Relevance Feedback in Content-Based Image Retrieval: Bayesian Framework, Feature Subspaces, and Progressive Learning

Zhong Su, Hongjiang Zhang, Stan Li, and Shaoping Ma

***Abstract*—Research has been devoted in the past few years to relevance feedback as an effective solution to improve performance of content-based image retrieval (CBIR). In this paper, we propose a new feedback approach with progressive learning capability combined with a novel method for the feature subspace extraction. The proposed approach is based on a Bayesian classifier and treats positive and negative feedback examples with different strategies. Positive examples are used to estimate a Gaussian distribution that represents the desired images for a given query; while the negative examples are used to modify the ranking of the retrieved candidates. In addition, feature subspace is extracted and updated during the feedback process using a Principal Component Analysis (PCA) technique and based on user's feedback. That is, in addition to reducing the dimensionality of feature spaces, a proper subspace for each type of features is obtained in the feedback process to further improve the retrieval accuracy. Experiments demonstrate that the proposed method increases the retrieval speed, reduces the required memory and improves the retrieval accuracy significantly.**

***Index Terms*—Bayesian estimation, content-based image retrieval, principal component analysis (PCA), relevance feedback (RF).**

I. Introduction

CONTENT-BASED image retrieval (CBIR) is a process to find images similar in visual content to a given query from an image database. It is usually performed based on a comparison of low level features, such as color, texture or shape features, extracted from the images themselves. While there is much research effort addressing content-based image retrieval issues [1], [11], [19], the performance of content-based image retrieval methods are still limited, especially in the two aspects of retrieval accuracy and response time.

The limited retrieval accuracy is because of the big gap between semantic concepts and low-level image features, which is the biggest problem in content-based image retrieval. For example, for different queries, different types of features have different significance; an issue is how to derive a weighting scheme to balance the relative importance of different feature type and there is no universal formula for all queries. The relevance feedback technique can be used to bridge the gap [3], [12], [13], [17], [24]–[26].

Relevance feedback, originally developed for information retrieval [16], is a supervised learning technique used to improve the effectiveness of information retrieval systems. The main idea of relevance feedback is using positive and negative examples provided by the user to improve the system's performance. For a given query, the system first retrieves a list of ranked images according to predefined similarity metrics, which are often defined as the distance between feature vectors of images. Then, the user selects a set of positive and/or negative examples from the retrieved images, and the system subsequently refines the query and retrieves a new list of images. The key issue is how to incorporate positive and negative examples to refine the query and how to adjust the similarity measure according to the feedback.

The original relevance feedback method, in which the vector model [1], [20], [21] is used for document retrieval, can be illustrated by the Rocchio's formula [16] as

$$Q' = \alpha Q + \beta \left(\frac{1}{N_{R'}} \sum_{i \in D_{R}'} D_i \right) - \gamma \left(\frac{1}{N_{N'}} \sum_{i \in D_{N}'} D_i \right) \tag{1}$$

where α, β, and γ are suitable constants and $N_{R'}$ and $N_{N'}$ are the number of documents in D'_R and D'_N, respectively. That is, for a given initial query Q, and a set of relevant documents D'_R and nonrelevant documents D'_N given by the user, the optimal new query, Q', is the one that is moved toward positive example points and away from negative example points. This technique is also implemented in many content-based image retrieval systems [9], [12]. Experiments show that the retrieval performance can be improved considerably by using this approach. Generally speaking, previous relevance feedback methods can be classified into two approaches: "the weighing approach" and "the probability approach." Most existing work in content-based image retrieval uses the former approach.

The weighting method [9], [17], [19] associates larger weights with more important dimensions and smaller weights with less important ones. For example, [19] generalizes a relevance feedback framework of the low-level feature-based relevance feedback methods. An ideal query vector for each

Manuscript received July 2, 2001; revised March 26, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Christine Guillemot.

Z. Su is with the State Key Lab of Intelligent Tech. and Systems, Tsinghua University, Beijing 100084, China (e-mail: suzhong_bj@hotmail.com).

H. Zhang is with the Microsoft Research Asia, Beijing 100080, China (e-mail: hjzhang@microsoft.com).

S. Li is with the Microsoft Research Asia, 5F, Beijing Sigma Center, Beijing 100080, China (e-mail: szli@microsoft.com).

S. Ma is with the State Key Lab of Intelligent Tech. and Systems, Tsinghua University, Beijing 100084, China (e-mail: msp@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TIP.2003.815254

feature i is described by the weighted sum of all positive feedback images as

$$q_i = \frac{\pi^T Y_i}{\sum_{j=1}^{n} \pi_j} \tag{2}$$

where Y_i is the $n \times K_i$ (K_i is the length of feature i) training sample matrix for feature i obtained by stacking the n positive feedback training vectors X_i^+ into a matrix. The n element vector $\pi = [\pi_1, \ldots \pi_n]$ represents the degree of relevance (to the query) of each of the n positive feedback images, which can be determined by the user at each feedback iteration. The system then uses q_i as the optimal query to evaluate the relevance of the images in database. This strategy is widely used by many other image retrieval and relevance feedback systems [9], [17], [19].

Bayesian estimation methods have been used in the probabilistic approaches to relevance feedback. Cox *et al.* [3], Vasconcelos and Lippman [26], Meilhac and Nastar [13] all used Bayesian learning to incorporate user feedbacks to update the probability distribution of all the images in the database. They consider the feedback examples as a sequence of independent queries and try to minimize the retrieval error by Bayesian rules. That is, given a sequence of queries, they try to minimize the probability of retrieval error as

$$\begin{aligned} g(x) &= \arg\max_i P(y = i \,|\, x_1, \ldots, x_t) \\ &= \arg\max_i \{P(x_t \,|\, y = i\} \, P(y = i \,|\, x_1, \ldots, x_{t-1})\} \end{aligned} \tag{3}$$

where $\{x_1, \ldots, x_t\}$ is a sequence of queries (feedback examples) and $P(y = i \,|\, x_1, \ldots, x_t)$ is a prior belief about the ability of the ith image class to explain the queries.

Efforts have also been made to address the problem of slow response time in content-based image retrieval, the problem being caused mainly by the high dimensionality of the feature space, typically hundreds to thousands. Ng and Sedighian [14] made direct use of eigenimages, a method from face recognition [10], to carry out the dimension reduction. Faloutsos and Lin [6], Chandrasekaren *et al.* [2] and Brunelli and Mich [15] used principal component analysis (PCA) to perform the dimension reduction in feature spaces. Experimental results in these works show that most real image feature sets can be considerably reduced in dimension without significant degradation in retrieval quality. However, there are two problems with the use of PCA in these works. Firstly, they adopted a fixed number for the dimension size. This strategy is questionable because for images of different complexity, the intrinsic dimensions are usually different. Secondly, the subspaces are fixed once the PCA is performed the first time and do not adapt to users' subjectivity. Generally, this kind of blind dimension reduction can be dangerous, since information can be lost if the reduction is below the embedded dimension.

In this paper, we propose a new relevance feedback approach [24], [25], which is based on a Bayesian classifier and treats positive and negative feedback examples with different strategies. Not only can the retrieval performance be improved for the current user, but the improvements can also help subsequent users. Moreover, by applying the Principal Component Analysis (PCA) technique, the feature subspace is extracted and updated during the feedback process, so as to reduce the dimensionality of feature spaces, reduce noise contained in the original feature representation, and hence to define a proper subspace for each type of feature as implied in the feedback. These are performed according to positive feedbacks and hence consistently with the subjective image content.

The proposed new feedback algorithm uses a Gaussian estimator to incorporate positive examples in refining retrieval results. By assuming that all of the positive examples in one retrieval iteration belong to the same semantic class with common semantic object(s) or meaning(s) and the features from that semantic class follow a Gaussian distribution, we use features of all positive examples in a query iteration to calculate and update the parameters of its corresponding semantic Gaussian class. That is, the image retrieval becomes a process to estimate Gaussian parameters of a semantic class and the query refinement becomes a process of updating the Gaussian parameters. Then we use a Bayesian classifier to re-rank the images in the database. This process is progressive so that every feedback can have an impact on later retrieval processes. Experimental results show that such a feedback approach improves the retrieval accuracy significantly compared with previous methods.

Another objective of the work presented in this paper is to extract more effective, lower-dimensional features from the originally given ones, by constructing proper feature subspaces from the original spaces, to improve the retrieval performance in terms of speed, storage and accuracy. In this paper, we propose to use the PCA technique to derive a more effective, efficient and compact feature representation supervised by the relevance feedback process.

