Chapter 2

INTERACTIVE CONTENT-BASED IMAGE RETRIEVAL

1. Introduction

The central problems regarding the retrieval task are concerned with "interpreting" the contents of the images in a collection and ranking these 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 needs. 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.

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 [3–6],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 main activities in this approach consist in analyzing a user's goals from feedback information on the desired images, and adjusting the search strategy accordingly. Here, the user manages the retrieval system, via the interface, through the 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, since motivations and behaviors are more difficult to measure and characterize. Recently, 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 [6–8] for examples). This idea can be further generalized by incorporating limited adaptivity in the form of a relevance feedback scheme [3–5, 9–11]. Here, weighting is modified according to the user's preference. However, from the user's viewpoint, the limited degree of adaptivity offered, and the restriction of the distance measure to a quadratic form, is not adequate for modeling perceptual difference.

Application of Nonlinear Human-Controled Interactive (HCI) Retrieval

In this chapter, a *nonlinear* approach is presented to address some of these problems to simulate human perception in human-controlled interactive CBR (HCI-CBR). This effectively bridges the gap between the low-level features used in retrieval and the high-level semantics in human perception. The traditional relevance feedback is replaced by a specialized radial-basis function (RBF) network [12, 13] for learning the user's 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 RBFs in the network. This concept is adaptively re-defined in accordance with different user's preferences and different types of images, instead of relying on any pre-conceived 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 RBF centers and widths. This allows a more accurate modeling of the notion of similarity from the user's viewpoint.

The new HCI-CBR is applied to two image database applications. The first domain application is texture retrieval. In this domain, content-based image retrieval is very useful for effective querying using texture patterns that represent a region of interest in a large collection of satellite air photos, as demonstrated in [24, 29, 30]. In the second domain, the HCI-CBR is integrated with an interactive search engine, the Interactive-based Analysis and Retrieval of Multimedia system (iARM) [144], to support image searching tools in large image collections over the internet.

In this chapter, the content-based image retrieval (CBIR) techniques also applied to a computer aided referral (CAR) system. This system implements CBIR process to assist operators for the detection of underwater mine-like objects in side-scan sonar images.

2. Interactive Framework

The most important part in the interactive process is to analyze the role of the user in perceiving image similarity according to preferred image selections. To perform this analysis, a nonlinear model is employed to establish the link between human perception and distance calculation.

In general, learning systems implement a mapping $f_s : \mathcal{R}^P \rightarrow \mathcal{R}$ which is given by:

$$y_s = f_s(\mathbf{x}) \tag{2.1}$$

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

As many attempts to perform similarity analyzing have focused on linear models, an introduction to linear-based approaches and their limitations are in order. These can be organized into two categories: (1) an approach based on a query reformulation model ([4, 9, 11]); and (2) an approach based on an adaptive metric model ([7, 8, 14, 70]).

Query Reformulation Method

Among the early attempts in the interactive CBIR systems, MARS-1 (Multimedia Analysis and Retrieval System, version 1) [5, 9] implemented the mapping in the form of the query reformulation model (originally proposed by Salton [25] for text retrieval),

$$y_s = f_{\text{cosine}}(\mathbf{x}, \mathbf{x}_{\hat{q}}) \tag{2.2}$$

$$\mathbf{x}_{\hat{q}} = \alpha \mathbf{x}_q + \beta \left(\underset{l_i=1}{mean}\{\mathbf{x}_i\} \right) - \varepsilon \left(\underset{l_i=0}{mean}\{\mathbf{x}_i\} \right) \tag{2.3}$$

where f_{cosine} denotes the cosine measure; $\mathbf{x}_q$ denotes the original query vector; $\mathbf{x}_{\hat{q}}$ denotes the modified query vector; and $(\alpha, \beta, \varepsilon)$ are suitable parameters. 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) [25, 27] and many image retrieval systems [4, 11]. A chief disadvantage of this integration model is the requirement of an indexing structure to follow term-weighting models used in text retrievals for greater effectiveness. Specifically, the model works on the assumption that the query index terms are sparse and are usually of a binary vector representation. However, in image content-based retrieval, vectors are mostly real vectors. In order to overcome this problem, Muller *et al* [11] utilize more than 80,000 feature variables to characterize each image. This method, however, increases computational complexity.

Adaptive Similarity Function

In its later form, weight distance is a common strategy for obtaining the mapping function. This is the case in [6, 10, 64–67, 70, 75], and in the MARS-2 (Multimedia Analysis and Retrieval, version 2) system [8]. In general, the similarity function may be described as:

$$f_s(\mathbf{x}, \mathbf{x}_q) = \sum_{i=1}^{P} h(d_i) \tag{2.4}$$

$$= (\mathbf{x} - \mathbf{x}_q)^T W (\mathbf{x} - \mathbf{x}_q) \tag{2.5}$$

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

$$W = diag[w_1, w_2, ..., w_P] \tag{2.6}$$

The weight parameters $w_i, i = 1, ..., P$ are called *relevance weights*, and are 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 [8, 65, 66] or a probabilistic feature relevance method [6].

In addition, different types of basis function have been used [68–70]. Sclaroff *et al* [70] introduced an algorithm for selecting appropriate Minkowski distance metrics according to a minimum distance within the relevant class. This is based on the assumption that metrics are optimal for different classes of query images. With a similar assumption, Bhanu *et al* [67] developed the algorithm for selecting appropriate metrics using reinforcement learning. Choi *et al* [69] proposed an improved similarity function by taking into account the interdependencies between feature elements. This method applies a fuzzy measure over the set of features extracted from feedback samples, and uses the Choquet Integral as the similarity function.

Query and metric Adaptations

In [14, 71–74], an adaptive system combines a query reformulation model with the similarity function (e.g. Eqs.(2.3)-(2.4)) in order to speed up convergence. Some of these works also implemented a query shifting model, using different techniques, such as linear discrimination analysis [74], and a probabilistic distribution analysis on the positive and negative samples [72].

Rui *et al* [14] have derived an optimum solution for the similarity function (Eq.(2.4)), and the query shifting model. This method is referred to as an optimal learning relevance feedback (OPT-RF) method. Using Lagrange multipliers, an optimum solution for a query model is the weighted average of the training samples:

$$\mathbf{x}_{\hat{q}} = \frac{\mathbf{X}^T \mathbf{v}}{\sum_{i=1}^{N} v_i} \tag{2.7}$$

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

$$W = \begin{cases} (\det(C))^{\frac{1}{K}} C^{-1} & \det(C) \neq 0 \\ diag(\frac{1}{C_{11}}, \frac{1}{C_{22}}, ..., \frac{1}{C_{PP}}) & \text{otherwise} \end{cases} \tag{2.8}$$

where C denotes the weighted covariance matrix,

$$C_{rs} = \frac{\sum_{i=1}^{N} v_i (x_{ir} - x_{\hat{q}r})(x_{is} - x_{\hat{q}s})}{\sum_{i=1}^{N} v_i}, \quad r, s = 1, ..., P \tag{2.9}$$

OPT-RF intelligently switches W between a full matrix and a diagonal matrix, to overcome possible singularity issues 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 when only a *small* size of training samples is preferred, i.e., $N < P$.

Ashwin *et al* [71] implement an improved OPT-RF method. Here, the negative samples are used to modify the weight parameters computed, as in Eqs.(2.8)-(2.9), so that the ellipsoids represented by the similarity metric (Eq.(2.5)) better capture more positive samples while excluding negative ones.

Nonlinear Model-based Relevance Feedback

The methods outlined above are referred to as linear-based learning that restricts the mapping function to quadratic form, which cannot cope with a complex decision boundary. Although these learning methods provide a mathematical framework for evaluating image similarity, they are not adequate for the nonlinear nature of human perception. For instance, one-dimension distance mapping $h(d_i)$ in Eq.(2.4) takes the following form:

$$h(d_i) = w_i d_i^2 \tag{2.10}$$

It has no nonlinear capacity, such as:

$$\frac{\partial f_s(\mathbf{x})}{\partial d_i} = 2 w_i d_i \tag{2.11}$$

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 magnitudes of their distances. In comparison, the assumption for nonlinear approach is that the same portions of the distances do not always give the same degree of similarity when judged by humans [17].

