Optimizing Learning in Image Retrieval

Yong Rui
Microsoft Research
One Microsoft Way, Redmond, WA 98052
yongrui@microsoft.com

Thomas Huang
University of Illinois at Urbana-Champaign
405 N. Matthews Ave., Urbana, IL 61801
huang@ifp.uiuc.edu

Abstract

Combining learning with vision techniques in interactive image retrieval has been an active research topic during the past few years. However, existing learning techniques either are based on heuristics or fail to analyze the working conditions. Furthermore, there is almost no in depth study on how to effectively learn from the users when there are multiple visual features in the retrieval system. To address these limitations, in this paper, we present a vigorous optimization formulation of the learning process and solve the problem in a principled way. By using Lagrange multipliers, we have derived explicit solutions, which are both optimal and fast to compute. Extensive comparisons against state-of-the-art techniques have been performed. Experiments were carried out on a large-size heterogeneous image collection consisting of 17,000 images. Retrieval performance was tested under a wide range of conditions. Various evaluation criteria, including precision-recall curve and rank measure, have demonstrated the effectiveness and robustness of the proposed technique.

1. Introduction

Early research in image retrieval has been focused on low-level vision alone [1, 2]. Unfortunately, after years of research, the retrieval performance is still far from users' expectations. Past efforts have made it clear that learning techniques need to be integrated into the retrieval system. Learning is a general concept. It can be from statistic models. It can also be from humans who are already in the vision system. This paper focuses on the latter learning paradigm.

One of the interactive learning techniques is relevance feedback, originally developed in the information retrieval community [3]. In recent years, it has been brought to visual image retrieval [4, 5, 6]. During retrieval, the users interact with the system and rate the "relevance" of the images retrieved by the system according to their true information needs. Based on the feedback, the system dynamically learns and updates its query structure that best captures users' concepts.

There are two important components to be learned in the retrieval systems. One is an appropriate transformation that maps the original visual feature space into a space that better models user desired high-level concepts. As a special case, this transformation can be as simple as re-weighting different axes in the original feature space. The other important component is the "ideal" query in the user's mind. For example, a user may not initially have the query image at hand or the ideal query may evolve during the retrieval process.

By converting the retrieval process into a learning process of the above two components, we can avoid *ad hoc* solutions and can approach this problem in a principled way. There exist various techniques in learning the above two components. MARS [4] proposed two independent learning techniques for the two components based on intuitive heuristics. MindReader [7] developed a more vigorous formulation of the problem but failed to analyze the working conditions. To address these limitations, we will propose an optimization-based learning technique in this paper that not only works in all conditions but also has principled explicit solutions.

The rest of the paper is organized as follows. In Section 2, we introduce important concepts and notations used in the paper. In Section 3, we review related work in this research field and discuss their strength and weakness. Efforts of resolving the limitations in the existing techniques lead to the global optimization approach proposed in Section 4. We will give detailed descriptions of the problem formulation, derivation of explicit optimal solutions, and computation complexity analysis. Evaluation of an image retrieval system's performance has been a weak spot in the past. In this paper, we have performed extensive experiments over a large heterogeneous image collection consisting of 17,000 real-world images. Various retrieval performance criteria, such as precision-recall curve and rank measure, have been used to validate the proposed algorithm. These experimental results are reported in Section 5. Discussions, conclusions and future work are given in Section 6.

2. Concepts and Notations

In this section, we describe important concepts and their notations that will be used throughout the paper. Let I be the number of features we are studying and let M be the total number of images in the database. We use $\vec{x_{mi}} = [x_{mi1}, \cdots, x_{mik}, \cdots x_{miK_i}]$ to denote the i^{th} feature vector of the m^{th} image, where K_i is the length of the feature vector i. For example, for a six-element color moment feature vector $K_i = 6$.

Let $\vec{q_i} = [q_{i1}, \cdots, q_{ik}, \cdots q_{iK_i}]$ be a query vector in feature i's feature space. To compute the distance g_{mi} between the two points $\vec{q_i}$ and $\vec{x_{mi}}$, we need to define a distance metric. The Norm-2 (Euclidean) metric is chosen because of its nice properties in quadratic optimization. There are several variants of the Euclidean distance: plain Euclidean, weighted Euclidean and generalized Euclidean.

- Plain Euclidean

$$g_{mi} = (\vec{q_i} - \vec{x_{mi}})^T (\vec{q_i} - \vec{x_{mi}}) \tag{1}$$

- Weighted Euclidean

$$g_{mi} = (\vec{q_i} - \vec{x_{mi}})^T \ \Lambda_i \ (\vec{q_i} - \vec{x_{mi}}) \tag{2}$$

  where Λ_i is a diagonal matrix and its diagonal elements model the different importance of x_{mik}.

- Generalized Euclidean

$$g_{mi} = (\vec{q_i} - \vec{x_{mi}})^T \ W_i \ (\vec{q_i} - \vec{x_{mi}}) \tag{3}$$

  where W_i is a real symmetric full matrix.