PCA is a statistical tool for data analysis [8]. It decorrelates second order moments corresponding to low frequencies, and identifies directions of principal variations in the data. We incorporate PCA into the relevance feedback framework to extract feature subspaces in order to represent the subjective class implied in the positive feedback examples. This leads to the following benefits: 1) whitening feature distributions so that distance metrics can be defined more rationally; 2) reducing possible noise contained in the original feature representation; 3) reducing dimensionality of feature spaces, and hence 4) defining a proper subspace for each type of feature, as implied in the feedback. As mentioned earlier, different types of features have different significance for different queries, and their subspaces should have varying dimensionalities. A procedure is provided to adjust the dimensionalities based on the evidences provided by the feedback.

Benefits brought about by the PCA-based extraction of feature subspaces are much lower storage, much faster speed and higher accuracy. Our analysis shows that the computational time and memory requirements in online retrieval are linear with the total feature dimension. When only about 30% of the original total dimensions are used, which is the case in our system, only 30% of the memory is needed and the proposed method is three times faster than of those methods without the PCA dimension reduction. In other words, by applying PCA dimension reduction, the retrieval system can afford nine times more images. Experiments on more than 10 000 Corel images show that such a speed-up is achieved without sacrifice in the retrieval accuracy.

TABLE I
LOW LEVEL FEATURES USED IN THE EVALUATION

Features	Dimension
Color histogram in HSV space with quantization 256	256
First and second color moments in *Lab* space	9
Color coherence vector in LUV space with quantization 64	128
Tamura coarseness histogram	10
Tamura directionality	8
Pyramid wavelet texture feature	24

This paper is organized as follows. Section II describes our method of relevance feedback based on Bayesian estimation. Section III first introduces the PCA process, followed by a detailed presentation on the Bayesian relevance feedback algorithm in PCA feature subspaces. Also, a complexity analysis of the algorithm is given in this section. The experimental results are shown in Section IV. The concluding remarks will be given in the final section.

II. RELEVENCE FEEDBACK BASED ON BAYESIAN ESTIMATION

In the proposed relevance feedback approach, positive and negative feedback examples are incorporated in the query refinement process with different strategies. To incorporate positive feedback in refining image retrieval, we assume that all of the positive examples in a feedback iteration belong to the same semantic class whose features follow a Gaussian distribution. Features of all positive examples are used to calculate and update the parameters of its corresponding semantic Gaussian class and we use a Bayesian classifier to re-rank the images in the database. To incorporate negative feedback examples, we apply a penalty function in calculating the final ranking of an image to the query image. That is, if an image is similar to a negative example, its rank will be lowed depending on the degree of the similarity to the negative example. The details of these two strategies are described in detail in this section.

A. Positive Feedbacks and Progressive Learning Process

The Gaussian density is often used for characterizing probability because of its computational tractability and the fact that it adequately models a large number of cases. Consider a vector x that obeys a Gaussian distribution; then, the probability density function is

$$p(x) = \frac{1}{(2\pi)^{d/2} \mid R_x \mid^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\bar{x})^T R_x^{-1}(x-\bar{x})} \tag{4}$$

where $R_x(x-\bar{x})$ is the covariance matrix of feature x and $|R_x|$ is used to normalize the probability density function.

Let us define some notations before going further. Denote the image database by D and let n be the total number of positive feedback examples for query image Q. Suppose that m types of features are used in the retrieval and so an image is represented by $\vec{x} = [\overrightarrow{x_1}, \ldots, \overrightarrow{x_i}, \ldots, \overrightarrow{x_m}]$, where $\overrightarrow{x_i}$ is the feature vector in the n_i-dimensional feature space S_i for type i features. We assume that each type of feature is Gaussian distributed, $\overrightarrow{x_i} \sim N(\bar{x}_i, R_i)$, where R_i (more generally R_x in (4)) is the $n_i \times n_i$ dimension covariance matrix and $\bar{x}_i$ (more generally, $\bar{x}$ in (4)) is the n_i dimensional mean vector. That is, we calculate a Gaussian distribution for each type of feature used to represent images in the image database. We use a diagonal matrix $\mathrm{diag}\{\sigma_i^2\}$ to represent the intra-component correlation, for simplicity. This is to assume that the feature components we use to represent visual content of an image are orthogonal and independent to each other. The practical reason of this simplification is that the intra-component correlation cannot be estimated practically and reliably, especially when there are not enough feedback examples. We understand this is a strong assumption. However, for the features we used in our system as listed in Table I, this assumption is appropriate.

Our idea about relevance feedback is the following: It is reasonable to assume that all the positive examples belong to the class of images containing the desired object or semantic meaning and the features of images belonging to the semantic classes obey the Gaussian distribution. The parameters for a semantic Gaussian class can be estimated using the feature vectors of all the positive examples. Hence, the image retrieval becomes a process of estimating the probability of belonging to a semantic class and the query refinement by relevance feedback becomes a process of updating the Gaussian distribution parameters. The log posterior probability that the feature vector x belongs to the semantic class c_i implied in the positive examples is estimated using the following Bayesian formulation:

$$\begin{aligned} g_i(x) &= \ln P(c_i \mid x) \propto \ln p(x \mid c_i) + \ln P(c_i) \\ &= -\frac{1}{2} \frac{(x-\bar{x}_i)^T}{\Sigma_i (x-\bar{x}_i)} + \ln P(c_i) + \mathrm{const}_i \end{aligned} \tag{5}$$

where const_i are fixed in one feedback iteration. Assuming that different feature types are independent of each other, the log posterior probability is calculated as the sum of the individual $g_i(x)$'s. Experiments show that the use of this posterior probability as the ranking metric improves the performance of relevance feedback in content-based image retrieval [17], [19], [26].

There are three parameters in the Bayesian estimator in (5): ε_i, Σ_i and the probability of the semantic class $P(c_i)$. They need to be updated when more positive examples are provided by the user through relevance feedback. Actually such a process could be considered as the combination of two Gaussian classes. So it is easy to get the updating process. Denote the current set of positive examples by U, in which each example is denoted by $\underline{u}$. In other words, there are totally $|U|$ positive examples in the current feedback iteration. The updating of the Gaussian parameters is performed as follows:

$$n' = n + |U| \tag{6}$$

$$\begin{aligned}\sigma'^2_i = n\sigma_i^2 &+ \frac{n|U|\bar{x}_i^2 - 2n\bar{x}_i \sum_{u\in U} u}{n+|U|} \\ &+ \sum_{u\in U} u^2 - \frac{(\sum_{u\in U} u)^2}{n+|U|}\end{aligned} \tag{7}$$

$$\bar{x}'_i = \frac{n\bar{x}_i + \sum_{u\in U} u}{n_+|U|} \tag{8}$$

where n and n' are the total number of positive examples accumulated before and after the current feedback iteration, respectively; $\sum_{u\in U} u$ represents the mean of the positive examples in the current iteration.

In the current implementation, the probability of a semantic class $P(c_i)$ is assumed equal for all semantic classes and remains constant in the relevance feedback process.

The following describes how positive feedback is performed.

1) Initialization of System:

1) Feature Normalization: This allows equal emphasis on all the feature components. For $\vec{x}_i$ the normalized vector is $\vec{x}'_i = [x'_{il}, \ldots, x'_{in_i}]$, where $x'_{i_m} = (x''_{i_m} - \bar{x}(x''_{i_m})/3\sigma(x_{i_m}))$ and $x''_{i_m} = (x_{i_m} - \min(x_{i_m})/\max(x_{i_m}) - \min(x_{i_m}))$. If x_{i_m} satisfies the Gaussian distribution, it is easy to prove that the probability of x'_{i_m} being in the range of $[-1, 1]$ is 99%.
2) Initialization: Initialize $\sigma_i = \mathbf{I}$ (identity matrix) and $\bar{x}_i = \overrightarrow{x_i}$, $n = 1$.

2) Retrieval and Feedback:

1) Update the retrieval parameters $\sigma_i, \bar{x}_i$, and n according to (6)–(8) using the information provided by the current set of positive examples U.
2) Distance Calculation: For each image $K, K \in D$, its distance d_i is calculated using (5) in the retrieval after the feedback, $d_i = -g(K)$. That is, the similarity of each image in the database to the refined query is determined by (5) based on the positive examples.
3) Sorting by distances and provide the new ranking list to the user.

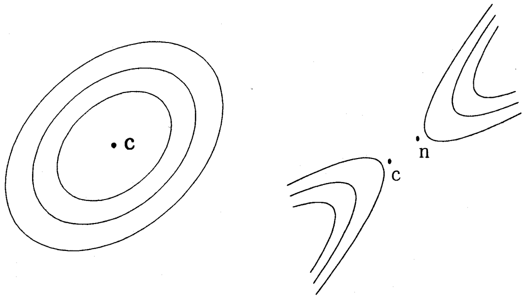


Fig. 1. The left figure shows the contours of the Bayesian classifier function in 2-D space, where "c" is the class distribution center. Right-hand side shows the effect of considering the negative examples as belonging to the same distribution. The contours of the classifier turn out to be a set of hyperbola in 2-D space, where "n" is the center of the negative examples.

