Multi Example Based Image Retrieval : An ICA Based Approach

Jayanta Basak
IBM India Research Lab
IBM, India
bjayanta@in.ibm.com

Koustav Bhattacharya
Dept. of Electrical Engg.
IIT Delhi, India
koustav79@hotmail.com

Santanu Chaudhury
Dept. of Electrical Engg.
IIT Delhi, India
santanuc@ee.iitd.ernet.in

Abstract

In this paper, we consider the problem of designing a CBIR (content based image retrieval) system where multiple query examples can be used to indicate the need to retrieve not only images similar to the individual examples but also those images which actually represent a combination of the content of query images. We propose a scheme for representing content of an image as a combination of features from multiple examples. We have designed novel query processing schemes for image retrieval based upon this representation. We have also shown applicablity of relevance feedback based learning scheme for processing multi-example based queries. Extensive experimental results with facial and natural image databases have validated effectiveness of our approach.

1. Introduction

Collection of digital images or image databases are used in a variety of applications such as journalism, surveillance, fashion design, crime detection, defense, entertainment and e-commerce. Indexing these databases for efficient retrieval is essential for many of these applications. Indexing based upon manual annotation can be tedious, and also highly subjective. Content based image retrieval techniques overcome these difficulties [8] by making use of image based features and attributes.

CBIR(Content based image retrieval) systems[6, 7] use various image features such as color, texture, shape, etc. for indexing images [8]. Various soft-computing models such fuzzy set theoretic approaches, neural networks and genetic algorithms have also been used for the image retrieval applications [8]. Use of learning for retrieval and classification applications are not also uncommon. Many content based retrieval systems adopt QBE (query by example) as the primary query model and have been developed with the fundamental assumption that the user query is specified through an example image or features of the example image. Although a single example image may not always have the ability to represent information need of an user, CBIR systems,in general, do not provide any mechanism for formulating query in terms of multiple examples.

In this paper, we consider the problem of designing a CBIR system where multiple query examples can be used to indicate the need to retrieve not only images similar to the individual examples but, more importantly, also those images which actually represent a combination of the content of query images. A novel query processing scheme has been designed for this purpose. We have also shown applicability of relevance feedback for processing multi-example queries. Experimental results with database of facial images and natural images have validated effectiveness of our approach.

2. Multiple Exemplar Based Retrieval Scheme

2.1. Overview

In order to process multi-exemplar based retrieval queries we need a canonical image representation scheme. Almost all existing CBIR systems are based on image features such as color, texture, and shape descriptors. However, it is difficult to represent a mixture of images using these classical image features. Here we canonically represent images as a collection of independent components [5, 10].We choose a set of windows of specified size from each query image. In our algorithm, the window weights can be fixed or they can be learned. Considering the ensemble of these image windows, we find the independent components from these windows. We then take only the informative independent components considering the non-Gaussianity measure. Each database image is then viewed as a combination of these informative independent components such that we approximate each database image window as an additive mixture of these components. The approximation error reflects how far the database image content matches with the content of query images. A matching score based on this approximation error is used for ranking retrieved images. Retrieved images are similar to given individual examples as well as combination of the content of these examples. We have used relevance feedback to learn to associate different

weights to the windows of the query images for more effective retrieval. Use of independent components for multi-example based image retrieval is a novel contribution of this work.

2.2. Representation

We choose M windows of size $w \times w$ from every query image, the window locations being the same in all images. Each window, therefore, consists of w^2 pixels which can be viewed as a stream (or signal) of pixels of length w^2. We have M such stream of pixels. We view each stream as a linear combination of M independent components which can be obtained by independent component analysis of the M streams together. Figure 1 demonstrates the representation that we used. For every database image we choose M

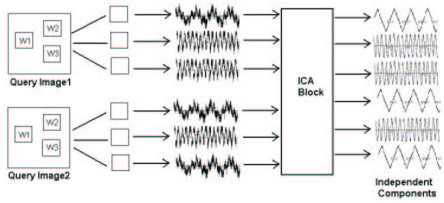

Figure 1. Representation Scheme

windows of the same size and location as the query images. Each window is converted to a pixel stream using the same scanning order as performed for query images. We then find if these pixel stream for a window can be expressed as a linear combination of the meaningful independent components as obtained from the query image windows.

Formally, let the query image windows be represented as $\mathbf{x_1}, \mathbf{x_2}, \cdots, \mathbf{x_{MQ}}$ where each $\mathbf{x_i}$ is stream (or signal) of length w^2. M is the number of windows in each image and Q is the number of query images. We represent the independent components derived from query image windows as $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_{MQ}}$ where each $\mathbf{y_i}$ represents a signal of length w^2. We choose the K informative independent components based on the non-Gaussianity measure such that we have $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_K}$ independent components. Let the database image windows be represented as $\mathbf{d_1}, \mathbf{d_2}, \cdots, \mathbf{d_M}$ where each $\mathbf{d_i}$ represents a stream of w^2 pixels. We then approximate each $\mathbf{d_i}$ as a linear combination of $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_K}$ and find out the error in approximation.