Plain Euclidean cannot model any transformation between different feature spaces. Weighted Euclidean can reweight the original feature space. Generalized Euclidean can both map the original space to a new space and reweight the transformed space.

[Theorem 1] For a real symmetric matrix W_i, it can be decomposed into the following form [8]:

$$W_i = P_i^T \ \Lambda_i \ P_i \tag{4}$$

where P_i is an orthonormal matrix consisting W_i's eigen vectors and Λ_i is a diagonal matrix whose diagonal elements are the eigen values of W_i.

Based on the theorem, the generalized Euclidean distance can be re-written as:

$$\begin{aligned} g_{mi} &= (\vec{q_i} - \vec{x_{mi}})^T \ W_i \ (\vec{q_i} - \vec{x_{mi}}) \\ &= (\vec{q_i} - \vec{x_{mi}})^T \ P_i^T \ \Lambda_i \ P_i \ (\vec{q_i} - \vec{x_{mi}}) \\ &= (P_i \ (\vec{q_i} - \vec{x_{mi}}))^T \ \Lambda_i \ (P_i \ (\vec{q_i} - \vec{x_{mi}})) \end{aligned}$$

The above derivation says that the old feature space is first transformed into a new feature space by P_i and then the new feature space is re-weighted by Λ_i.

So far we have only discussed how to compute image distances based on an individual feature. As for the overall distance d_m based on multiple features, it can be computed in two ways. One way is to not differentiate the difference between a feature element and a feature and stack all the feature elements (from all the individual features) into a big overall feature vector and then use Equations 1 - 3 to compute d_m. This approach was used in most of the existing systems. Because this model has no hierarchy, we refer it as the "flat model" in this paper. Another way is to construct a hierarchical model, where the overall distance d_m is defined as:

$$d_m = U(g_{mi}) \tag{5}$$

where $U(.)$ is a function that combines the individual distances g_{mi} to form the overall distance d_m. We will refer this model as the "hierarchical model". This model is a fundamental part of the proposed approach. We will show in Section 5 how this model significantly outperforms the flat model.

As stated in Section 1, there are two components that need to be learned by relevance feedback. One is the feature space transformation and the other is the optimal query vector. Following this section's notations, the former includes the learning of W_i and $U(.)$ and the latter is to learn $\vec{q_i}$.

3. Related Work

Most of the existing techniques have used the flat model and ignored $U(.)$. Even for learning the flat model (W_i and $\vec{q_i}$ only), there is still much room for improvements.

3.1. The MARS approach

The MARS system was among the first in the field that introduced relevance feedback into image retrieval [4]. It proposed two independent techniques for learning W_i and $\vec{q_i}$. For the former, the MARS system assumes W_i will take a diagonal form, thus using the weighted Euclidean metric. The heuristics for learning the weights (diagonal elements) were based on the following observation. If a particular feature element captures a user's query concept, that element's values x_{ik} will be consistent among all the positive examples given by the user. The standard deviation of all the x_{ik}'s will therefore be small. The inverse of the standard deviation thus furnishes a good estimate of the weight for feature element x_{ik}.

$$w_{i_{kk}} = \frac{1}{\sigma_{ik}} \tag{6}$$

where $w_{i_{kk}}$ is the kk^{th} element of matrix W_i and σ_{ik} is the standard deviation of the sequence of x_{ik}'s.

The MARS system also proposed a technique for learning the query vectors. The learned query vector should move towards the positive examples and away from negative examples:

$$\begin{aligned} \vec{q}'_i &= \alpha \vec{q_i} + \beta (\frac{1}{N_{R'}} \sum_{n \in D'_R} \vec{x}_{ni}) - \gamma (\frac{1}{N_{N'}} \sum_{n \in D'_N} \vec{x}_{ni}) \\ i &= 1, \cdots, I \end{aligned}$$

where $alpha, \beta$, and γ are suitable constants [3]; $N_{R'}$ and $N_{N'}$ are the numbers of images in the relevant set D'_R and non-relevant set D'_N; and $\vec{x}_{ni}$ is the n^{th} training sample in the sets D'_R and D'_N.

Even though working reasonably well, the MARS techniques were based on *ad hoc* heuristics and did not have a solid theoretical foundation. Since its appearance, many improved versions have been proposed. One of the most elegant approaches is MindReader.

3.2. The MindReader approach

The MindReader system was developed by Ishikawa et al. [7]. This system integrated the two independent learning processes in MARS into a single algorithm and proposed a well-founded theoretical framework for the learning process.