Our feedback algorithm is incremental. This has the following advantages: Not only can the retrieval performance be improved for the current user, but the improvements can also help the subsequent users. The parameters of the Gaussian semantic class corresponding to each query will be refined by its positive examples according to (6)–(8), and could be used for other users. The updating process is an online process, so the computational complexity is quite important. However, the updating process only needs to deal with the current positive examples, so the computation cost can be neglected compared to the whole retrieval process.

B. Negative Feedbacks: "Dibbling" Process

Most methods previously described in the literature apply the same methodology to negative and positive examples, based on the assumption that the negative examples have the same feature distribution as the positive ones; or otherwise ignore the negative examples completely in the feedback process. In our opinion, negative examples should be treated differently. Positive examples are usually considered to belong to the same semantic class. However, according to many experiment observations, negative examples are often not semantically related. Moreover, as we have discussed above, the features of images from a certain semantic class can be considered as Gaussian distribution. If we consider the negative examples to be a Gaussian distribution as well and use the difference between the distance to the positive center and that to the negative one as the final distance measure[17], the contours of the classifier will change from closed ellipses to open hyperboloids in feature space, and this will induce a conflict between the original assumptions of the data distribution. Fig. 1 shows such a change in 2-D space.

Negative examples are often *isolated and independent*. Thus they need to be treated differently from the positive examples. We apply the following method to deal with the negative examples. We only "punish" those images in the database that are very near to the negative samples and do not let the negative samples influence the other images. Under this strategy, we penalize images near the negative examples by increasing their distance to the query. Such a *penalty* function seems like a 'dibbling' process in feature space. By extensive simulation, we have found that the penalty function can be approximated by

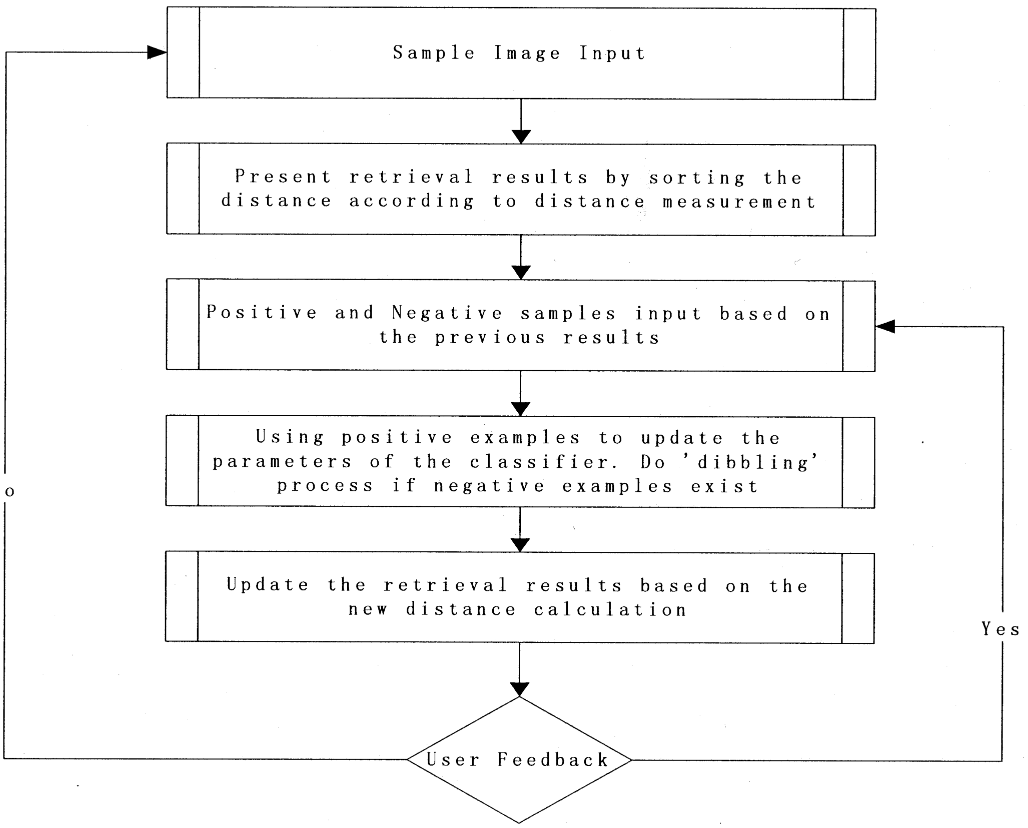

Sample Image Input
Present retrieval results by sorting the distance according to distance measurement
Positive and Negative samples input based on the previous results
Using positive examples to update the parameters of the classifier. Do 'dibbling' process if negative examples exist
Update the retrieval results based on the new distance calculation
User Feedback
No
Yes

Fig. 2. Flow chart of the whole feedback process.

a Gaussian function for each negative example. Denote the current set of negative examples by V, for each image $K, K \in D$ as before, the distance punish function is defined as

$$\mathrm{Pun}(d_i) = \sum_{v \in V} g_v(d_v) \tag{9}$$

where $g_v(d_v)$ is the Gaussian function whose parameters are determined experimentally; and d_v is the distance between image K and a negative example, $v, v \in V$. d_v can be calculated using the Euclidian distance between the feature vectors of image K and the negative example. $\Sigma_{v \in V}$ sums up penalty contributions from all negative examples to the image K. That is, if an image in the database is close to all negative examples, the penalty is high; in contrast, if the image is far away from all negative examples, the penalty function will decreased to zero, according to the Gaussian distribution.

So the distance after negative feedback is defined as

$$d'_i = d_i + \mathrm{Pun}(d_i). \tag{10}$$

That is, if an image in the database is close to negative examples, its distance to the query is increased by $\mathrm{Pun}(d_i)$ as defined in (10).

Fig. 2 shows the whole feedback process of using both the positive and negative ones.

III. Bayesian Relevence Feedback in the PCA Feature Subspaces

The other major contribution in the proposed relevance feedback approach to content-based image retrieval is to apply the principal component analysis technique to select and updated a proper feature subspace during the feedback process. This algorithm extracts more effective, lower-dimensional features from the originally given ones, by constructing proper feature subspaces from the original spaces, to improve the retrieval performance in terms of speed, storage requirement and accuracy. In this section, we first present the PCA algorithm, followed by a detailed description of how we apply PCA in relevance feedback in content-based image retrieval.

A. Principal Component Analysis

Consider an ensemble of n-dimensional vectors $\{\boldsymbol{x} = [x_1, \ldots, x_n]^T\}$ whose distribution is centered at the origin $E(x) = 0$. The covariance between each pair of variable is $r_{ij} = E\{(x_i - \bar{x}_i)(x_j - \bar{x}_j)\} = E\{x_i x_j\}$, where E is the expectation operator. The parameters r_{ij} can be arranged to form the $n \times n$ covariance matrix

$$R_x = E\{(x - \bar{x})(x - \bar{x})^T\} = E\{xx^T\}. \tag{11}$$

Assuming $\det(R_x) \neq 0$, then by applying eigenvector decomposition, R_x can be decomposed into the product of three matrices

$$R_x = W \Lambda W^{-1} \tag{12}$$

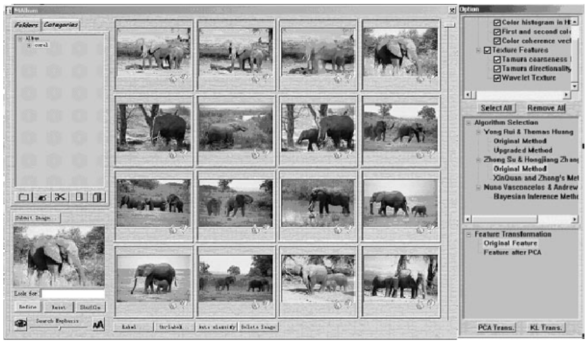


Fig. 3. Retrieval and feedback interface of *MiAlbum*. Example image is shown on the bottom left.