The visual section of the human brain uses a nonlinear processing system for tasks such as pattern recognition and classification [12]. The application of nonlinear criterion is therefore suitable in performing a simulation task. The current work is different from MARS-2 and OPT-RF in two aspects. First, a nonlinear kernel is applied for the evaluation of image similarity. Second, both positive and negative feedback strategies are utilized for greater effectiveness of the learning capability. By embedding these two properties, the current retrieval system shows a high performance in learning with small user feedback samples, and convergence occurs quickly (results shown in Section 5-6).

The Basic Model

To simulate human perception, a radial basis function (RBF) network [12, 13] is employed 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 $\mathcal{X}$ to evaluate the degree of similarity (according to a given query), by a combination of activation functions associated as a nonlinear transformation.

The input-output mapping function, $f(\mathbf{x})$, is performed on the basis of a method called *regularization* [19]. 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 [20]. Based on the regularization method, a one-dimensional Gaussian-shaped radial basis function is utilized to form a basic model:

$$G(x) = \exp\left(-\frac{(x-z)^2}{2\sigma^2}\right) \tag{2.12}$$

where z denotes the center of the function and σ denotes its width. The activity of function $G(x)$ is to perform a Gaussian transformation of the distance $d = |x - z|$, 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 Guassian RBF is expanded through both its center and width, yielding different RBFs which are then formed as an RBF network. Its expansion is implemented via interactive learning, where the expanded RBFs can optimize weighting, to capture human perception similarity as discussed in the following section.

3. Radial Basis Function (RBF)-Based Relevance Feedback (RF)

Radial basis function (RBF) networks possess an excellent nonlinear approximation capability [12, 13]. This property is utilized 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:

$$f(\mathbf{x}) = \sum_{j=1}^{N} w_j G(\mathbf{x}, \mathbf{z}_j) \tag{2.13}$$

$$= \sum_{j=1}^{N} w_j \exp\left(-\frac{1}{2\sigma_j^2}\sum_{i=1}^{P}(x_i - z_{ji})^2\right) \tag{2.14}$$

where $\mathbf{z}_j \in \mathcal{R}^P$ denotes the center of the function $G(\mathbf{x}, \mathbf{z}_j)$, σ_j denotes its width, and $\mathbf{x} \in \mathcal{R}^P$ denotes the input vector. There are N Gaussian units 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 correspondance between the training input samples and the function $G(\mathbf{x}, \mathbf{z}_j)$, by which each training sample is associated with the center $\mathbf{z}_j, j \in \{1, N\}$.

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 functions in mind, a one-dimensional Gaussian-shaped RBF is associated with each component of the feature vector, as follows:

$$f(\mathbf{x}) = \sum_{i=1}^{P} G_i(x_i, z_i) \tag{2.15}$$

$$= \sum_{i=1}^{P} \exp\left(-\frac{(x_i - z_i)^2}{2\sigma_i^2}\right) \tag{2.16}$$

where $\mathbf{z} = [z_1, ..., z_i, ..., z_P]^T$ is the adjustable query position or the center of the RBF function, $\sigma_i, i = 1, ..., P$ are the tuning parameters in the form of RBF widths, and $\mathbf{x} = [x_1, ..., x_i, ..., x_P]^T$ 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 simulation results in Section 5, the new single unit RBF network is effective in learning and quickly converges for one-class-relevance classification using small volume of training sets.

Expansion of RBFs

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

$$\{\sigma_i, z_i\}, i = 1, ..., P, \tag{2.17}$$

to form a set of P basis functions,

$$\{G_1(\sigma_1; z_1), G_2(\sigma_2; z_2), ..., G_P(\sigma_P; z_P)\}. \tag{2.18}$$

These parameters are estimated and updated via learning algorithms. The first assumption behind the 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 [21, 6]. However, in this case, the weighting process is controlled by an expanded set of tuning parameters, $\sigma_i, i = 1, ..., P$, which reflects the relevance of individual features. If a feature is highly relevant, the value of σ_i should be small to allow greater sensitivity to any change of the distance $d_i = |x_i - z_i|$. In contrast, a large value of σ_i is assigned to the non-relevant 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 σ according to this criterion will be discussed in Section 4.0.

The second assumption concerns 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, the associated feature vector may not be in a position close enough to those stored vectors associated with other relevant images. This initial query may form a decision region that contains only a local cluster of the desired images in the database. The goal here, then, 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, $z_i, i = 1, ..., P$, as will be described in the following section.

4. Learning and Characterization

The learning algorithms 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_c 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 in order 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.

Center selection

Given a set of images, the human user may easily distinguish the relevant and nonrelevant images according to their own information needs (Fig. 2.1(a)). In contrast, a computer interprets relevance as the distance between low-level image features (Fig. 2.1(b)), which could be very different from that shown in Fig. 2.1(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, the low-level query vector is modified via the learning algorithm. This aims at optimizing the current search. The expected effect is that the new query will move towards the relevant items (corresponding to the desired images) and away from the non-relevant ones, whereas the user's information need remains the same throughout the query modifying process. In the following discussion, the basic optimization procedure known as learning vector quantization (LVQ) [12] is firstly described. Then, a modified LVQ is presented to obtain a proper choice for the RBF center associated with the new query vector.

LVQ

LVQ [12] is a supervised learning technique used to optimize vector structures in a *code book* for the purpose of data compression [23]. 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 $\mathbf{x}$.

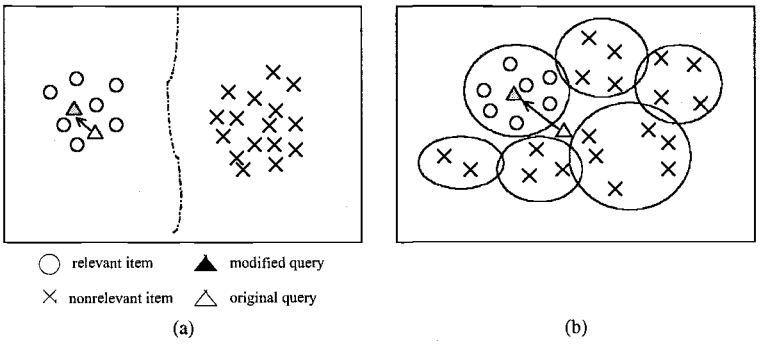


Figure 2.1. Query modification; (a) relevance judgment based on human vision; (b) relevance clustering in the feature space. In (a), given an image collection set, the human user may easily distinguish the relevant images from the high-level semantics according to his/her own understanding and expression of require information. 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.

The modification of the Voronoi vectors is usually carried out by an iterative process, where $n = 0, 1, 2, ..., n_{\max -1}$ 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 Voronoi vector $\mathbf{z}_c(n)$ is identified by the condition:

$$c = \arg\min_j \{||\mathbf{x}_i - \mathbf{z}_j||\} \tag{2.19}$$

Let $\ell_{\mathbf{z}_c}$ denote the class label 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*)

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) + \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)] \tag{2.20}$$

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

$$\mathbf{z}_c(n+1) = \mathbf{z}_c(n) - \alpha_n[\mathbf{x}_i(n) - \mathbf{z}_c(n)] \tag{2.21}$$

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, α_n, decreases monotonically with the number of iterations and where $0 < \alpha_n < 1$. After several passes through the training data, the Voronoi vectors typically converge, and the training process is completed.

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

A Modified LVQ

In an interactive retrieval session, it is desirable to a reduce the processing time to minimum without affecting the overall performance. So, the LVQ method can be considered for query modification, without 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, the database feature space can be clustered 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. The objective, 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:

$$\mathcal{T}_{(t)} = (\mathbf{x}_i, l_i),\ i = 1, \ldots, N \tag{2.22}$$

$$= \{\mathbf{x}'_m,\ |\ l_m = 1\} \cup \{\mathbf{x}''_k,\ |\ l_k = 0\} \tag{2.23}$$

$$m = 1, ..., M \tag{2.24}$$

$$k = 1, ..., K \tag{2.25}$$

where $\mathbf{x}_i \in \mathcal{R}^P$ is a feature vector; $l_i \in \{0, 1\}$ is a class label; $\mathbf{x}'$ and $\mathbf{x}''$ are the positive and negative sample, respectively. The set of vectors in Eq.(2.22) 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, it is observed 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 the previous discussion, after the training process is completed 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. Combining these ideas, a modified LVQ algorithm is now obtained, to adjust the query vector $\mathbf{z}_q(t)$, by approximating the *modified* Voronoi vector $\tilde{\mathbf{z}}_q$ upon convergence:

$$\mathbf{z}_q(t+1) = \mathbf{z}_q(t) + \alpha_R(\overline{\mathbf{x}}' - \mathbf{z}_q(t)) - \alpha_N(\overline{\mathbf{x}}'' - \mathbf{z}_q(t)) \quad (2.26)$$

$$\overline{\mathbf{x}}' = \frac{1}{M}\sum_{m=1}^{M} \mathbf{x}'_m \quad (2.27)$$

$$\overline{\mathbf{x}}'' = \frac{1}{K}\sum_{k=1}^{K} \mathbf{x}''_k \quad (2.28)$$

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; $\mathbf{x}''_k = [x''_{k1}, \ldots, x''_{ki}, \ldots, x''_{kP}]^T$ is the k-th feature vector of nonrelevant images; α_R and α_N are suitable positive constants; and M and K are, respectively, the number of relevant and nonrelevant images in the training set. The application of the query modification in Eq.(2.26) is to allow the new query, $\mathbf{z}_q(t+1)$, to move towards the new region populated by the relevant images as well as to move away from those regions populated by non-relevant images.