Instead of being a diagonal matrix as in the MARS system, W_i is a full matrix in this algorithm to model the generalized Euclidean distance. By minimizing the distances between the query vector and all the positive fedback examples, MindReader system obtained the following optimal solutions to $\vec{q}_i$ and W_i [7]:

$$\vec{q}_i^{\,T^*} = \frac{\vec{\pi}^{\,T} X_i}{\sum_{n=1}^{N} \pi_n} \quad (7)$$

$$W_i^* = (det(C_i))^{\frac{1}{K_i}} C_i^{-1} \quad (8)$$

where N is the number of positive examples and π_n is the degree of relevance for image n given by the user. X_i is the example matrix obtained by stacking the N training vectors ($\vec{x}_{ni}$) into a matrix. It is therefore a $(N \times K_i)$ matrix. The term C_i is the weighted $(K_i \times K_i)$ covariance matrix of X_i. That is,

$$C_{i_{rs}} = \frac{\sum_{n=1}^{N} \pi_n \, (x_{nr} - q_r) \, (x_{ns} - q_s)}{\sum_{n=1}^{N} \pi_n}$$
$$r, s = 1, \cdots, K_i$$

A major *difference* between the MindReader approach (Equation 8) and the MARS approach (Equation 6) is that W_i is a full matrix in the former but a diagonal matrix in the latter. The advantages and disadvantages of these two methods will be demonstrated by experiments in Section 5.

The MindReader approach avoided *ad hoc* heuristics and developed a mathematical framework for learning W_i and $\vec{q}_i$. However, it failed to analyze the working conditions. In fact, even though elegant in theory, it faces many difficulties in reality.

3.2.1 Discussions

In order to obtain W_i (Equation (8)), we need to compute the inverse of the covariance matrix C_i. It is clear that, if $N < K_i$, then C_i is not invertible and we cannot obtain W_i. In MindReader, the authors proposed a solution to solve this by using a pseudo-inverse defined below [7].

The singular value decomposition (SVD) of C_i is

$$C_i = A \, \Lambda \, B^T \quad (9)$$

where Λ is a diagonal matrix: $diag(\lambda_1, \cdots, \lambda_k, \cdots, \lambda_{K_i})$. Those λ's are either positive or zero. Suppose there are L nonzero λ's, the pseudo-inverse of C_i is defined as

$$C_i^+ = A \, \Lambda^+ \, B^T$$
$$\Lambda^+ = diag(\frac{1}{\lambda_1}, \cdots, \frac{1}{\lambda_L}, 0, \cdots, 0).$$

where $+$ denotes the pseudo-inverse of a matrix. The approximation solution to W_i^* is then [7]

$$W_i^* = (\prod_{l=1}^{L} \lambda_l)^{\frac{1}{L}} \, C_i^+ \quad (10)$$

Even though, in theory, we can get around the *singular* problem by using the above procedure, in reality this solution does not give satisfactory results. This is especially true when N is far less than K_i. Remember, we need to use $(N-1) \times K_i$ numbers from the training samples to estimate $\frac{K_i \, (K_i+1)}{2}$ parameters in matrix C_i. In MindReader, the authors used a $K_i = 2$ example to show the performance of the algorithm. However, in real image retrieval systems, feature vectors' dimensions are much higher. For example, in HSV color histograms, the feature vector's dimension can be as high as $(8 \times 4 \times 1 = 32)$ [2]. During retrieval, in most situations, the condition $N > K_i$ will not be satisfied and this algorithm performs poorly (see Section 5).

4. The Proposed Approach

As reviewed above, there are three major difficulties in the existing systems: *ad hoc* heuristics, limited working conditions, and most importantly utilizing the flat model to compute the overall distance. To address these difficulties, in this section, we will propose an optimization-based learning algorithm that not only works in all conditions, but also has explicit optimal solutions for multiple visual features simultaneously.

4.1. Problem formulation

We model each individual feature's similarity as the generalized Euclidean distance because of its powerfulness and model the overall similarity as linear combinations of each individual feature's similarity because of its simplicity. That is, W_i takes the form of a matrix and $U(.)$ takes the form of a vector $\vec{u} = [u_1, \cdots, u_i, \cdots, u_I]$. The above choices are after careful considerations which, for clarity, will be presented in Section 6.

Let N be the number of retrieved relevant images (training samples). Let π_n be the degree of relevance for training sample n given by the user. The overall distance between a training sample and a query is defined as:

$$d_n = \vec{u}^T \vec{g}_n \quad (11)$$
$$\vec{g}_n = [g_{n1}, \cdots, g_{ni}, \cdots, g_{nI}]^T \quad (12)$$
$$g_{ni} = (\vec{x}_{ni} - \vec{q}_i)^T W_i (\vec{x}_{ni} - \vec{q}_i) \quad (13)$$

The above distance definition leads to the following optimization problem:

$$min\ J = \vec{\pi}^T \times \vec{d} \quad (14)$$
$$\vec{d} = [d_1, \cdots, d_n, \cdots, d_N]^T \quad (15)$$
$$d_n = \vec{u}^T \vec{g}_n \quad (16)$$
$$\vec{g}_n = [g_{n1}, \cdots, g_{ni}, \cdots, g_{nI}]^T \quad (17)$$
$$g_{ni} = (\vec{x}_{ni} - \vec{q}_i)^T W_i (\vec{x}_{ni} - \vec{q}_i) \quad (18)$$
$$s.t. \quad \sum_{i=1}^{I} \frac{1}{u_i} = 1 \quad (19)$$
$$det(W_i) = 1 \quad (20)$$
$$n = 1, \cdots, N \quad (21)$$
$$i = 1, \cdots, I \quad (22)$$

It is easy to see that if there are no constraints for $\vec{u}$ and W_i, this optimization problem will reduce to a trivial solution of all zeros. We therefore enforce Equations (19) and (20) as constraints for scaling purposes. This problem formulation is a general framework which can include both MARS and MindReader. If we would disregard the overall distance (d_n) and only concentrate on each individual distance (g_{ni}), a diagonal matrix of W_i would reduce this formulation to the MARS algorithm and a full matrix of W_i would reduce this formulation to the MindReader approach.