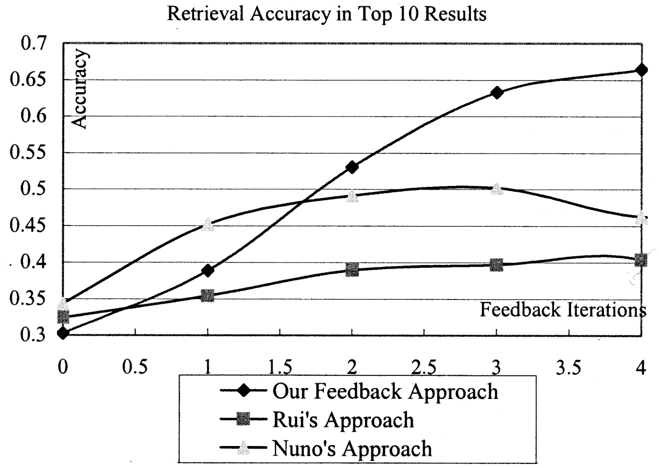


Fig. 4. Retrieval accuracy for top ten results in original feature space.

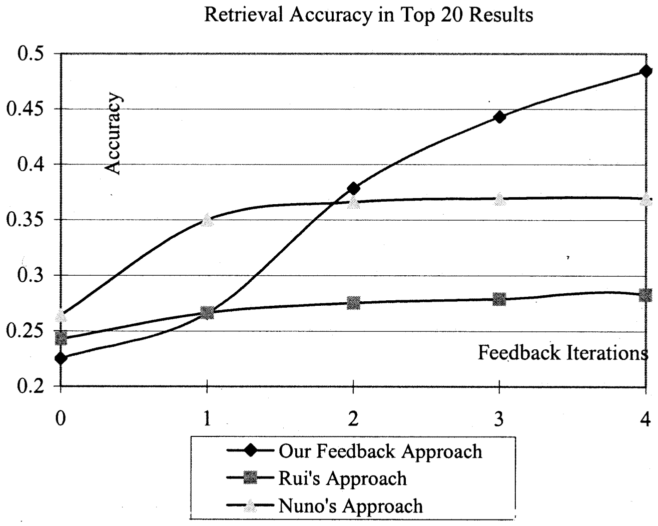


Fig. 5. Retrieval accuracy for top 20 results in original feature space.

where $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_n\}$ are the eigenvalues and $\boldsymbol{W} = [w_1, \ldots, w_n]^T$ are the corresponding eigenvectors. W is orthogonal in that $\boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}$. So the columns of $\boldsymbol{W}$ form a new orthogonal basis that is a linear transformation from of the original basis.

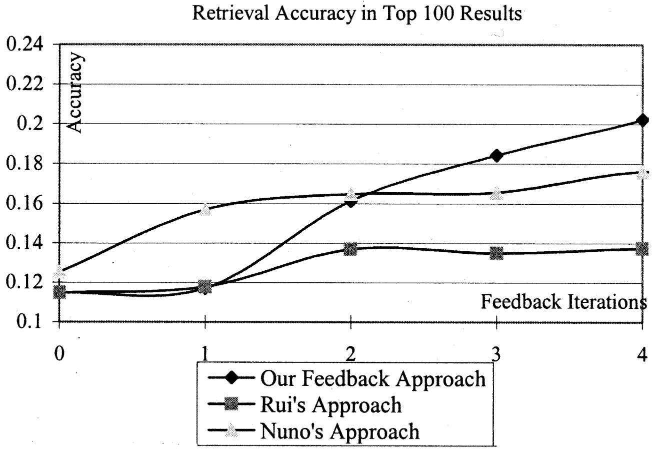


Fig. 6. Retrieval accuracy for top 100 results in original feature space.

The eigenvector decomposition can be used to whiten the feature distributions as follows. Project the original feature vectors $\boldsymbol{x}$ onto the eigenvector basis (without dimension reduction), obtaining the coordinates x, which is equivalent to rotating the feature basis; then rescale the coordinates by the factor of $1/\sqrt{\lambda_j}$ to obtain the whitened feature vector $\boldsymbol{y}$ and $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x}$. After the whitening, we are able to calculate the Mahalanobis distance between $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ in the original feature space by the simple Euclidean distance between the corresponding $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$, in the whitened feature space, i.e.,

$$\mathrm{dist}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{(\boldsymbol{x}_1 - \boldsymbol{x}_2)^T}{\sum(\boldsymbol{x}_1 - \boldsymbol{x}_2)} = \|\boldsymbol{y}_1 - \boldsymbol{y}_2\|. \tag{13}$$

If we only select the first m eigenvectors as the orthonormal basis vectors to form a subspace $\boldsymbol{L} = \mathrm{span}\ (\boldsymbol{W}')$, then any vector $\boldsymbol{x}$ in the original space can be linearly transformed to $\boldsymbol{L}$ with the new representation $\boldsymbol{y}'$

$$\boldsymbol{y}' = \boldsymbol{W}'\boldsymbol{x}. \tag{14}$$

An approximation to the original $\boldsymbol{x}$ can be reconstructed from the projection $\boldsymbol{y}'$ as $\boldsymbol{x}' = \boldsymbol{W}'^{T}_{\boldsymbol{y}}{}' = \boldsymbol{W}'^T\boldsymbol{W}'x$. The mean squared reconstruction error is

$$J_e(m) = E\{\|\boldsymbol{x} - \boldsymbol{x}'\|^2\} = \sum_{i=m+1}^{n} \lambda_i. \tag{15}$$

We can choose the set of eigenvectors used for the reconstruction to minimize this: Sort eigen values in descending order so that $\lambda_i > \lambda_j > 0$, where $i > j$; this also sorts the corresponding eigenvectors in the descending order of their significance. The mean square reconstruction error J_e can thus be minimized.

There are two advantages of using PCA: 1) dimension reduction is achieved when $k < n$ and x is represented by the projected coefficients; 2) noise reduction is achieved because as indicated in our experiments that high frequency components corresponding the smallest eigenvalues often correspond noises in image retrieval applications; thus, removing these high frequency components in PCA process will reduce the noise in retrieval.

An issue should be mentioned here: Addition of new images may require re-learning of the PCA basis vectors. For retrieval systems that entirely rely on low-level image features, adding new images simply involves extracting various feature vectors

TABLE II
RETRIEVAL ACCURACY AFTER NEGATIVE FEEDBACK

	Our Method	Previous method
Accuracy Increase in Top 10	13.4%	2.1%
Accuracy Increase in Top 20	7.8%	0.1%
Accuracy Increase in Top 100	2.6%	-0.1%