Eq.(2.26) is illustrated in Fig. 2.2. Let the centers of the relevant image set and nonrelevant image set in the training data, be R and N, respectively. Also, let $z_q(t) = z_c$. As shown in Fig. 2.2, the effect of the second term on the right hand side of Eq.(2.26) is to allow the new query to move towards 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. The is because the set of relevant images is usually tightly clustered due to the similarities among its member images, and thus satisfies the modified query with little ambiguity. This is illustrated in Fig 2.3(a). 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, $\alpha_R > \alpha_N$ is chosen for Eq.(2.26) to allow a more definite movement toward the set of relevant images, while permitting slight movement away from the non-relevant regions.

The current approach works well when the sets of relevant and non-relevant images are well-separated, as in Fig. 2.3(a). In practice, the set of non-relevant

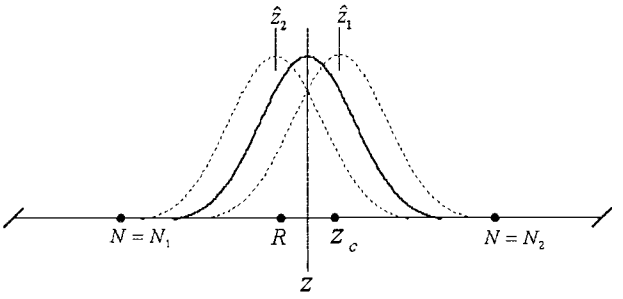


Figure 2.2. Illustration of query modification.

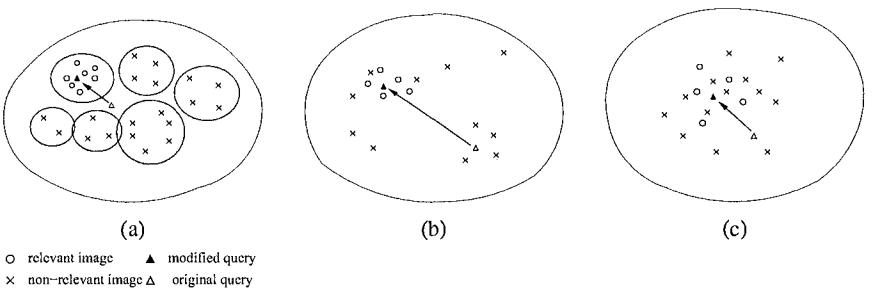


Figure 2.3. Query modification in the feature space; (a) ideal configuration; (b) favorable configuration; (c) unfavorable configuration.

images usually covers a wider region of the space, as shown in Figs. 2.3(b) and (c). The effectiveness of the current approach will thus depend on the exact distribution of the non-relevant images in the space. Fig. 2.3(b) illustrates a particular distribution which is favorable to the current approach, while the case illustrated in Fig. 2.3(c) may compromise performance.

Model 2: In order to provide a simpler procedure and a direct movement of the new query towards the relevant set, Eq.(2.26) is reduced to

$$\mathbf{z}_q(t+1) \;=\; \overline{\mathbf{x}}' - \alpha_N(\overline{\mathbf{x}}'' - \mathbf{z}_q(t)) \tag{2.29}$$

The first and the second terms in the right hand side of Eq.(2.26) are replaced by $\overline{\mathbf{x}}'$ (centroid of the relevant vectors). Since the relevant image group indicates the user's preference, the presentation of $\overline{\mathbf{x}}'$ for the new query will give a reasonable representation of the desired image. In particular, the mean value,

$\overline{x}'_i = (1/M)\sum_{m=1}^{M} x'_{mi}$, is a statistical measure providing a good representation of the i-th feature component since this is the value which minimizes the average distance $(1/M)\sum_{m=1}^{M}(x'_{mi} - \overline{x}'_i)^2$. Further, the exclusion of the parameter α_R from Eq.(2.26) permits greater flexibility, since only one procedural parameter is necessary for the final fine tuning of a new query.

Finally, a comparison of the optimum query model, based on Eq.(2.7) to Eq.(2.26) and Eq.(2.29), shows the following: Eq.(2.7) is optimal on a criterion of minimum distance to relevant retrievals, while Eq.(2.26) and Eq.(2.29) are multiple criterion optimizations, minimizing distance to relevant samples and maximizing distance to nonrelevant samples.

Selection of RBF width

As observed from the previous discussions, the nonlinear transformation associated with the output unit(s) of the Gaussian-shaped RBF are adjusted in accordance with different user's preferences 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 Eq.(2.22). 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 $\mathbf{R}$:

$$\begin{aligned} \mathbf{R} &= [\mathbf{x}'_1, \ldots, \mathbf{x}'_m, \ldots, \mathbf{x}'_M]^T \\ &= [x'_{mi}] \quad m = 1, \ldots, M,\ i = 1, \ldots, P \end{aligned} \tag{2.30}$$

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 previous discussion, the tuning parameter σ_i should reflect the relevance of individual features. It was proposed, in [6, 21], 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.

$$\sigma_i = \eta \max_m |x'_{mi} - z_i| \tag{2.31}$$

The factor η guarantees a reasonably large output $f(\mathbf{x})$ for the Gaussian RBF unit, which indicates the degree of similarity, e.g., η=3.

The second criterion is also considered for estimating the tuning parameters. This is obtained by nonlinear weighting of the sample variance in the relevant set as follows:

$$\sigma_i = \exp(\beta \cdot \mathrm{STD}_i) \tag{2.32}$$

$$\mathrm{STD}_i = \left(\frac{1}{M-1} \sum_{m=1}^{M} (x'_{mi} - \bar{x}'_i)^2 \right)^{\frac{1}{2}} \tag{2.33}$$

where 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 β can be chosen to maximize or minimize the influence of STD_i on σ_i. For example, when β is large, a change in STD_i will be exponentially reflected in σ_i.

As a result, Eqs.(2.31)-(2.32) provide a small value of σ_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 $d_i = |x_i - z_i|$. In contrast, a high value of σ_i is assigned to the non-relevant features, so that the corresponding vector component can be disregarded when determining the similarity.

5. Application to Texture Image Retrieval

These experiments 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 6 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, the new RBF algorithm is evaluated and compared to other HCI-CBR systems, using these factors.

Comparison Method

The RBF's retrieval performance was compared to the MARS-1, which was developed early in the texture retrieval domain [9, 119]. The retrieval strategy in MARS-1 has also been extended and used in other works, such as [3, 11]. The applications compared are as follows:

1 The radial basis function (RBF) methods: the RBF1 method uses model 1 for determining the RBF center (Eq.(2.26)), and Eq.(2.32) for the RBF width. The RBF2 method uses model 2 for determining the RBF center (Eq.(2.29)) and Eq.(2.31) for learning RBF width.

2 The relevance feedback method (RFM) is described in the MARS-1 system [9, 119], is employed by the PicToSeek system [3], and is also used in [4, 10, 11].

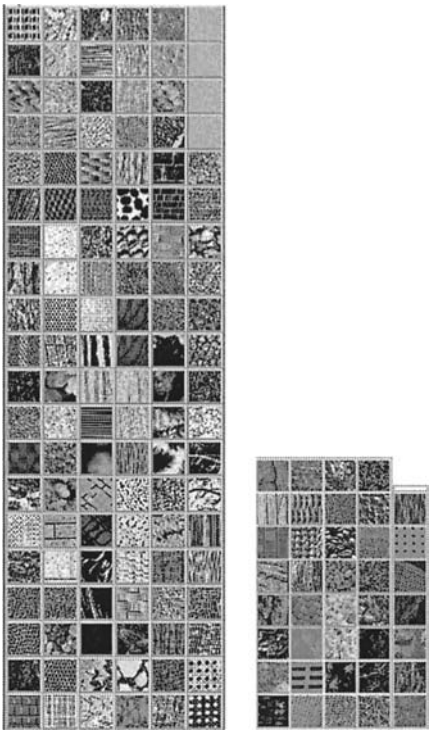
3 Method 3: simple CBIR uses a non-interactive retrieval method, which corresponds to the first iteration of interactive search. This method employs different types of similarity functions, including weighted distance (as in Eq.(2.35) below), cosine distance, and the histogram intersection, corresponding to the first iteration found in RBF, MARS-1, and PicToSeek.