The above objective function says that optimality will be achieved only if both the transformations ($\vec{u}$ and W_i) and query vectors $\vec{q}_i$ are optimally learned. This will be accomplished by minimizing the distances between the "ideal" query and all the positive fedback examples. The degree of relevance π_n of each example is given by the user according to his or her judgment. The objective function J is linear in $\vec{u}$ and W_i and quadratic in $\vec{q}_i$. We will first use Lagrange multipliers to reduce this constrained problem to an unconstrained one, and then de-couple the problem by first solving $\vec{q}_i$, and then W_i and $\vec{u}$. The following is the unconstrained problem:

$$L = \vec{\pi}^T \times \vec{d} - \lambda(\sum_{i=1}^{I} \frac{1}{u_i} - 1) - \sum_{i=1}^{I} \lambda_i (det(W_i) - 1) \quad (23)$$

4.2. Optimal solution for $\vec{q}_i$

$$\frac{\partial L}{\partial \vec{q}_i} = \vec{\pi}^T \times \begin{bmatrix} \frac{\partial d_1}{\partial \vec{q}_i} \\ \cdots \\ \frac{\partial d_n}{\partial \vec{q}_i} \\ \cdots \\ \frac{\partial d_N}{\partial \vec{q}_i} \end{bmatrix}$$

$$= \vec{\pi}^T \times \begin{bmatrix} -2\ u_i\ (\vec{x}_{1i} - \vec{q}_i)^T\ W_i \\ \cdots \\ -2\ u_i\ (\vec{x}_{ni} - \vec{q}_i)^T\ W_i \\ \cdots \\ -2\ u_i\ (\vec{x}_{Ni} - \vec{q}_i)^T\ W_i \end{bmatrix}$$

By setting the above equation to zero, we can obtain the final solution to $\vec{q}_i$:

$$\vec{q}_i^{\ T^*} = \frac{\vec{\pi}^T X_i}{\sum_{n=1}^{N} \pi_n} \quad (24)$$

where X_i is the training sample matrix for feature i, obtained by stacking the N training vectors ($\vec{x}_{ni}$) into a matrix. It is therefore an ($N \times K_i$) matrix. Equation (24) closely matches our intuition. That is, $\vec{q}_i^{T^*}$ (the optimal query vector for feature i) is nothing but the weighted average of the training samples for feature i.

4.3. Optimal solution for W_i

$$\frac{\partial L}{\partial w_{i_{rs}}} = \vec{\pi}^T \times \begin{bmatrix} \vec{u}^T \frac{\partial \vec{g}_1}{\partial w_{i_{rs}}} \\ \cdots \\ \vec{u}^T \frac{\partial \vec{g}_n}{\partial w_{i_{rs}}} \\ \cdots \\ \vec{u}^T \frac{\partial \vec{g}_N}{\partial w_{i_{rs}}} \end{bmatrix} - \lambda_i\ (-1)^{r+s} det(W_{i_{rs}})$$

$$= \sum_{n=1}^{N} \pi_n\ (x_{nir} - q_{ir})(x_{1is} - q_{is}) - \lambda_i\ (-1)^{r+s} det(W_{i_{rs}})$$

After setting the above equation to zero, we get:

$$W_i^* = (det(C_i))^{\frac{1}{K_i}}\ C_i^{-1} \quad (25)$$

where the term C_i is the ($K_i \times K_i$) weighted covariance matrix of X_i. That is, $C_{i_{rs}} = \sum_{n=1}^{N} \pi_n (x_{nir} - q_{ir})(x_{1is} - q_{is}) / \sum_{n=1}^{N} \pi_n, r, s = 1, \cdots, K_i$.

Note that in MARS, W_i is always a diagonal matrix. This limits its ability to modeling transformations between feature spaces. On the other hand, MindReader's W_i is always a full matrix. It cannot be reliably estimated when the number of training samples (N) is less than the length of the feature vector (K_i). Unlike these two algorithms, the proposed technique dynamically and intelligently switches between a diagonal matrix and a full matrix, depending on the relationship between N and K_i. When $N < K_i$, the proposed algorithm forms a diagonal matrix to ensure reliable estimation; and when $N > K_i$, it will form a full matrix to take full advantage of the training samples.