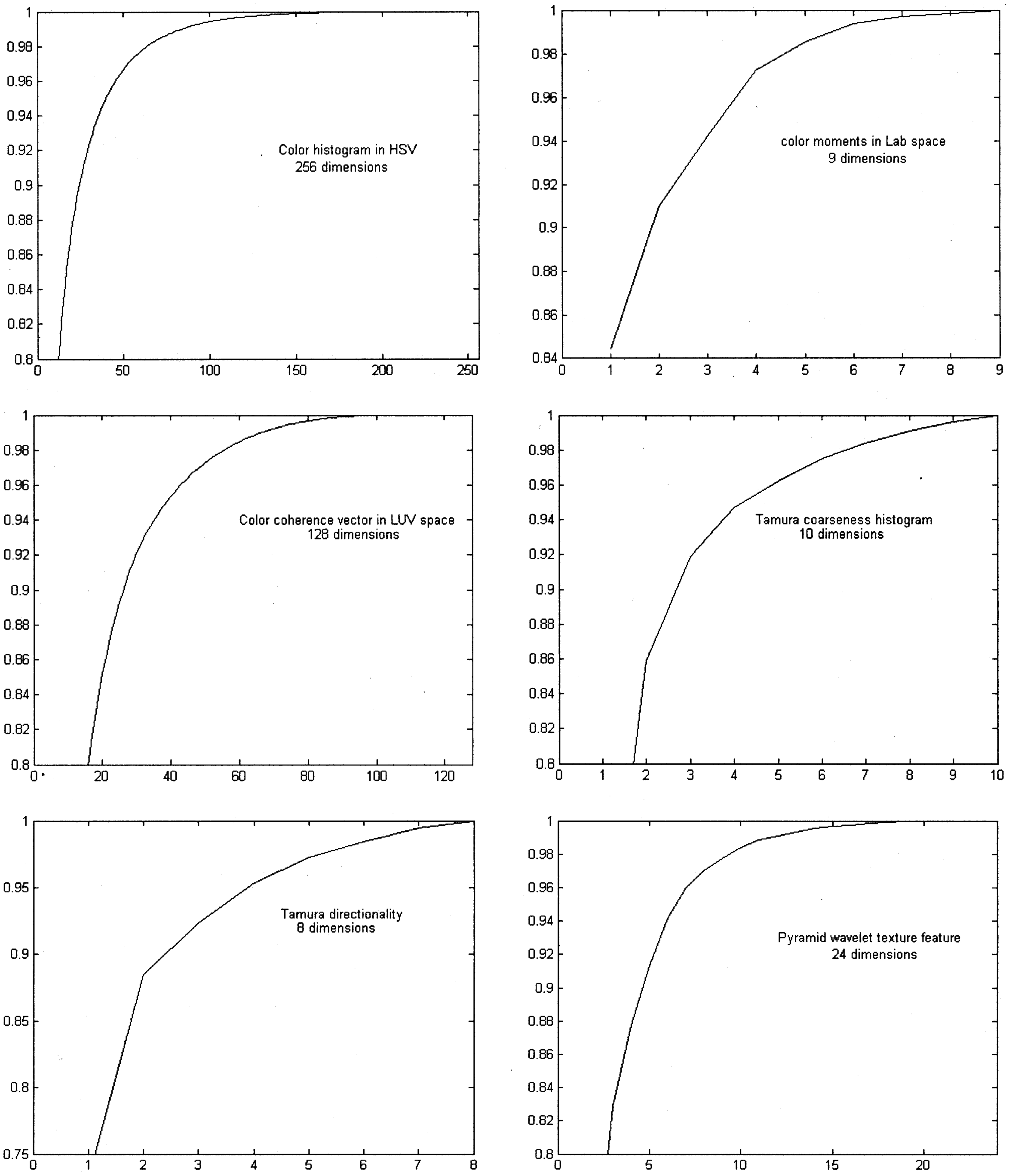


Fig. 7. Accumulate eigenvalues of individual low-level features after PCA process.

for the set of new images. However, the insertion of new images may need some processing in the approach proposed here. This is because PCA subspaces need to be re-learned if new images significantly change the covariance matrix.

TABLE II
RETRIEVAL ACCURACY AFTER NEGATIVE FEEDBACK

	Our Method	Previous method
Accuracy Increase in Top 10	13.4%	2.1%
Accuracy Increase in Top 20	7.8%	0.1%
Accuracy Increase in Top 100	2.6%	-0.1%

B. Relevance Feedback in PCA Feature Subspaces

As described in the last subsection, PCA can be used to reduce the dimensionality of the feature space, and to extract the principal lower-dimensional subspace of the original feature space. A reasonable deduction in dimensionality causes little decrease in performance. This is especially true in content-based image retrieval since the components removed from the original image feature space often correspond to noise. According to our experimental results on a large amount of data, dropping 80% of the feature dimensions leads to only about 5% reconstruction error; dropping 90% dimensions gives only about 10% reconstruction error. Yet the retrieval speed has been improved significantly as a result of such dimension reductions.

A key issue is how to determine the intrinsic dimensionality of the feature subspace for each feature, given that the rate of dimension reduction should vary for different types of features. We propose to use the following idea to perform the PCA in the feedback framework.

In the first round of retrieval, we use only a few significant components for each feature. That is, for each feature type i, select the first m_i most significant eigenvectors for the type i feature, where m_i is quite small comparing with its original feature dimension n_i. The initial value of m_i of the type of feature is calculated by: $m_i = \psi \times n_i$ where ψ is the dimension reduction rate; e.g., $\psi = 0.1$. In other words, for simplicity, the PCA process in our implementation is performed for each feature type, instead of all features together. It will be a much more complicated process to perform the PCA process for all types of features concatenated together. Also, as the Gaussian class discussed in Section 2 is also defined for each type of feature, it is much easier to embed the PCA process in the relevance feedback and retrieval process. Therefore, instead of doing PCA for all features together, we introduce the following scheme for the determination of the dimensionality of feature subspaces for the subsequent iterations after initialization.

Let the dimension reduction rate initially set to be the same, e.g., $\psi = 0.1$, for all features before the relevance feedback begins. As iterative relevance feedback continues, the dimensions to be retained for good feature types are increased, while the number of dimensions for bad feature types is decreased. This is achieved by adjust m_i according to a goodness measure for a feature type as follows.

As before, let Q be the query, $K(K \in D)$ be an image in the image database and the current set of positive examples be U, then the evaluation measurement can be defined as follows:

$$M_i = \frac{\sum_{K_i \in D} \frac{d(K_i,Q)}{|D|}}{\sum_{u \in U} \frac{d(u,Q)}{|U|}} \qquad (16)$$

where $d(\ )$ refers to the distance between two features in the PCA subspace. We can see from the above equation that the *good* feature will have small distances from its positive examples to the query class center compared to other images in databases.

After calculating M_i for each feature type i, we sort the M_i in descending order to get the rank r_i of M_i. This rank can reflect the effectiveness of the feature in the current retrieval session relative to the others. It will be high for a good feature type and low for a bad one. So we can update the number of dimensions, m_i of feature type i based on the calculated rank value as

$$m'_i = \left(m_i + \tau \times \left(\left[\frac{m}{2}\right] - r_i - 1\right)\right) \qquad (17)$$

where τ is a constant factor, [ ] is the floor operation, and m is the total number of feature types and $m > 1$. From the above equation we can see that for a good feature, its corresponding m_i will be increased so that more dimensionality could be availed of in the next retrieval iteration. The choice of τ depends on the number of feature types, the original dimension of each feature type, and the desired increment scale. In our experiment, we set τ to 1 and $m = 6$. That is, if the feature type is ranked higher than 3, then its dimension number will be increased; otherwise, it will be lowered.

The following describes the PCA embedded algorithm.

1) Initialization of System: For each feature type $i = 1, \ldots, m$, do the following:

1) Perform PCA on all the images in the original feature space S_i, obtaining the eigenvalues and the corresponding eigenvectors calculated by (11) and (12). Sort the eigenvectors in the order of descending eigenvalues.
2) Whiten the distribution of the feature vectors by first rotating the feature space and then dividing the coordinate by the square root of the corresponding eigenvalues (refer to Section 3.1). Such pre-processing does not lose any information because the dimensionalities are not reduced in the process.
3) Initialize the parameters of $\sigma_i \varepsilon_i$ and n as before.

The following describes how to update the retrieval and feedback process.

2) Retrieval and Feedback:

1) Update the parameter of the Gaussian class according to (6)–(8).
2) Calculate M_i according to (16), where $\mathrm{i} = 1, \ldots, \mathrm{m}$, sorting the M_i in decreasing order to get the rank r_i of M_i.
3) Update the m_i according (17).