2.3. Error in Approximation of Database Image by Independent Components

First we obtain the optimal linear fit of the informative independent components to a database image window $\mathbf{d_i}$ by solving the linear regression equation given as

$$\mathbf{Y a_i} = \mathbf{d_i} \tag{1}$$

where $\mathbf{Y}$ is a matrix whose column vectors are the informative independent components $\mathbf{y_1}, \mathbf{y_2}, \cdots, \mathbf{y_K}$ such that $\mathbf{Y}$ is a $w^2 \times K$ matrix and $\mathbf{d_i}$ is a $w^2 \times 1$ vector. The coefficients $\mathbf{a_i}$ (which is a $K \times 1$ vector) can be solved for $K \leq w^2$. In the case $K = w^2$, we can get an exact solution provided $\mathbf{Y}$ is invertible. For $K > w^2$, this is an under-complete representation and the problem becomes ill-posed. In general, we have $K < w^2$ (overcomplete representation) which essentially gives us a constraint on the trade-off between the number of windows and the window size. An optimal solution for the overcomplete representation can be obtained by linear regression as

$$\mathbf{a_i} = (\mathbf{Y}^T\mathbf{Y})^{-1}(\mathbf{Y}^T\mathbf{d_i}) \tag{2}$$

Note that, for overcomplete representation, in general, the matrix $\mathbf{Y}^T\mathbf{Y}$ is invertible. The error in the linear regression fit can be obtained as

$$E(\mathbf{d_i}) = \|\mathbf{d_i} - \mathbf{Y a_i}\|^2 \tag{3}$$

Since the matrix $\mathbf{Y}^T\mathbf{Y}$ is symmetric, simple algebraic manipulation gives us

$$E(\mathbf{d_i}) = \mathbf{d_i}^T(\mathbf{I} - \mathbf{Y}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T)\mathbf{d_i} \tag{4}$$

where $\mathbf{I}$ is the identity matrix.

2.4. Ranking of Database Images

For every database image window we obtain an absolute error measure $E(\mathbf{d_i})$ which reflects the error in approximating the database image window by the informative independent components obtained from the set of query images. Thus we obtain M such errors for M windows in every database image. Intuitively, if $E(\mathbf{d_i}) = 0$ for any window $\mathbf{d_i}$ of a database image then it indicates that the content of that window perfectly matches with the content of query images. If for all M windows the error is low then the database image should have high score to indicate that it has a perfect match with the content of query images. We therefore, combine these error measures as

$$f(k) = \sum_{i=1}^{M} V(\mathbf{d_{ik}}) \tag{5}$$

where $f(k)$ denotes the score of the database image k consisting of the windows $\mathbf{d_{1k}}, \mathbf{d_{2k}}, \cdots, \mathbf{d_{Mk}}$. We define $V(\mathbf{d_{ik}})$ as follows:

$$V = \frac{1}{1 + \exp(m(E(\mathbf{d_{ik}}) - \theta))} \tag{6}$$

where m and θ being two parameters controlling the shape of the function. In Figure 2 we plot the nature of the function for various values of m and θ. We choose the values of

m and θ in such a way that a perfect match, i.e. $E(\mathbf{d_i}) = 0$, yields maximum score and as the error values approaches θ the score decreases marginally. However,if the error value is greater than θ then the score becomes very low. The function can be viewed as a soft threshold function such that all images having errors of mismatch less than the threshold θ have higher scores and other images having errors of mismatch greater than θ score very low. Finally the scores of the images are sorted in the descending order and are presented to the user.

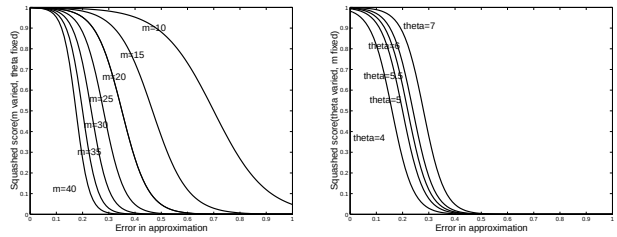

Figure 2. Plot of Eqn.(6) for various values of m and θ

For color images, we compute the score $f(.)$ for each individual channels of red, green and blue. We then combine these three scores by averaging. Alternatively, we can also consider the maximum of these three scores depending upon how stringent we are in the matching process.

2.5. Interpretation of retrieved images

Our scheme intends to retrieve those images which have features similar to those provided by the query images as well as the mixture of those features. We try to approximate a database image as a linear combination of ICs provided by the query images. These informative ICs capture the features of various query images. Retrieved images with low approximation error ($E(\mathbf{d_i}) \approx 0$) can be represented as a linear combination of the informative IC's extracted from query images. Hence, these images are guaranteed to posses features of query images and/or their combinations. We, however, seek efficient learning schemes to determine the appropriate weights for the linear combination so that we can highlight those features which the user is looking for in the query.