4.4. Optimal Solution for $\vec{u}$

To obtain u_i^*, set the partial derivative to zero. We then have

$$\frac{\partial L}{\partial u_i} = \sum_{n=1}^{N} \pi_n \, g_{ni} + \lambda \, u_i^{-2} = 0, \; \forall i \qquad (26)$$

Multiply both sides by u_i and summarize over i. We have

$$\sum_{i=1}^{I} u_i \, (\sum_{n=1}^{N} \pi_n \, g_{ni}) + \lambda (\sum_{i=1}^{I} \frac{1}{u_i}) = 0 \qquad (27)$$

Since $\sum_{i=1}^{I} \frac{1}{u_i} = 1$, the optimal λ is

$$\lambda^* = - \sum_{i=1}^{I} u_i \, f_i \qquad (28)$$

where $f_i = \sum_{n=1}^{N} \pi_n \, g_{ni}$. This will lead to the optimal solution for u_i:

$$u_i^* = \sum_{j=1}^{I} \sqrt{\frac{f_j}{f_i}} \qquad (29)$$

This solution tells us, if the total distance (f_i) of feature i is small (meaning it is close to the ideal query), this feature should receive a higher weight and vice versa.

The solutions for $\vec{q}_i$ and W_i have been partially studied in MARS and MindReader. The solution for u_i, however, has not been investigated by either system. Both MARS and MindReader do not differentiate the difference between feature elements and features and use a flat image content model. This is not only computationally expensive, but also far less effective in retrieval performance. For computation complexity, take MindReader as an example. It needs $O((\sum_i^I K_i)^3 + 2N(\sum_i^I K_i)^2))$ multiplications or divisions while the proposed algorithm only needs $O(\sum_i^I \; ((K_i)^3 + 2N(K_i)^2))$ operations. Note that the different locations of $\sum_i^I$ in the two formulae result in *significantly* different computation counts.

5. Experiments, Results and Evaluations

5.1. Data set

In the experiments reported in this section, all the algorithms are tested on the Corel data set. This data set meets all the requirements to evaluate an image retrieval system. It is large, heterogeneous and has human annotated ground truth. This data set consists of 17,000 images, covering a wide variety of content ranging from animals and birds to Tibet and Czech Republic. Each category contains 100 images and these images are classified by domain professionals. In the experiments, images from the same category are considered relevant. Note that the ground truth we used in the experiments are based on high-level concepts. They are much more difficult to achieve than visual similarities. But they are the *ultimate* queries that users would like to ask. We therefore did not count an image as a correct answer even if it is visually similar to the query image but represents different high-level concepts.

The Corel data set was also used in other systems and relatively high retrieval performance was reported. However, those systems only used pre-selected categories with distinctive visual characteristics (e.g., cars vs. mountains). In our experiments, no pre-selection is made. We believe only in this manner can we obtain an objective evaluation of different retrieval techniques.

5.2. Queries

Some existing systems only used pre-selected images as the queries. It is arguable that those systems will perform equally well on other not-selected images. Other systems only tested on queries with unique answers. This is called "point queries" in database research community. This type of queries is used to model *exact* matches, e.g., name = "John Smith". On the other hand, "range queries" are used to accomplish similarity-based matches, e.g., find all students whose ages are between 10 and 20. It is therefore more appropriate to use range queries to evaluate image retrieval systems. For example, find all the images that contain animals. In our experiments reported here, there is no pre-selected query images and all the queries are range queries. We randomly generated 400 queries for each retrieval condition. The reported retrieval performance is then the average of all the 400 queries against ground truth as annotated by Corel professionals. We execute queries in this very careful manner to ensure meaningful evaluations.

5.3. Visual features

There are three features used in the system: color moments, wavelet based texture, and water-fill edge feature. The color space we use is HSV because of its decorrelated coordinates and its perceptual uniformity [2]. We extract the first two moments (mean and standard deviation) from the three color channels and therefore have a color feature vector of length $3 \times 2 = 6$.

For wavelet based texture, the original image is fed into a wavelet filter bank and is decomposed into 10 de-correlated sub-bands. Each sub-band captures the characteristics of a certain scale and orientation of the original image. For each sub-band, we extract the standard deviation of the wavelet coefficients and therefore have a texture feature vector of length 10.

For water-fill edge feature vector, we first pass the original images through an edge detector to generate their corresponding edge maps. We then extract eighteen (18) elements from the edge maps, including *max fill time*, *max fork count*, etc. For a complete description of this edge feature vector, interested readers are referred to [9].

5.4. Performance measures

Precision-recall curve is the conventional information retrieval (IR) performance measure [3]. Precision (Pr) is de-

fined as the number of retrieved relevant objects (i.e., N) over the number of total retrieved objects. Recall (Re) is defined as the number of retrieved relevant objects (i.e., N) over the total number of relevant object (in our case 99). The performance for an "ideal" system is to have both high Pr and Re. Unfortunately, they are conflicting entities and cannot be at high values at the same time. Because of this, instead of using a single value of Pr and Re, a $Pr(Re)$ curve is normally used to characterize the performance of an IR system.