Databases and Ground Truth Classes

The databases and the corresponding ground truth data are generated in the same ways as in previous works [6, 9, 64, 65, 67], as well as in MPEG-7 Texture and Color Core Experiments [88, 115]. Performance evaluations of retrieval were carried out using two standard texture databases: (1) the MIT texture collections, and (2) the Brodatz database [24]. In the first collection, the original test images were obtained from MIT Media Laboratories [116]. There were 39 texture images from different classes manually established through a process of visual inspection. The large image, 512×512 pixels in size, was divided into 16 non-overlapping sub-images, 128×128 in size, creating a database of 624 texture images (shown in Fig. 2.4). In the second database, texture images and a feature set were created by Ma and Manjunath [24] at the University of California at Santa Barbara (UCSB). The Brodatz database contains 1,856 patterns obtained from 116 different texture classes. Each class contains 16 similar patterns. Fig. 2.4 shows all the texture classes, which include both homogeneous and non-homogeneous textures. The strategy used to obtained this database is also based on tiling a large image into smaller sub-images. These can be observed in Fig. 2.5.

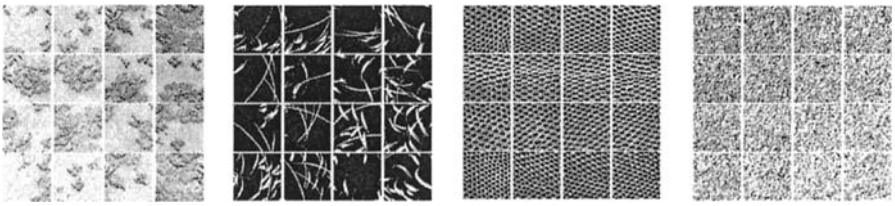
Homogeneous Texture Descriptor (HTD)

The HTD is one of the MPEG-7 texture descriptors described in [88]. HTD characterizes the homogeneous texture pattern based on computing the local spatial-frequency statistics of the texture. Each texture in the database is described by a 48-dimensional HTD vector, which is constructed as follows:



(a) Brodatz database (b) MIT database

Figure 2.4. (a) 116 texture image classes in Brodatz database; (b) 39 texture image classes in MIT database.



(a) (b) (c) (d)

Figure 2.5. (a)-(b) Some examples of non-homogeneous texture; (c)-(d) some examples of homogeneous texture. In each case, the large image is partitioned into 16 sub-images.

firstly, the Gabor wavelet transform is applied to the image, where the set of basis functions consists of Gabor wavelets spanning four scales and size orientations. The mean and standard deviation of the transform coefficients are then

used to form the feature vector. The HTD is described by

$$\mathrm{TD} = [e_1, e_2, \ldots, e_{24}, d_1, d_2, \ldots, d_{24}]^T \tag{2.34}$$

where e_i and d_i represent the mean and standard deviation of the ith feature channel. Since the dynamic range of each feature component is different, a suitable similarity measure for this feature is computed by the following distance measure [24, 88]:

$$d(\mathrm{TD}_{\mathrm{qyery}}, \mathrm{TD}_{\mathrm{Database}}) = \sum_k \left| \frac{\mathrm{TD}_{\mathrm{query}}(k) - \mathrm{TD}_{\mathrm{Database}}(k)}{\alpha(k)} \right| \tag{2.35}$$

where $\alpha(k)$ is the standard deviation of $\mathrm{TD}_{\mathrm{Database}}(k)$ for a given database.

Summary of Comparison

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, all the top 16 retrievals ideally are from the same classes. Similarly, a total of 116 images, one from each class, were selected as the query images from the Brodatz database. In the two databases, performance was measured in terms of retrieval rate (RR), which is defined by [88]:

$$\mathrm{RR}(q) = \frac{\mathrm{NF}(a, q)}{\mathrm{NG}(q)} \in [0, 1] \tag{2.36}$$

where $\mathrm{NG}(q)$ denotes the size of the ground truth set for a query q; and $\mathrm{NF}(a, q)$ denotes the number of ground truth images found within the first $a = 16$ retrievals. For the whole set of NQ queries, the average retrieval rate (AVR) is given by:

$$\mathrm{AVR} = \frac{1}{\mathrm{NQ}} \sum_{q=1}^{\mathrm{NQ}} \mathrm{RR}(q) \tag{2.37}$$

where $\mathrm{NQ} = 39$ and 116 for MIT database and Brodatze databases, respectively.

Summary of Retrieval on MIT Database

The average retrieval rate of the 39 query images is summarized in Table 2.1, where t denotes the number of iterations. The following observations are based on the results.

Method	t=0	t=1	t=2	t=3	Parameters
RBF1	74.36	90.06	92.95	93.59	α_R =1.4, α_N = 0.4, β = 2.6
RBF2	74.36	88.62	91.67	92.79	α_N = 0.65
MARS-1	64.26	77.73	79.97	80.13	(α, γ, ε) = (1, 5, 0.5)

Table 2.1. Average retrieval rate (%) for the 39 query images in MIT database, using Gabor texture feature representation.

First, for all methods, the performance with interactive learning after 3 iterations (t=3) was substantially better than non-interactive cases ($t = 0$). The improvements are quite striking. *Second*, after 3 rounds of interactive learning, the 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 even after 3 iterations (80.13%). *Third*, for all three interactive methods, convergence is achieved within a few iterations.

Fig. 2.6 shows two retrieval sessions performed by RBF1 in comparison with MARS-1. It clearly illustrates the superiority of the nonlinear method. It was 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 more effectively learn and capture user input on image similarity.

Retrieval on the Brodatz Database

Fig. 2.7(a) summarizes the experimental results obtained from the Brodatz database. It shows the average retrieval accuracy of the different methods for the 116 query images, each of which was selected from a different class. It can be seen that all interactive methods demonstrate significant performance improvement across the task. The final results, after learning, show that RBF1 gave the best performance at 90.5% of correct retrieval, followed by RBF2 (87.6%), with MARS-1 (78.5%) a distant third. It was observed earlier that the characteristics of the 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.

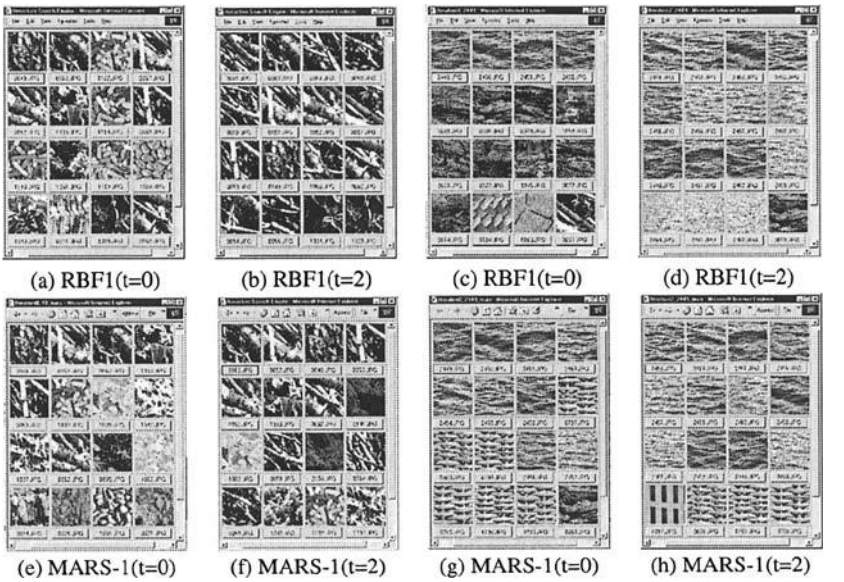


Figure 2.6. Pattern retrieval results before and after learning similarity, using MIT database. Results show a comparison between RBF1 and MARS-1 using Gabor wavelet features; (a), (b), (e), and (f) show retrieval results in the answer to the query 'Bark.0003'; (c), (d), (g), and (h) show retrieval results in the answer to the query 'Water.0000'. In each case, the images are ordered according to decreasing similarity among the 16 best matches, from left to right and top to bottom.

