.

From 1989 to 1992, he was a Research Engineer with Array Systems Computing, Inc., Toronto, ON, Canada, in machine vision and signal processing.

From October 1992 to April 2001, he was a Faculty Member with the University of Sydney, Sydney, Australia. Since May 2001, he has been a Professor in Department of Electrical and Computer Engineering, Ryerson Polytechnic University, Toronto, ON, Canada, where he was subsequently appointed to the position of Canada Research Chair (Tier I). In 1994, he was a Visiting Fellow at British Telecom. In 1999, he was a Visiting Professorial Fellow at the Tokyo Institute of Technology, Tokyo, Japan. In 2000, he was on sabbatical leave at Princeton University, Princeton, NJ. His research interests include multimedia processing and systems, optimal information search engine, signal processing for wireless multimedia communications, computational intelligence and machine learning, adaptive image and signal processing. He has published more than 150 technical articles, and is the editor/author of two books, *Multimedia Image and Video Processing* and *Adaptive Image Processing: A Computational Intelligence Perspective*.

Dr. Guan is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, and several other international journals. In 1999, he co-guest-edited the special issues on computational intelligence for the PROCEEDINGS OF THE IEEE. He also serves on the editorial board of CRC Press' *Book Series on Image Processing*. He has been involved in organizing numerous international conferences. He played the leading role in the inauguration of the IEEE Pacific-Rim Conference on Multimedia, and served as the Founding General Chair in Sydney 2000. He is a member of IAPR and SPIE. He is currently serving on IEEE Signal Processing Society Technical Committee on Multimedia Signal Processing.