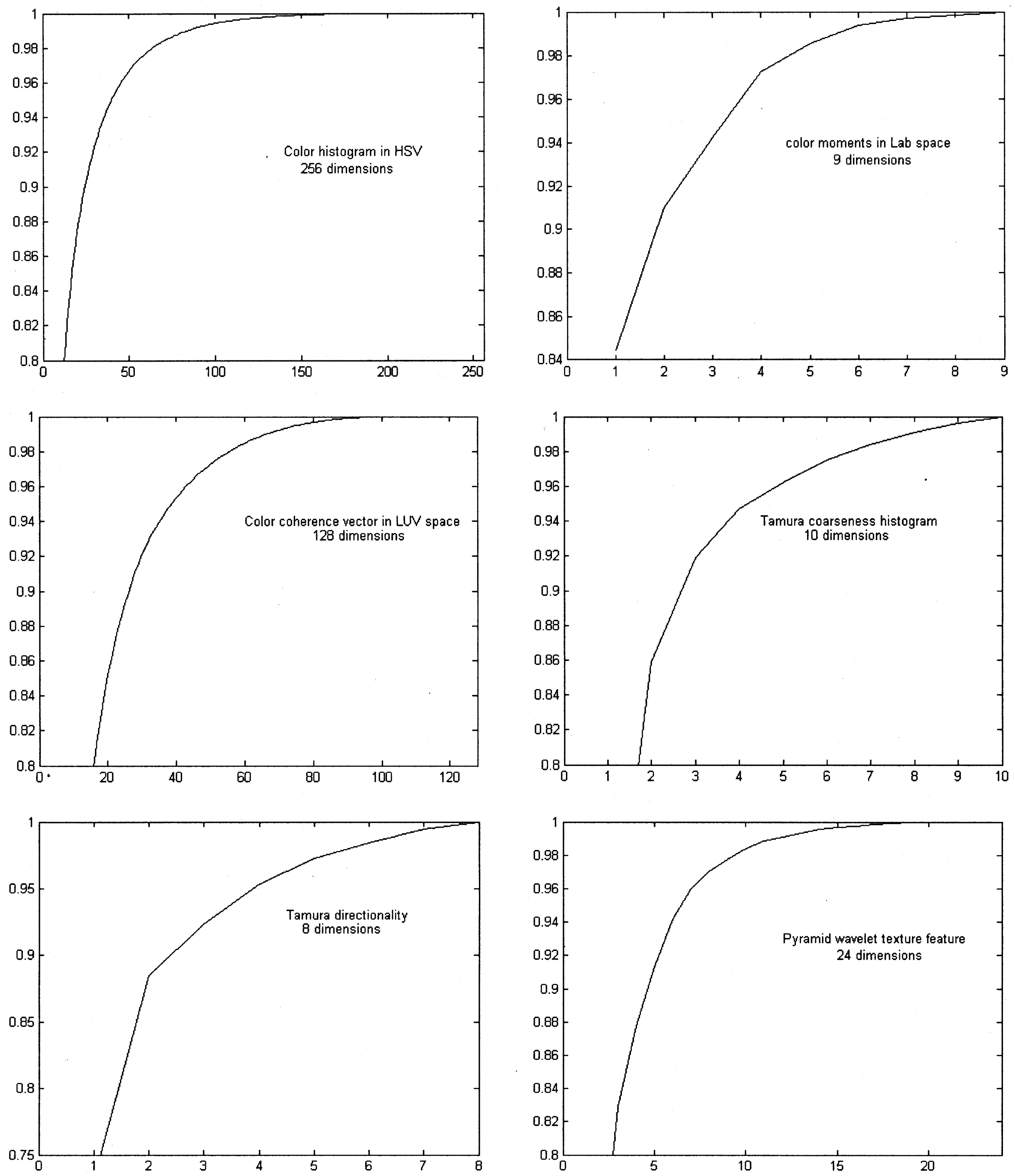


Fig. 7. Accumulate eigenvalues of individual low-level features after PCA process.

4) Calculate the distance between all images in the database D and the current query Q in the m_i dimensional feature subspace.
5) Sort by distances and provide the new ranking list to the user.

The differences between the current retrieval and relevance feedback method and that described in Section II are the following.

1) The retrieval and feedback are performed based on the features in the extracted feature subspaces. For feature type i, we only use the first m_i dimensions to perform the retrieval and feedback process.
2) In addition to updating the parameters of Gaussian class as before (this costs little because the number of positive examples is usually small), the m_i are also updated so that better features will have more impact than bad ones.

In this way, the system learns from the user's feedback in a feature-based manner. The dimensionalities of the individual feature subspaces are adjusted dynamically according to the retrieval performance of the corresponding features in the current feedback session.

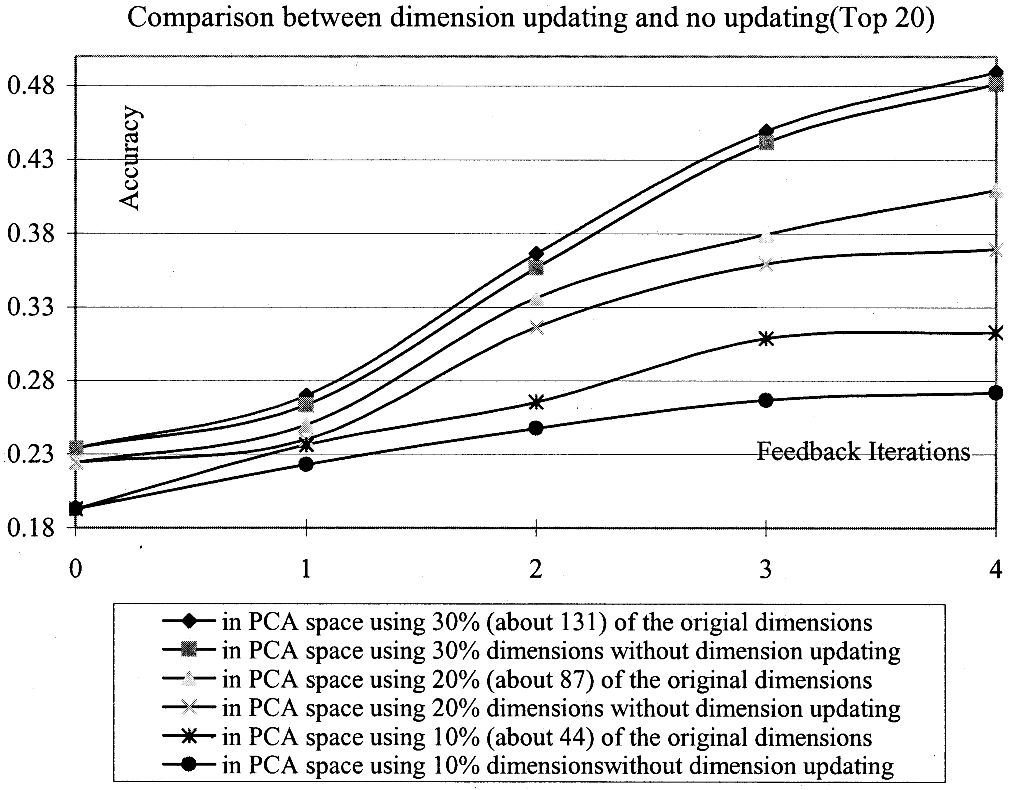


Fig. 8. Comparing the retrieval accuracy of top 20 results in PCA feature space between the dimension updating and no dimension updating method.

C. Complexity Analysis of the PCA Subspace Relevance Feedback Algorithm

In this subsection, we briefly analyze the complexity of the proposed PCA subspace relevance feedback method and show that the proposed method can save memory and speed up the computation significantly.

The requirements for memory consist of two parts: the basis vectors and the feature coordinates in the feature subspaces. The size of the former part is fixed regardless of the database size and hence can be ignored when the database is large. Therefore, the memory requirement depends on the total number of images in the database, which in turn depends, linearly, on the total dimension of the feature vectors, as we need to load the feature vector of each image in the database, at least one at a time, when computing the similarities. The computation process of covariance matrix of $\boldsymbol{R}_x$ in The PCA learning is very time consuming, which is O $(\text{Total\_image\_number} * n_i * n_i)$, where n_i is the feature dimensions of feature i. Fortunately, this process is an offline procedure.

The biggest bottleneck of online retrieval is the distance calculation. The computation complexity of distance measure is $O(N)$ where N is the number of the vector dimension. For example, the Euclidean distance needs $O(N)$ multiplication and $O(N)$ addition operation, which is the same as our method. Therefore, the total computation complexity in the online retrieval is

$$\sum_{i=1}^{m} O(\text{Total\_image\_number} \times n_i)$$
$$= O(\text{Total\_image\_number} \times N), \quad \text{where } N = \sum_{i=1}^{m} n_i.$$

Therefore, the time and memory costs the relevance feedback process are linear with the total feature dimension. That is, if we use only about $\psi = 10\%$ of the original total dimensions, only 10% of the memory is required and the speed of a PCA method will be nine times faster than methods without PCA dimension reduction. This clearly shows the advantage of using PCA in the relevance feedback process.

IV. Experimental Results

A. Test Data and Features

The image set we used in our evaluation is the Corel Image Gallery. 10 000 images of 79 semantic categories are selected to calculate the performance statistics, of which 80% images are used for learning PCA basis vectors and 20% for testing. Whether a retrieved image is correct or incorrect is judged according to the ground truth class.

Three types of color features and three types of texture features are used in our system, as shown in Table I. The total dimension is 435.

B. Retrieval Interface

The *MiAlbum* image retrieval system implements the framework discussed in this paper. It is an image retrieval system for PC users. In this paper, we only focus on retrieval based on low-level features. So search by example is the interaction mode we focus on here. However, the *MiAlbum* system also supports other two modes of interaction: keyword-based search and browsing the entire image database.

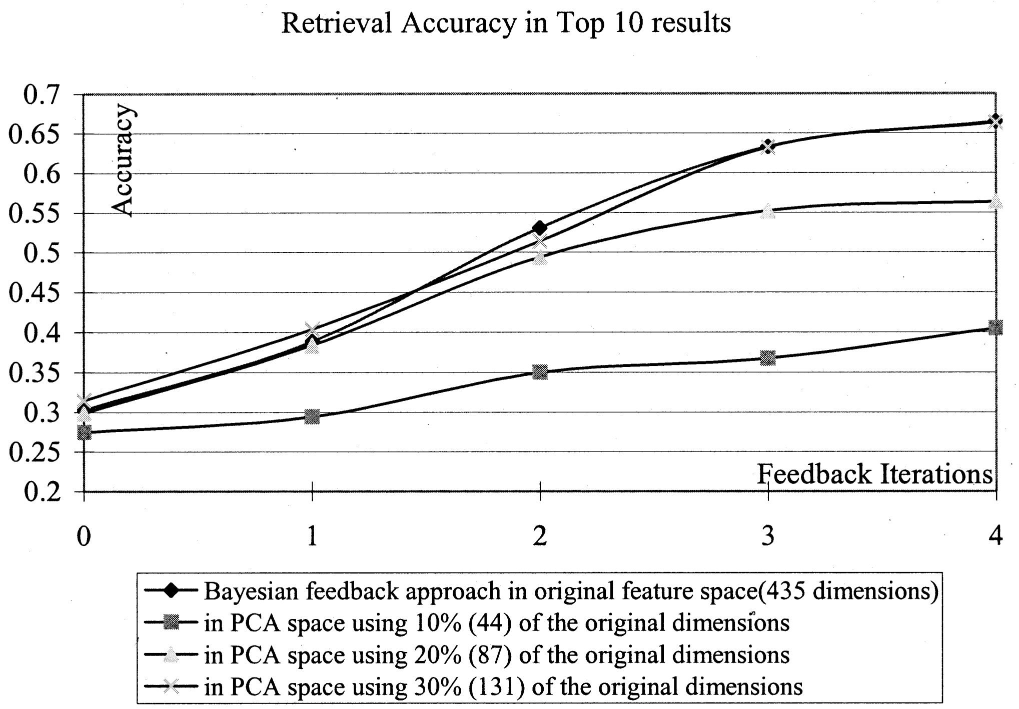


Fig. 9. Comparing the retrieval accuracy in top ten results between the original feature space and PCA feature space.