System performance varied greatly depending on the nature of the query. Some queries were easy (i.e., retrieval rate at t=0 is more than 87.5%). Here, the relevant sets were similarly acquired by every method, and the performance after interactive learning was often perfect. By contrast, with a more difficult query, the relevant sets varied greatly in size and composition. In such cases, the effect of interactive learning fluctuated more dramatically within the different methods. To compare the retrieval performances more subjectively, the average retrieval rates were re-calculated excluding 26 query images with a retrieval rate of 100% at t=0. This result is shown in Fig. 2.7(b). Consequently, it was observed that the highest rate was at 87.8% (RBF1), an improvement of 21.7% from 66.1%.

Fig. 2.8 illustrates retrieval examples with and without learning similarity. It shows some of the difficult patterns analyzed, which clearly illustrates the superiority of the RBF method.

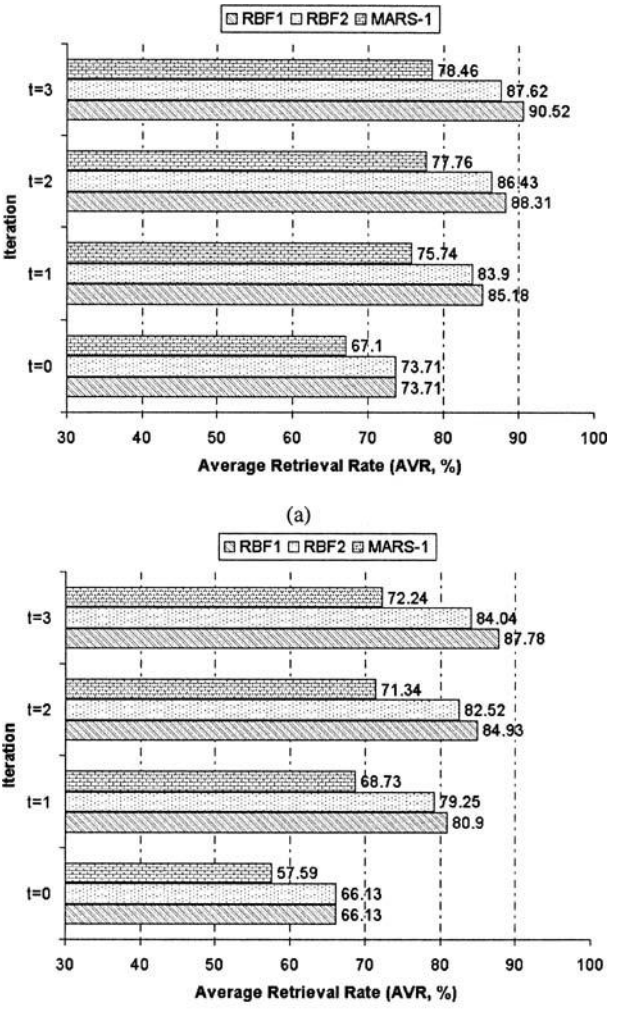


Figure 2.7. Average Retrieval Rate, AVR (%) obtained by retrieving 116 query images in the Brodatz database, using Gabor wavelet representation: (a) AVR of all 116 queries, (b) AVR of (a) excluding 26 query images that had a rate of 100% at t=0.

6. Application to Digital Photograph Collection

In this section, the interactive system is applied to photograph collection from the Corel Gallery 65000 product [22]. The database contains 40000 real-life

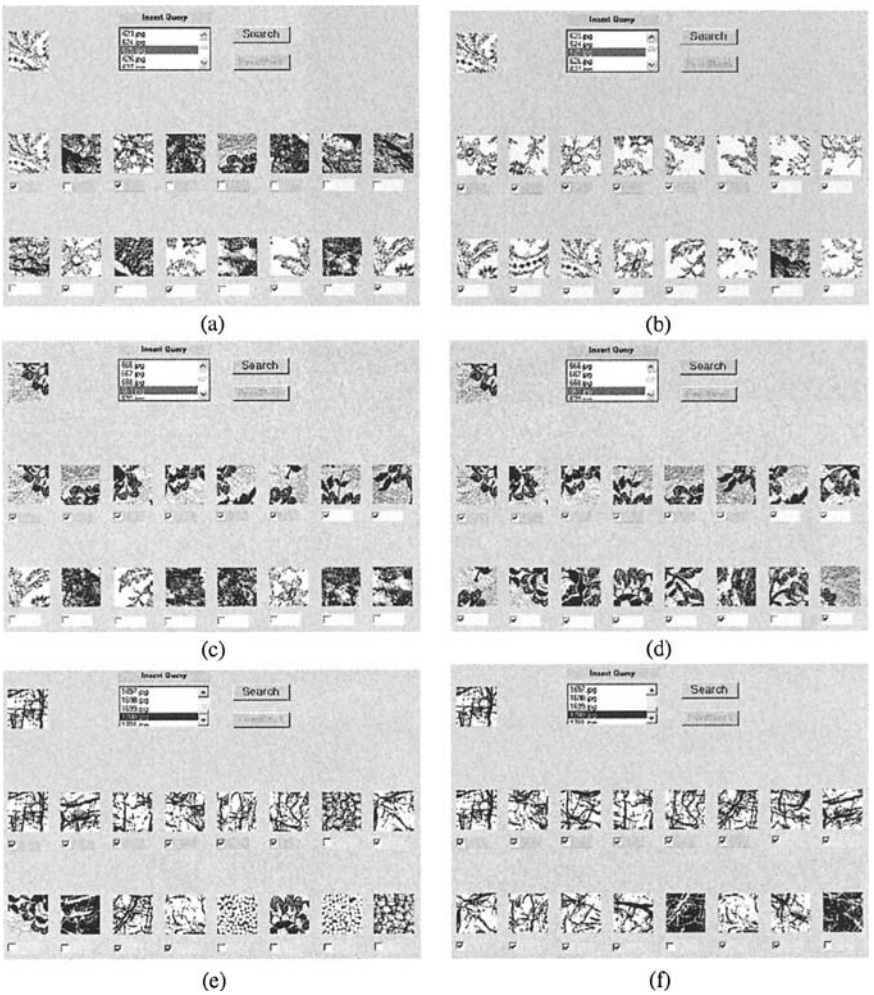


Figure 2.8. Top sixteen retrievals obtained by retrieving textures D625, D669, D1700, and D240 from Brodatz database, using RBF1. Image on the left, (a), (c), and (e) show results before learning, and on the right, (b), (d), and (f), show results after learning.

photographs, in two groups, each of which is either 384 × 256 or 256 × 384 pixels in size (shown in Fig. 2.9). It is organized into 400 categories by Corel professionals. These categories were used as a ground truth in this evaluation. For indexing purposes, each image is characterized by visual descriptors using multiple types of features, $F = \{F_{color}, F_{texture}, F_{shape}\}$ where the representations are color histogram [98] and color moments for color descriptors;

GW transform for texture descriptors [24]; and Fourier descriptor for shape descriptors [31]. The algorithms for obtaining these descriptors are summarized in Table 2.2. Note that the resulting feature database, which is a matrix of size $M \times D$, ($M = 40000$ and $N = 114$), was scaled by feature mean values and standard deviations to remove unequal dynamic ranges of each feature variable.

The following simulations shows performance comparisons between the nonlinear RBF method, MARS-2 [8] and OPT-RF [14] systems (described in Section 2). MARS-2 is relatively newer than MARS-1, and has been intensively tested on the large Corel image collection in [8]. This has become a popular benchmark for image retrieval. In [14], OPT-RF has recently proven to be the optimization framework, more so than in previous studies on interactive CBIR systems. The major difference between these two systems is that the learning algorithm in OPT-RF has both an optimum query and a switching option of the weight matrix W (cf. Eq.(2.8)) between a full matrix and a diagonal matrix. Particularly in this practical application, since the feature dimensions are very high ($D = 114$), OPT-RF was implemented 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 along a slider bar (cf. Fig. 2.11). Moreover, the 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 (APR) and CPU time required are summarized in Table 2.3. These are obtained using the RBF method (Eq.(2.29) and Eq.(2.32)), MARS-2 system (Eqs.(2.4)-(2.6)), and OPT-RF system (Eqs.(2.7)-(2.8)). Notice that all methods employ norm-1 metric distance to obtain initial retrieval results at $t = 0$. A total of 35 queries were selected from different categories. 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 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 RBF interactive learning, the performance can be improved to greater than 90% precision. Due to the limitation in the degrees of adaptivity, MARS-2 provides the lowest performance gains and converges at about 62% precision. It is observed that the learning capability of RBF is more robust than that of OPT-RF, not only in retrieval accuracy, but also learning speed. As presented in Table 2.3, results after *one* round of the RBF is similar to results after *three* rounds of the

Figure 2.9. Example images from Corel database.