Even though well suited for text-based IR, $Pr(Re)$ is less meaningful in image retrieval systems where recall is consistently low. More and more researchers are adopting precision-scope curve to evaluate image retrieval performance [10]. Scope (Sc) specifies the number of images returned to the user. For a particular scope Sc, e.g., top 20 images, $Pr(Sc)$ can be computed as:

$$Pr(Sc) = \frac{N}{Sc} \tag{30}$$

Huang et. al. proposed another performance measure: the rank (Ra) measure [10]. The rank measure is defined as the average rank of the retrieved relevant images. It is clear that the smaller the rank, the better the performance. While $Pr(Sc)$ only cares if a relevant image is retrieved or not, $Ra(Sc)$ also cares what's the rank of that image. Caution must be taken when using $Ra(Sc)$, though. If $Pr_A(Sc) > Pr_B(Sc)$ and $Ra_A(Sc) < Ra_B(Sc)$, it says A is definitely better than B, because not only A retrieves more relevant images than B, but also all those retrieved images are closer to top in A than in B. But if $Pr_A(Sc) > Pr_B(Sc)$ and $Ra_A(Sc) > Ra_B(Sc)$, no conclusion can be made based on Ra.

5.5. System description

We have constructed an image retrieval system based on the optimization algorithm developed in Section 4. Figure 1 is its interface.

On the left are the query image and returned results (the top-left image is the query image). For each returned image, there is a degree-of-relevance slider. A user uses these sliders to give his or her relevance feedback to the system. On the right-hand side, there are progress controls displaying how W_i and $\vec{u}$ dynamically change during the retrieval.

5.6. Results and observations

The proposed approach (PP) differs from the MARS (MS) and MindReader (MR) approaches in two major ways. First, PP models image content hierarchically. It has a two-level feature transformation $\vec{u}$ and W_i. The learning via relevance feedback is also hierarchical. MS and MR, on the other hand, do not differentiate a feature element x_{nik} and a feature $\vec{x_{ni}}$ and use a flat image content model. The other

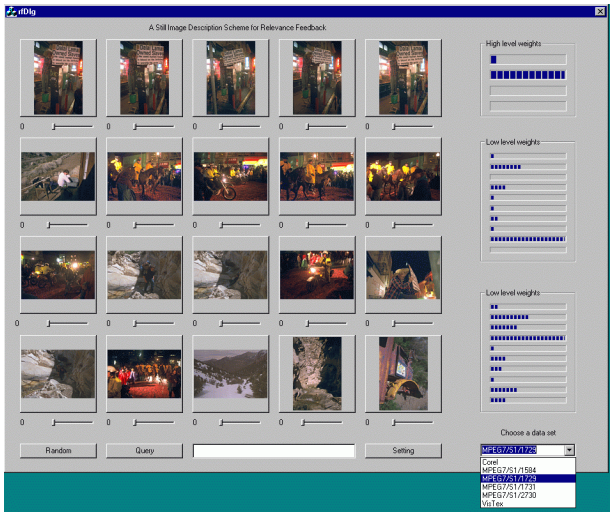


Figure 1. **The interface of the system**

major difference is the form of W_i. While MS uses a strict diagonal matrix and MR uses a strict full matrix, PP adaptively switches between the two forms depending on the relationship between N and K_i (Section 4.4). In addition to evaluate the above two differences, we will also study the working conditions for each of the approaches.

The experiments are configured into two cases. Case one uses only the color feature (referred as Case C) and case two uses all the three features (referred as Case CTE). Since the color feature has only 6 elements ($K_i = 6$), Case C simulates the condition that K_i is comparable to N. Note that we can not explicitly control the value of N, the number of relevant images, but we can implicitly control it by using different values of Sc. In general, a larger Sc implies a larger N, as illustrated in Figure 4 (N is proportional to recall Re given the total number of relevant images is a constant of 99). Since there is only a single feature in Case C, the flat model and the hierarchical model are the same in this case. The performance differences between the three approaches are coming from the form of W_i only. This gives us a concrete situation to quantify the amount of contribution from adaptive W_i switching alone(Section 4.3). Case CTE has multiple features. For the PP approach, $K_1 = 6$, $K_2 = 10$ and $K_3 = 18$. For MS and MR, $K_1 = 6 + 10 + 18 = 34$. This case gives us an ideal situation to study how the hierarchical content model affects retrieval performance and under which conditions each algorithm will work.