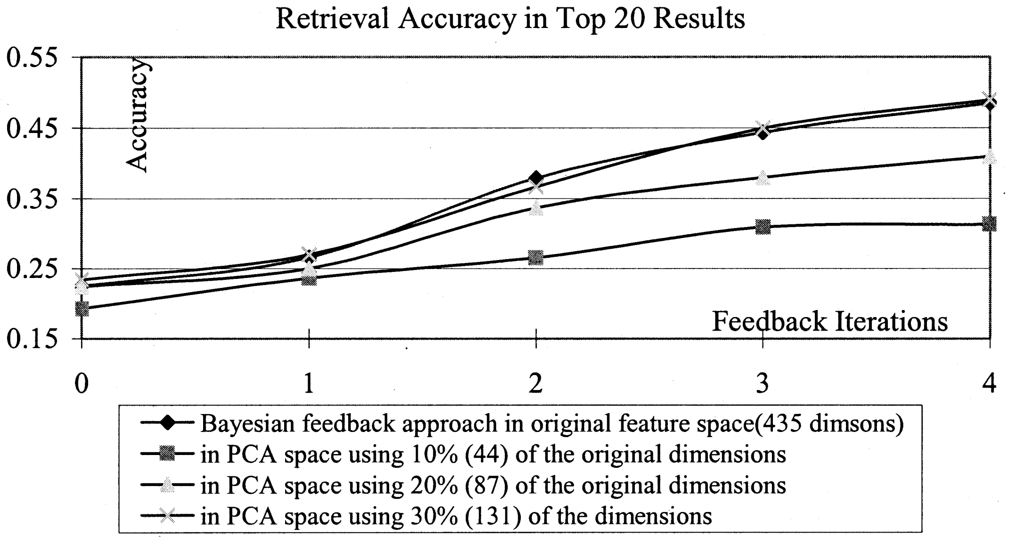


Fig. 10. Comparing the retrieval accuracy in top 20 results between the original feature space and PCA feature space.

The main user interface is shown in Fig. 3. It has a simple interface to search images by examples. Users can select the example image from the database or from the file system. Results are returned as a ranked list show in the right image view. The ranking sequence is from left to right and from top to bottom. During the retrieval process user can provide relevance feedback by clicking the "Yes" or "No" icon below the results. As we can see from Fig. 3, there is an Option Dialog in the main interface. The user can select the features and methods, as well as the choice between original features space and PCA feature space.

C. Experimental Results

The feedback process in the evaluation is carried out as follows. Given a query example from the test set, one different test image of the same category as the query is used in each round of feedback iteration as the positive example for updating the Gaussian parameters. For the negative feedback process, the first two irrelevant images are assigned as negative examples. We only pick one positive example at each iteration because we want to illustrate the worst case scenarios in which the semantic representation power of image features is low such that the positive examples at each iteration is rare. Also, we only

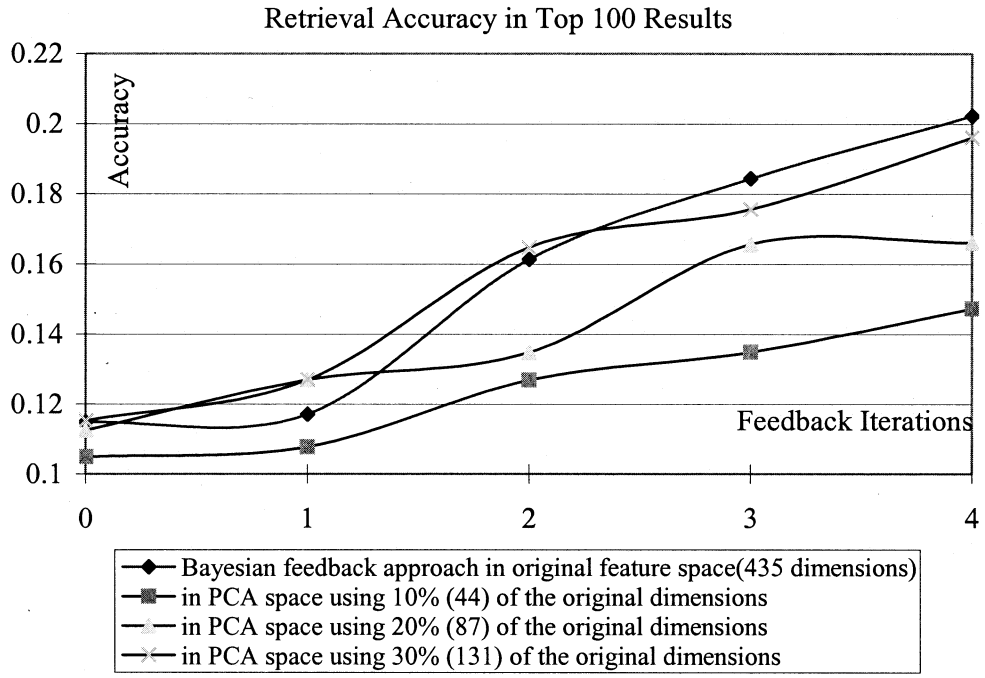


Fig. 11. Comparing the retrieval accuracy in top 100 results between the original feature space and PCA feature space.

consider two negative examples at each iteration because we assume users of image retrieval systems are impatient in general and not willing to pick up more negative examples.

The accuracy is defined as

$$\text{Accuracy} = \frac{\text{relevant images retrieved in top}\,T\,\text{returns}}{T}. \tag{18}$$

A number of experiments have been performed as follows. For all the experiments, the accuracy numbers are averaged over all test queries, which is totally 20% of the image database of 20 000 images.

Firstly, our Bayesian feedback scheme is compared with previous feedback approaches presented by Nuno [26] and Rui [17], [19]. This comparison is done in the original feature space. Figs. 4, 5 and 6 show that the accuracy of our Bayesian feedback method becomes higher than the other two methods after two feedback iterations. This demonstrates that the incorporated Bayesian estimation with the Gaussian parameter-updating scheme can improve the performance of image retrieval with relevance feedback.

Table II shows the experimental results after one iteration of negative feedback. Clearly, our strategy on negative feedback increases the retrieval accuracy much more than the previous method [17].

Fig. 7 demonstrates the fidelity of the reconstruction, which can be defined as $\sum_{i=1}^{m} \lambda_i$ (see Section III-A), as a function of retained dimension m_i, for the six types of features. It is shown that significant dimension reductions cause little reconstruction errors, for all the six cases: Dropping 80% of the dimensions causes only about 5% reconstruction error; cutting off 90% of the dimensions causes about 10% of loss. The discarded information can be due to noise because it corresponds to the least significant or high frequency components. However, keeping the reconstruction fidelity fixed, the minimum dimension for an individual subspace can vary when the type of the features is different, as can be seen from the differences of the six plots. Also, the dimensions should be adjusted dynamically according to the evidence gathered thus far. Therefore, the PCA dimension reduction must be completed by an appropriate method for the dynamic adjustment of subspace dimensions in order to achieve significant reduction with little loss of accuracy.

Next, we compare the results obtained with and without the dynamic adjustment of the dimensionalities of feature subspaces. As we can see from Fig. 8, the retrieval accuracy can be improved by the dynamic adjustment scheme, especially when the feature dimensions are significantly reduced, e.g., lower than 30% of the original dimensions. As the final dimension resulted from each query image after the dynamic dimension updating varies according to (17), the final dimension numbers listed in Fig. 8 are for references.