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 positive and negative learning strategies. Notice that OPT-RF requires the user to specify weight parameters in the form

of a slider bar for learning, whereas RBF automatically evaluates these weight parameters from the fed-back images.

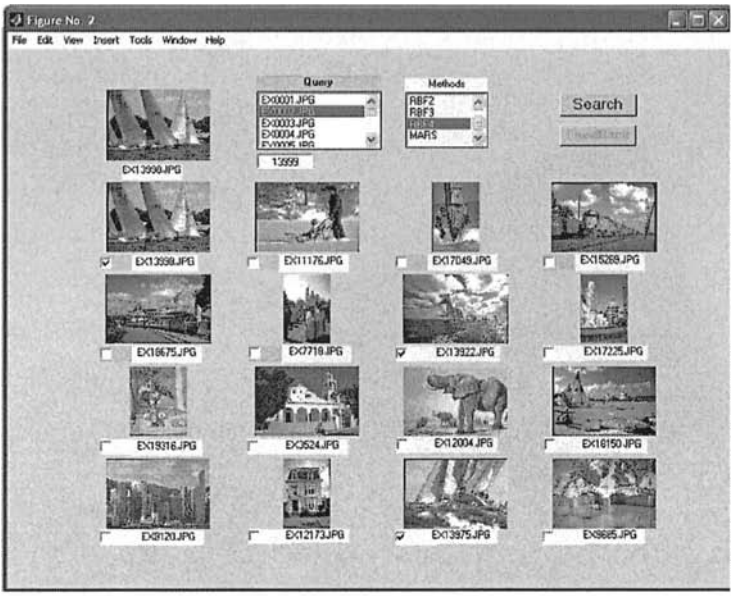
In regard to CPU time for the retrievals, the RBF approach is longer, at 2.34 seconds per iteration for a single query. However, the RBF method gains about 80% precision within only the first iteration, i.e., in only 2.34 seconds. By contrast, though faster, the OPT-RF needs three iterations to reach this underlined performance, i.e., taking $1.27 \times 3 = 3.81$ seconds. In other words, 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, 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. 2.10-2.13. Fig. 2.10 shows retrieval results of the "Yacht" query. Fig. 2.10(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. 2.10(b) shows the improvement of retrieval after three rounds of RBF interactive learning. This is superior to the results obtained by MARS-2 (cf. Fig. 2.11(a)) and OPT-RF (cf. Fig. 2.11(b)). The outstanding performance of the RBF method can also be seen from Figs. 2.12-2.13, showing the retrieval results in answering the "Tiger" query. As evidenced by the results of Fig. 2.10(b) and 2.12(b), it is observed that nonlinear analysis obtained by RBF can effectively capture high-level concepts in few retrieval sessions.

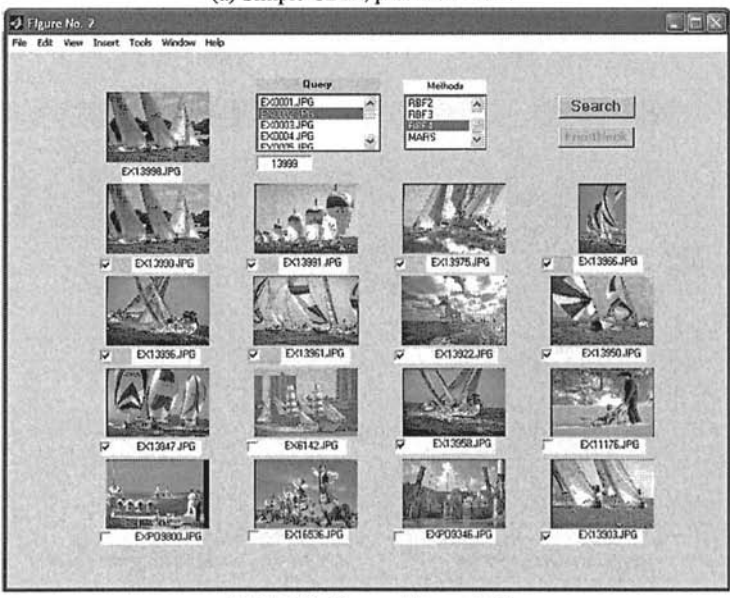
The above results were obtained from 'hard' queries, which require a high degree of nonlinear discrimination analysis. There are some queries that are relatively easier to retrieve, which are shown in Fig. 2.14. Those queries have prominent features, such as a shape in the "ROSE" query, and a combination of texture and color in the "POLO" query. In each case, it is observed that MARS-2 and OPT-RF show better performance than in the previous results. In such cases, however, the retrieval results obtained by RBF approached 100% precision.

Discussion

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



(a) Simple CBIR, precision = 0.19



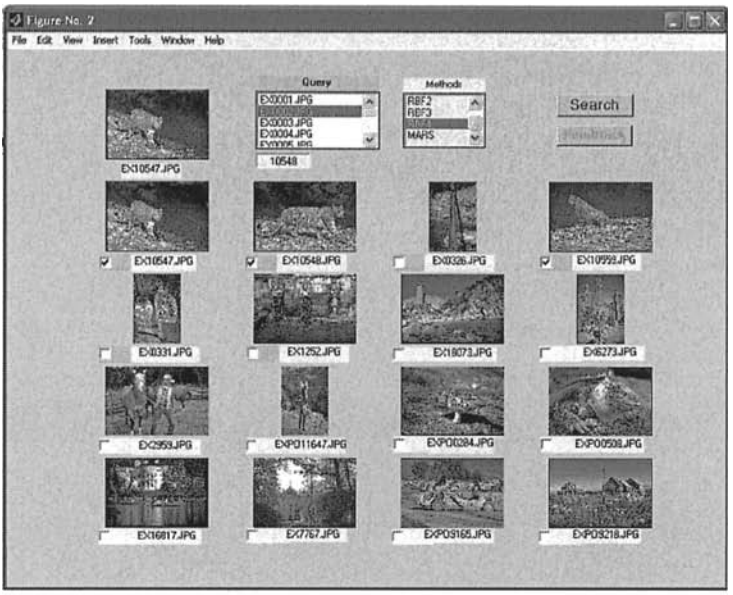
(b) RBF (t=3), precision = 0.69

Figure 2.10. Top sixteen retrieved images obtained by the "Yacht" query, using the Corel Database, (a) before RF learning, (b) after RF learning with RBF method

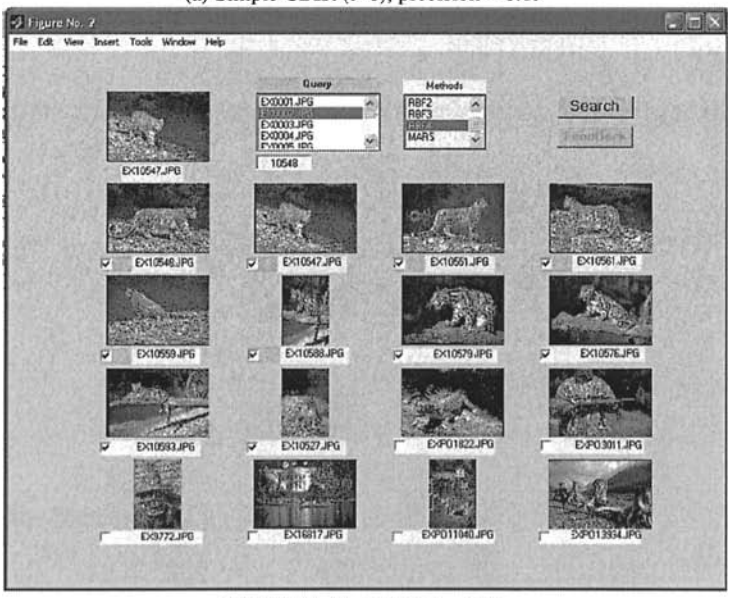
(a) MARS-2(t=3), precision = 0.31

(b) OPT-RF(t=3), precision = 0.19

Figure 2.11. (*a comparison to Fig. 2.10*) Top sixteen retrieved images obtained by the "Yacht" query , using the Corel Database, after RF learning with (a) MARS-2, and (b) OPT-RF.