Table 1 is for case C and Table 2 is for case CTE. The top three rows in the tables are the results for $Sc = 20$, the middle three rows are for $Sc = 100$, and the bottom three rows are for $Sc = 180$. The first three columns in the two tables are Pr (in percentage) for zero, one and two iterations of relevance feedback. The last three columns in the tables are Ra for zero, one and two iterations of relevance

Table 1. **Case C: Comparisons when Sc = 20, 100, 180**

	0 rf	1 rf	2 rf	0 rf	1 rf	2rf
C(MS)	7.52	9.75	10.27	2.77	1.52	1.25
C(MR)	7.52	3.48	4.95	2.77	1.64	1.38
C(PP)	7.52	9.75	10.65	2.77	1.46	1.20
C(MS)	4.81	6.98	7.85	26.81	18.29	16.04
C(MR)	4.81	6.18	7.43	26.81	21.98	17.57
C(PP)	4.81	7.49	8.76	26.81	16.29	12.64
C(MS)	3.95	5.85	6.52	55.90	40.91	37.82
C(MR)	3.95	5.81	6.82	55.90	43.46	36.06
C(PP)	3.95	6.35	7.40	55.90	34.98	27.75

Table 2. **Case CTE: Comparisons when Sc = 20, 100, 180**

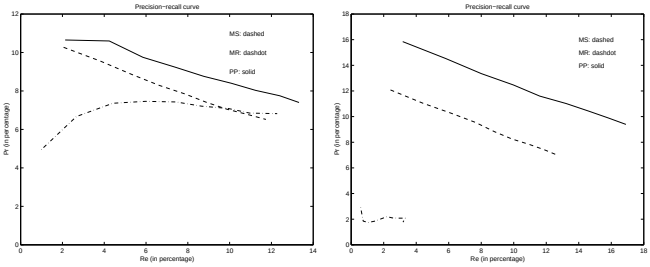
	0 rf	1 rf	2 rf	0 rf	1 rf	2rf
MS	7.23	10.99	12.09	3.00	1.56	1.27
MR	7.23	0.58	0.29	3.00	0.83	0.22
PP	10.18	14.18	15.85	1.71	1.20	1.10
MS	4.36	7.60	8.82	27.50	16.32	13.70
MR	4.36	1.02	2.20	27.50	24.61	14.72
PP	5.75	9.47	11.60	39.24	27.31	23.45
MS	3.53	6.00	7.02	53.83	35.88	30.81
MR	3.53	1.06	1.77	53.83	52.53	53.81
PP	4.63	7.78	9.39	125.56	83.74	67.47

feedback. The following observations can be made based the results of the two tables:

- PP approach performs consistently better in all conditions than the other two approaches. Case C (Table 1) demonstrates the gain of PP over MS and MR based on the adaptive switch. By utilizing this technique, the gain is about 5-10% increase. Note that, in this case, not only is PP's Pr higher than those of MS and MR, but also its rank is lower than those of MS and MR. That is, not only PP retrieves more relevant images than MS or MR, but also all the retrieved images are closer to top in PP than in MS or MR. Case CTE (Table 2) has multiple features. The gain that PP has over MS and MR is from both adaptive switching and hierarchical relevance feedback. The gain can be as much as 20-40%. This significant increase demonstrates the effectiveness of hierarchical image content modeling.
- MR approach achieves reasonable performance when N is comparable to or larger than K_i. For example, in Table 1 when $Sc = 180$, MR's performance is better than that of MS and is next to that of PP. This is because when there are sufficient training samples compared with K_i, the covariance matrix C_i can be reliably learned. This allows the algorithm to take advantage of the generalized Euclidean distance measure (Equation 3). But in situations where N is smaller than K_i, the algorithm simply falls apart, as indicated in Table 2 where $K_i = 34$.
- Overall, MS's performance ranks second. Its performance is comparable to PP when there is a single feature (Case C). Where there are multiple features, because it uses a flat image content model, its performance is significantly worse than that of PP. Furthermore, since it only uses diagonal matrix for W_i, this limits its ability to modeling transformations between feature spaces. In the case $Sc = 180$ in Table 1, its performance is even worse than that of MR.

Figures 2, 3 and 4 compare the $Pr(Re)$ curves, $Pr(Sc)$ curves, and $Re(Sc)$ curves in cases C and CTE, after two feedback iterations. The solid curves, dashed curves and dashdot curves are for PP, MS and MR, respectively. The values of Sc range from 20 to 180 with an increment of 20. We have the following observations based on the figures:

- $Pr(Sc)$ curve and $Pr(Re)$ curve depict the similar information. But as also being observed by other researchers [10], for image retrieval systems where Re is consistently low, $Pr(Sc)$ curve is more expressive for comparison than $Pr(Re)$ curve.
- Figures 3 and 4 tell us if we increase Sc, more relevant images will be retrieved with the sacrifice of precision.
- Independent of the feature sets used (C vs. CTE) and the number of images returned ($Sc = 20$ vs. $Sc = 180$), PP is the best in all $Pr(Re)$, $Pr(Sc)$ and $Re(Sc)$.
- Even though elegant in theory, MR performs poorly in most cases because its working conditions are not satisfied. More attentions should be paid on analyzing working conditions in future research.