Finally, the accuracy after PCA reduction is compared with that obtained in the original feature space. The results in Figs. 9–11 show that the accuracy can be maintained when the retained total dimension is 30% of the original or above; the benefit brought about by this reduction is that the retrieval speed is tripled and that the memory requirement is one third of the original.

V. Conclusion

In this paper, we have presented a new relevance feedback approach to content-based image retrieval by integrating a feature subspace extraction process into a Bayesian feedback process. Principal Component Analysis is used to reduce feature subspace dimensionalities. When multiple types of features (e.g., color and texture) are used, a method is proposed to adjust the subspace dimensionality for each type of feature, based on evidence obtained, to account for differences between individual feature subspaces as reflected in the recent feedback. This dynamic dimension adjusting method is especially effective when the feature dimensions are significantly reduced, e.g., lower than 30% of the original dimensions. The feedback process plays two roles: providing information for updating the Gaussian parameters in the Bayesian feedback, and providing evidence for the adjustment of feature subspace dimensionalities. Experimental results show that the proposed method can significantly improve the retrieval performance in speed, memory and the accuracy. In principle, the proposed feature subspace extraction method can be incorporated in any other content-based retrieval methods to save memory and to speed-up computation.

ACKNOWLEDGMENT

The work presented in this paper was performed at Microsoft Research Asia. The authors thank Y. Sun and C. Hu of MSRC Software Engineer Group for their great help in system work. They also thank J. Shen, M. Li, and W.-Y. Liu, who have gave them valuable comments and suggestions for this paper.

REFERENCES

[1] C. Buckley and G. Salton, “Optimization of relevance feedback weights,” in *Proc. SIGIR’95*.

[2] S. Chandrasekaran *et al.*, “An eigenspace update algorithm for image analysis,” *CVGIP: Graph. Models Image Process. J.*, 1997.

[3] I. J. Cox, T. P. Minka, T. V. Papathomas, and P. N. Yianilos, “The Bayesian image retrieval system, pichunter: Theory, implementation, and psychophysical experiments,” *IEEE Trans. Image Processing—Special Issue on Digital Libraries*, 2000.

[4] I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks, Theory and Applications*. New York: Wiley, 1996.

[5] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York, NY: Wiley, 1973.

[6] C. Faloutsos and K. Lin, “Fastmap: A fast algorithm for indexing, data-mining and visualization of traditional and multimedia,” in *Proc. SIGMOD*, 1995, pp. 163–174.

[7] M. Flickner *et al.*, “Query by image and video content: The QBIC system,” *Computer*, vol. 28, pp. 23–32, 1995.

[8] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.

[9] Y. Ishikawa, R. Subramanya, and C. Faloutsos, “Mindreader: Query databases through multiple examples,” in *Proc. 24th VLDB Conf.*, New York, 1998.

[10] M. Kirby and L. Sirovich, “Application of the Karhunen-Loeve procedure for the characterization of human faces,” *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 103–108, Jan. 1990.

[11] C. Lee, W. Y. Ma, and H. J. Zhang, “Information Embedding Based on User’s Relevance Feedback for Image Retrieval,” HP Labs, Tech. Rep., 1998.

[12] Y. Lu, C. Hu, X. Zhu, H. Zhang, and Q. Yang, “A unified framework for semantics and feature based relevance feedback in image retrieval systems,” in *Proc. 8th ACM Multimedia Int. Conf.*, Los Angeles, CA, Nov. 2000.

[13] C. Meilhac and C. Nastar, “Relevance feedback and category search in image databases,” in *IEEE Int. Conf. Multimedia Computing and Systems*, 1999.

[14] R. Ng and A. Sedighian, “Evaluating multi-dimensional indexing structures for images transformed by principal component analysis,” in *Proc. SPIE Storage and Retrieval for Image and Video Databases*, 1996.

[15] B. Roberto and M. Ornella, “Image retrieval by examples,” *IEEE Trans. Multimedia*, vol. 2, Sept. 2000.

[16] J. J. Rocchio Jr., “Relevance feedback in information retrieval,” in *The SMART Retrieval System: Experiments in Automatic Document Processing*, G. Salton, Ed. Englewood Cliffs, NJ: Prentice-Hall, 1971, pp. 313–323.

[17] Y. Rui and T. S. Huang, “Relevance feedback: A power tool for interactive content-based image retrieval,” *IEEE Circuits Syst. Video Technol.*, vol. 8, no. 5, 1999.

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

[19] Y. Rui and T. S. Huang, “A novel relevance feedback technique in image retrieval,” *ACM Multimedia*, 1999.

[20] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.

[21] W. M. Shaw, “Term-relevance computation and perfect retrieval performance,” *Inform. Process. Manage*, vol. 31, pp. 491–498, 1995.

[22] G. Sheikholeslami, W. Chang, and A. Zhang, “Semantic clustering and querying on heterogeneous features for visual data,” in *Proc. 6th ACM Int. Multimedia Conf. (ACM MULTIMEDIA’98)*, Bristol, U.K., Sept. 1998.

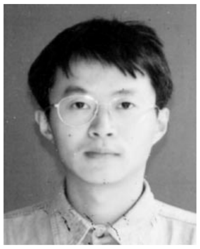
[23] H. S. Stone and C. S. Li, “Image matching by means of intensity and texture matching in the Fourier domain,” *Proc. IEEE Int. Conf. Image Processing*, Oct. 1997.

[24] Z. Su, H. Zhang, and S. Ma, “Using Bayesian classifier in relevant feedback of image retrieval,” in *IEEE Int. Conf. Tools with Artificial Intelligence*, Vancouver, BC, Canada, Nov. 2000.

[25] ——, “Relevant feedback using a Bayesian classifier in content-based image retrieval,” in *Proc. SPIE Electronic Imaging 2001*, San Jose, CA, Jan. 2001.

[26] N. Vasconcelos and A. Lippman, “Learning from user feedback in image retrieval systems,” in *Proc. NIPS’99*, Denver, CO, 1999.

[27] L. Zhu and A. Zhang, “Supporting multi-example image queries in image databases,” in *IEEE Int. Conf. Multimedia and Expo*, New York, July.



Zhong Su received the B.E. and Ph.D. degrees from the Department of Computer Science and Technology, Tsinghua University, China, in 1998 and 2002, respectively.

During 1999–2001, he was a Research Intern at the Media Computing Group, Microsoft Research Asia, Beijing, China. He is now a Research Staff at IBM China Research Lab. His currently research interests include knowledge management, information retrieval, and pattern recognition.

Hongjiang Zhang received the Ph.D. degree from the Technical University of Denmark and the B.S. degree from Zhengzhou University, China, both in electrical engineering, in 1982 and 1991, respectively.

From 1992 to 1995, he was with the Institute of Systems Science, National University of Singapore, where he led several projects in video and image content analysis and retrieval and computer vision. He also worked at MIT Media Laboratory in 1994 as a Visiting Researcher. From 1995 to 1999, he was a Research Manager at Hewlett-Packard Labs, where he was responsible for research and technology transfers in the areas of multimedia management; intelligent image processing, and Internet media. In 1999, he joined Microsoft Research Asia, Beijing, China, where he is currently a Senior Researcher and Assistant Managing Director in charge of media computing and information processing research. His current research interests are in the area of image and video processing, content-based media retrieval, and computer vision. He has published three books and over 240 refereed papers, and has also filed over 40 patents.

Dr. Zhang is a a member of ACM. He currently serves on the editorial boards of five IEEE and ACM journals and committees of a dozen international conferences.

Stan Li received the B.Eng. degree from Hunan University, the M.Eng. degree from National University of Defense Technology, and the Ph.D. degree from Surrey University where he also worked as a research fellow. All the degrees are in electrical and electronic engineering.

He is a Researcher at Microsoft Research Asia, Beijing, China. He joint Microsoft in May 2000 from his post as an Associate Professor of Nanyang Technological University Singapore. His current research interest is in pattern recognition and machine learning, image analysis, and face technologies. He has published two books, including *Markov Random Field Modeling in Image Analysis* (Berlin, Germany: Springer-Verlag, 2nd ed., 2001), and over 100 refereed papers and book chapters in these areas. He currently serves on the editorial board of *Pattern Recognition*.

Dr. Li is on the program committees of various international conferences.

Shaoping Ma received the B.E., M.E., and Ph.D. degrees from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 1982, 1984, and 1997, respectively.

He is a Professor at the Department of Computer Science and Technology, Tsinghua University. His current research interests include knowledge engineering, information retrieval, and pattern recognition. He has published two books and more than 50 refereed papers in these areas. Currently, he is the Deputy Director of the State Key Lab of Intelligent Technology and Systems, Director of Machine Learning Institute of China, and Assistant Director of Beijing Computer Institute.