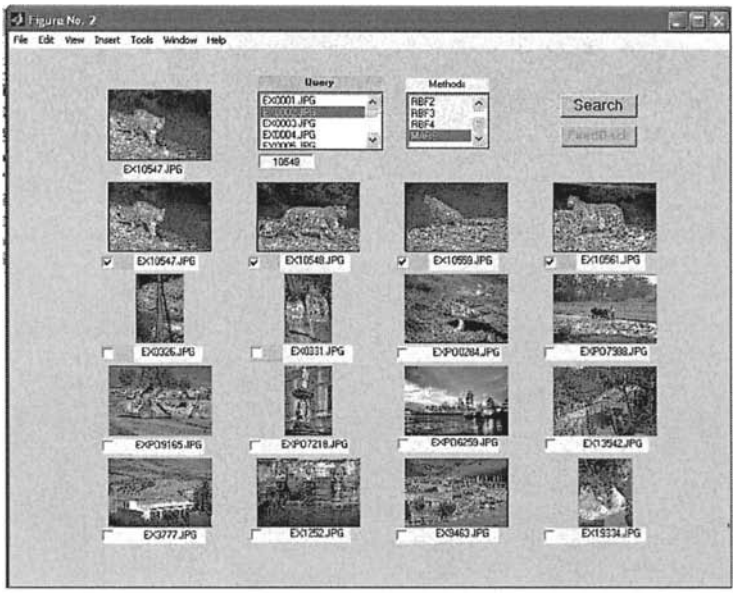


(a) Simple CBIR (t=0), precision = 0.19

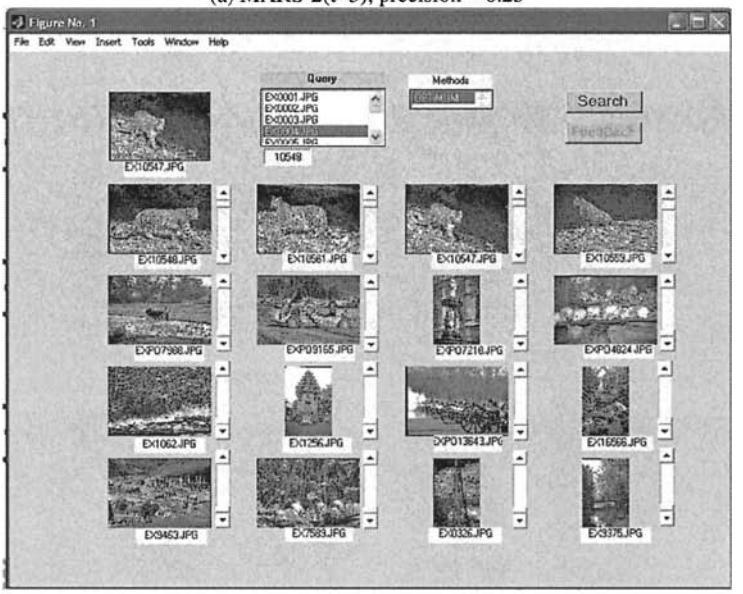


(b) RBF (t=3), precision = 0.63

Figure 2.12. Top sixteen retrieved images obtained by the "Tiger" query, using the Corel Database, (a) before RF learning, (b) after RF learning with RBF method.



(a) MARS-2(t=3), precision = 0.25



(b) OPT-RF(t=3), precision = 0.25

Figure 2.13. [*a comparison to Fig. 2.12*] Top sixteen retrieved images obtained by the "Tiger" query, using the Corel Database, after RF learning with (a) MARS-2, and (b) OPT-RF.

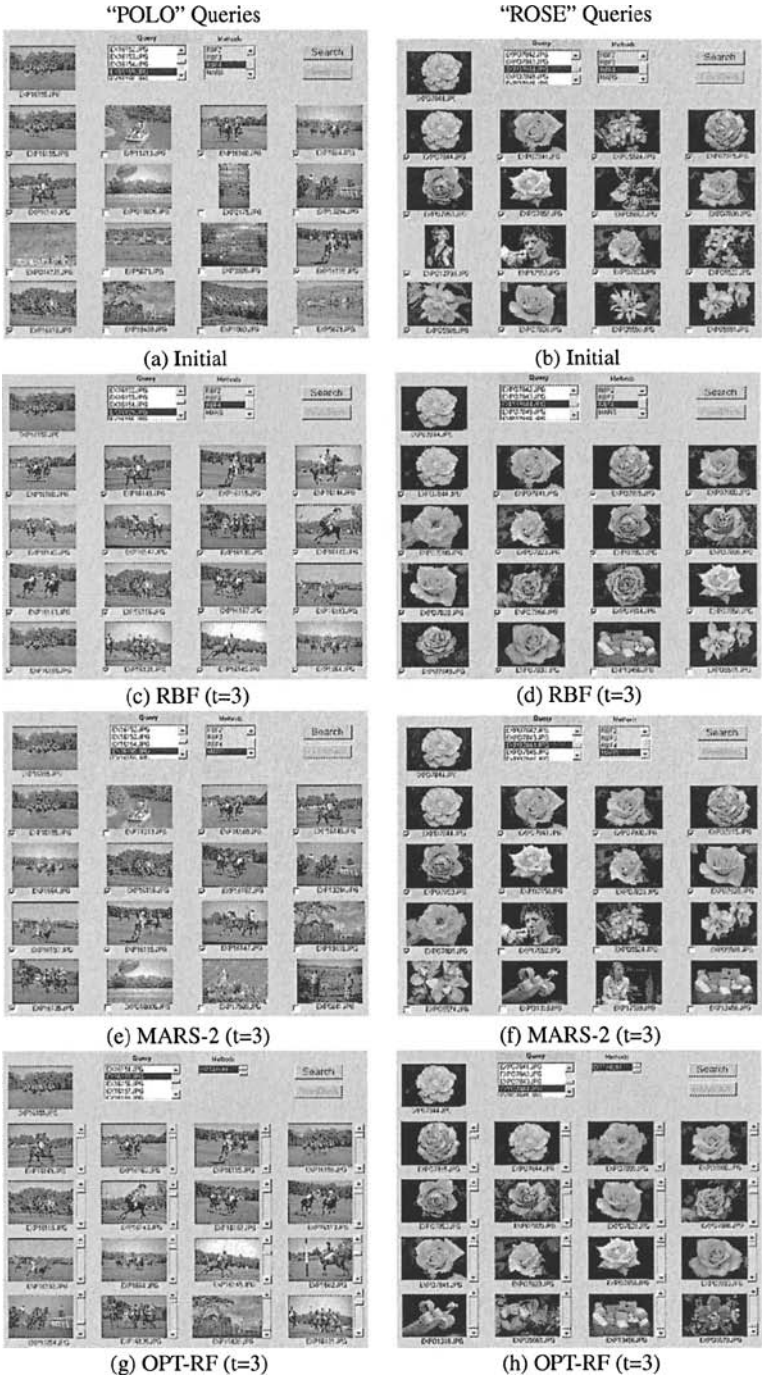


Figure 2.14. Retrieval results of "POLO" and "ROSE" queries, obtained by (a)-(b) Simple-CBIR, (c)-(d) RBF, (e)-(f) MARS-2, and (g)-(h) OPT-RF.

Color Descriptors	
Color Histogram (d=48) *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. The V component is discarded because of its sensitivity to the lighting condition [119]
Color Moments (d=9)	From a given RGB color image, the mean, standard deviation, and skew are extracted 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 128×128 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 are applied last to form the 48-demension 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 as polar coordinate. 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-dimensional feature vector.

Table 2.2. Content descriptions used for image characterization in the Corel database.