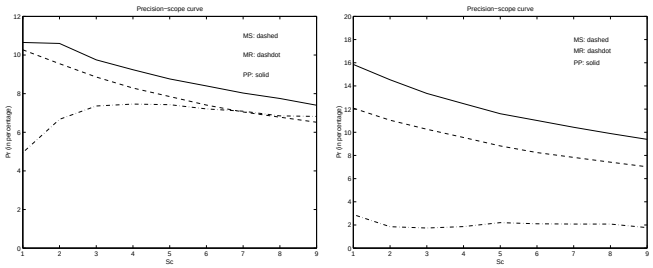


(a) (b)

Figure 2. **Precision-recall curve (a)Case C. (b)Case CTE.**

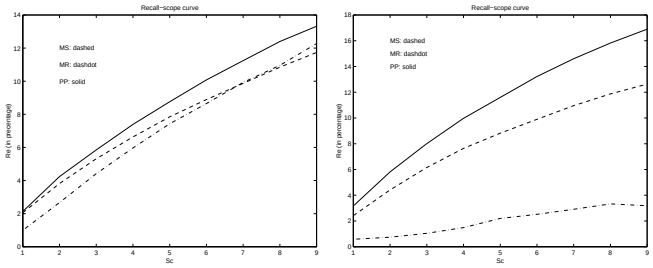
6. Discussions, Conclusions and Future Work

In Section 4, we used the generalized Euclidean distance for computing g_{ni} and linear combination for computing d_n. A natural thinking would be "how about choosing the generalized Euclidean distance to compute d_n as well?" That is, $d_n = \vec{g_n}^T U \vec{g_n}$, where U is an $(I \times I)$ matrix.



(a) (b)

Figure 3. **Precision-scope curve (a)Case C. (b)Case CTE.**



(a) (b)

Figure 4. **Recall-scope curve (a)Case C. (b)Case CTE.**

Indeed this formulation is more powerful to model non-linear (quadratic) relations in $\vec{g_n}$. Unfortunately, the objective function J of this formulation would then be a function of q_{ik}^4 and no *explicit* solutions can be derived. Optimal solutions for $\vec{q_i}$, W_i and U would only be obtained *iteratively*. This is extremely undesirable for image retrieval systems, because users need to wait for minutes before the iterative algorithm can converge. Being quadratic in g_{ni} and linear in d_n is the highest possible order for J to have *explicit* solutions. The flip side of the distance measure choices for g_{ni} and d_n is that for retrieval systems where "response time" is *not* a critical requirement, non-linear learning tools such as neural networks [11] and support vector machines [12] are worth exploring.

One thing worth pointing out is that the focus of this paper is not on finding the best visual features, but rather on exploring the best learning techniques. We are aware of sophisticated features including localized color and segmented shape [2]. We used less sophisticated features to obtain a bottom line for other systems to compare against. The proposed algorithm is an open framework and is ready to incorporate other more sophisticated features.

Vision and learning techniques are just some of the techniques that will make image retrieval successful. Other techniques, including information retrieval, database management and user interface, are also of crucial importance. However, these techniques, for example multi-dimensional indexing for faster search [2], are beyond the scope of this paper.

In conclusion, this paper developed a technique that gives optimized explicit solutions to hierarchical learning in image retrieval. Its image content model and adaptive W_i switching make it significantly outperform existing techniques. This has been demonstrated by the extensive experiments on a large heterogeneous image collection. However, there are still many dimensions to improve the current system. Both the low-level vision part (more sophisticated features [2]) and the learning part (more powerful tools [11, 12]) should continue to advance to meet users' true information needs.

7. Acknowledgment

The Corel data set of images were obtained from the Corel collection and used in accordance with their copyright statement.

References

[1] W. Niblack, R. Barber, and et al., "The QBIC project: Querying images by content using color, texture and shape," in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, Feb 1994.

[2] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," *Int. J. Vis. Commun. Image Rep.*, vol. 10, pp. 39–62.

[3] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill Book Company, 1982.

[4] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in *Proc. IEEE Int. Conf. on Image Proc.*, 1997.

[5] R. W. Picard, "Digital libraries: Meeting place for high-level and low-level vision," in *Proc. Asian Conf. on Comp. Vis.*, Dec. 1995.

[6] I. J. Cox, M. L. Miller, S. M. Omohundro, and P. N. Yianilos, "Target testing and the pichunter bayesian multimedia retrieval system," in *Advanced Digital Libraries Forum*, (Washington D.C.), May 1996.

[7] Y. Ishikawa, R. Subramanya, and C. Faloutsos, "Mindreader: Query databases through multiple examples," in *Proc. of the 24th VLDB Conference*, (New York), 1998.

[8] H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[9] S. X. Zhou, Y. Rui, and T. S. Huang, "Water-filling algorithm: A novel way for image feature extraction based on edge maps," in *Proc. IEEE Int. Conf. on Image Proc.*, 1999.

[10] J. Huang, S. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih, "Image indexing using color correlogram," in *Proc. IEEE Conf. on Comput. Vis. and Patt. Recog.*, 1997.

[11] S. Haykin, *Neural Networks: A Comprehensive Foundation*. Upper Saddle River, NJ: Prentice-Hall, 1999.

[12] J. Platt, "Fast training of support vector machines using sequential minimal optimization," *Advances in Kernel Methods - Support Vector Learning*, April 1999.