Method	t=0	t=1	t=2	t=3	CPU time (sec./iter.)
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

Table 2.3. Column 2-5: Average precision rate (%) obtained by retrieving 35 queries selected from different categories, using the Corel database; Column 6: Average CPU time (seconds per iteration) obtained by retrieving a single query, not including the time to display the retrieved images, using a 1.8 GHz Pentium IV processor and a MATLAB implementation.

shows that semantics and user request are more essential than 'index features' for optimum retrieval. This has directed a number of researchers to suggest that such a retrieval problem must be interpreted as human-centered, rather than computer centered [3, 4]. It has been shown in this chapter that these user information needs, in a visual-seeking environment, are well-addressed

by user-interface methodologies. User interface allows the retrieval system to overcome the problem of fuzzy understanding in the user's goals, and thus aid the expression of information needs. Two main points have been demonstrated by the current method: 1) learning-based systems can adjust their strategy in accordance with user input; and 2) user information 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. The RBF-based interactive method has emphasized the importance of "mapping" human perception onto the image-matching process. This 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 through learning from negative and positive samples. This results in a high performance learning machine that learns effectively and quickly from a small set of feedback data. It has been 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.

7. A Computer Aided Referral (CAR) System for Mine Target Detection in Side-Scan Sonar Images

Introduction

Computer aided detection (CAD) method has found many potential applications in our society, especially in defense and in medicine, such as the detection of underwater mine-like objects in side-scan sonar images and the detection of breast cancer in digital mammograms.

In the detection of underwater mine-like objects, the CAD method points to particular areas in a sonar image and label the areas as "potentially dangerous". Because of technological limitations imposed by state-of-the-art in image processing and pattern recognition, a significant percentage of the positive cases are not detected by the CAD systems in time for early actions. Similar problems have been observed in the detection of breast cancer: about 10% of the malignant cases are not detected in time for early treatment.

In light of the problems associated with the conventional CAD methods, it is proposed to investigate an alternative way, applying content-based image retrieval (CBIR) techniques to assist operators/doctors. The CBIR techniques take a computer aided referral (CAR) approach based on the content information embedded in the images. This section will present the work on the development of a CAR system for the detection of underwater mine-like objects in side-scan sonar images.

Underwater mines may cause serious problems for surface and submarine vessels. Side-scan sonar has been recognized as an effective way of detecting mine objects. However, human operators must monitor the images collected by the sonar, and targets might be missed due to inconsistencies in performance. Since the underwater environment is normally very complicated with noise and background clutter of various kinds, it is very important to identify the mine objects in the sonar images so that the operator can better correlate the detection results with the potential targets visually observed. In addition, an effective detection method is a vital step towards the recognition of the mines.

In the CAR approach, a digital library is first established which contains sonar images with signatures of various mine objects, and objects with signatures similar to mines. In operation, when a suspicious object is observed in a captured sonar image, this image will be compared with the images stored in the sonar image library based on their visual characteristics (features). Then the N images with the most similar characteristics to the captured image are brought to the attention of the operator. The operator will compare the captured image with those retrieved from the library. If the captured image indeed has similar visual features to one or more images with mines in the library, this image will be more carefully studied by the operator in order to make an inferred decision of the case on hand. Because the CAR approach takes a much broader range of possible characteristics of the mine signatures into consideration, instead of simply attempting to match to a "standard" template as a CAD method does, it can potentially minimize the high miss rate experienced by the CAD method.

Brief Description and Summary of the Preliminary Study

In a preliminary study, a library of sonar images was received from Defense Research and Development Canada (DRDC) Atlantic. The library contains 383 images with a variety of targets. Some of the images also have multiple targets. Fig. 2.15 show images of some of the targets. There are 10 different types of mines in these images. A CAR system is developed to aid the mine detection. This is a semi-automatic approach where we short list the number of probable

classes of the target to N ($N \leq 3$) using the content-based image retrieval techniques. The operators can then further investigate the short listed images for faster and more accurate detection of mines. In general, a CAR mine detection system consists of three stages, enhancing/denoising the images, locating the targets (both mine and non-mine objects) in the images, and detecting the mines.

In operation, the regions of interest was extracted from the sonar images in the database. These cropped images are then used for extraction of the features for the classification. The system consists of the following modules:

- Preprocessing Module
- Feature Extraction Module
- Search Engine based on linear similarity measurement
- Graphical User Interface (GUI)

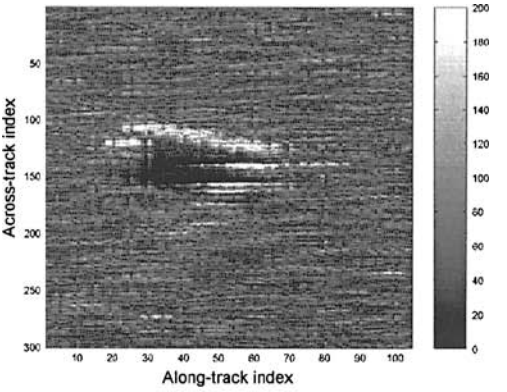
The image data is read from the files in the image library first. The preprocessing module is then used to convert the data into the required format and extract the region of interest from the images. Next the data is fed to the feature extraction module. This module extracts 29 statistical features representing each object in the sonar image. These features are then used by the search engine to find the top N matches. MatLab and SPSS were used for the implementation of the analysis and classification modules of the system. The system is designed in such a way that a user can present a sonar image for analysis through the GUI. The system extracts the features from the image and gives the N most probable classes of the sonar images in the image library to which the input image might belong. The user can then further investigate and find if the signature in the captured image is indeed that of a mine. It has been observed that, when setting $N = 2$, the detection rate is 77.29% with a false positive rate of 17.81% . If $N = 3$ was set, the detection rate is substantially increased to 95.12% , but the false positive rate is also high, over 40% .

Discussions

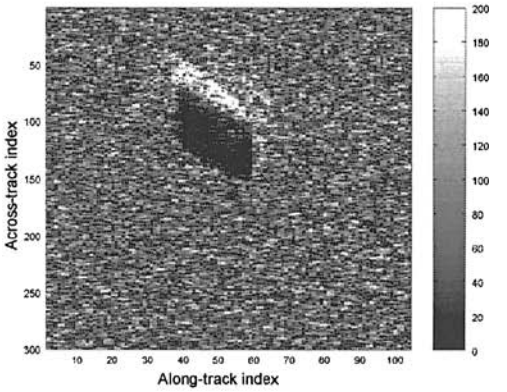
Our preliminary results demonstrated that the CAR approach is potentially an effective method in detecting underwater mine-like objects in side-scan sonar images. The more attractive result is observed when $N = 3$ with 95.12% detection rate. Although the false positive rate is also high, the key is that only very few mine signatures are missed. However, the high false positive rate has

to be addressed before the system could be useful. In fact, the high false positive rate is easy to explain: the set of data samples is too small, especially the samples in the mine classes. Since the CAR system is based on image retrieval, performance of which is critically dependent on the comprehensiveness of the image library and the quality of the features. With less than 90 samples, the mine classes are highly under-represented. Therefore, the foremost important issue to reduce the false positive rate while keeping the detection rate high is to have a large sonar image library which contains mine samples with all the possible characteristics, e.g. different scales, different angles of rotations, etc. In addition, the following steps should also improve the performance the CAR system.

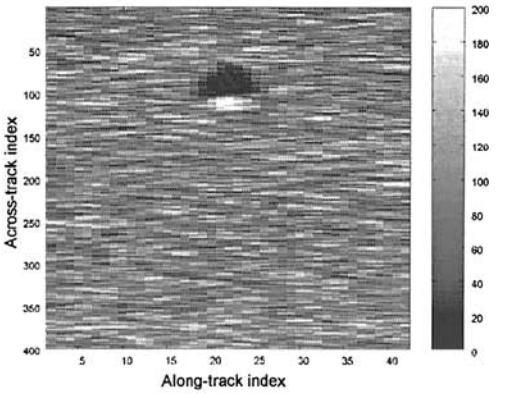
- Currently only low level texture features are used in retrieval. Exploring more effective features such as the newly developed wavelet modeling by mixture of Laplacians, and shape-based features will certainly enhance the performance of the system.
- A linear function is used for similarity matching in the CAR system. We can apply more advanced similarity measurements such as the RBF networks or support vector machines to improve the pattern recognition module of the system.
- Relevance feedback is an effective means to improve retrieval performance. Incorporating automatic relevance feedback techniques will improve the retrieval accuracy without adding burden to the operators.
- Some sonar images are subject to different kinds of noise. Denoising can help to improve the quality of the images and lead to better detection performance.



(a) MOG5 Cylinder



(c) Q260 Manta



(e) MK 56

Figure 2.15. Sindescan-sonar images of some of the targets

http://www.springer.com/978-0-387-25627-6

Multimedia Database Retrieval:
A Human-Centered Approach
Muneesawang, P.; Guan, L.
2006, VIII, 188 p. 61 illus., Hardcover
ISBN: 978-0-387-25627-6