3. Learning

For a simple unlearned system we partition the query images with a grid partioning scheme. We compute the score of match of a database image with the query images by simply summing up the scores of the individual windows obtained after a grid partioning of the query image. However, it is possible to judiciously choose weights of the windows in order to extract maximum information from the query images.

3.1. Basic model of the Relevance Feedback based learning scheme

In this subsection we propose a relevance feedback based interactive retrieval approach, which effectively takes into account the problem of learning the weights of the different image windows on the basis of user feedback.

Our relevance feedback technique to be used for CBIR, follows the model proposed in [9] and is based on an image object model, which consist of raw image data, a set of low level visual features associated with the image object, such as color, and a set of representations for a given feature which can itself be vector[9]. Different weights, are associated with features, representations, and components respectively[9]. The goal of relevance feedback is therefore, to find the appropriate weights to model user's information need.

3.2. Relevance Feedback technique using Independent Components

In our relevance feedback based learning technique we follow a two level weight updation scheme. The two level structure and updation scheme are discussed next.

i.*Lower level features and associated weights*

Let $\mathbf{V_R}, \mathbf{V_G}$, and $\mathbf{V_B}$ represent the matching score vectors obtained in the R, G and B streams respectively. Each of $\mathbf{V_R}$, $\mathbf{V_G}$, and $\mathbf{V_B}$ are vectors with M components whose values are given by Equation (6). We define $\mathbf{W_R}$, $\mathbf{W_G}$ and $\mathbf{W_B}$ as the lower level weight vectors. Thus, if we define S_R, S_G and S_B as the weighted scores in R, G and B streams respectively, we have

$$S_R = \mathbf{W_R^T}.\mathbf{V_R} \tag{7}$$

$$S_G = \mathbf{W_G^T}.\mathbf{V_G} \tag{8}$$

$$S_B = \mathbf{W_B^T}.\mathbf{V_B} \tag{9}$$

ii.*Higher level features and associated weights*

We define $\mathbf{W}$ as the higher level weight vector and is represented as $\mathbf{W} = [w_1 w_2 w_3]$. If we define the score vector $\mathbf{S}$ as $\mathbf{S} = [S_R S_G S_B]$ then final matching score of a database image can be obtained as

$$Score = \mathbf{W^T}.\mathbf{S} \tag{10}$$

iii. *Updation scheme for higher level weights*

The updation procedure for the weights at the higher level, can be outlined as follows. Let RT denote the set of retrieved images (consisting of N_{RT} images) ordered according to the value $Score$ (Equation 10). Let RT_1, RT_2 and RT_3 the set of retrieved images ordered by the scores S_R, S_G or S_B respectively. Given this, the higher level weights are calculated as follows:

$$W(j) = \begin{cases} W(j) + UF_l, \text{ if } RT_j^l \text{ is in } RT \\ W(j) + 0, \text{ if } RT_j^l \text{ is not in } RT \end{cases} \tag{11}$$

where $l = 0, ...N_{RT}$, $j = 1, 2$ or 3, and UF_l is the user's relevance feedback score on the l^{th} retrieved image.

We have found experimentally that for our system UF (Equation 11) scores of 5, 1, 0, -1 and -5 representing highly relevant, quite relevant, relevant, less relevant and not relevant respectively, results in stable learning of weights.

iv. *Updation scheme for lower level weights*

For calculating the lower level weights we simply stack the representation vector $\mathbf{V_R}$, $\mathbf{V_G}$, and $\mathbf{V_B}$ row-wise for all images. We then retain only those rows that finally corresponded to retrieved images which were marked relevant in the user's feedback. The weight is then simply the inverse of the variance along a column in this representation, i.e.,

$$W_j(k) = 1/\sigma_j(k) \tag{12}$$

where $j = R, G$ or B and $\sigma_j(k)$ denotes the variance of the k^{th} column for the corresponding value of j in the representation. Thus, if all elements in a column have similar values then that component is a good indicator of the user's information need and is associated with higher weight. On the other hand, if the values of the elements in a particular column have large deviations then that component is not a good indicator of user's need and hence associated with a lower weight.

4. A faster method for indexing based on clipping in IC-space

Till now we have been using the approach that the Independent Components are calculated from the query images provided, and for each database image the error in approximation was found. This was found to be computationally very costly.

To reduce the computational burden we propose an alternative computation scheme. For a given image database, we compute Independent Components(ICs) for the given set of images. We select a small number, in our case three ICs from that space, and project each database image to the IC-space formed by the three selected ICs. Hence database images can be considered as points in the IC space. Any query image projected onto this IC-space will be a point. Database images lying within a neighborhood of the given query image will be a candidate for retrieval. When multiple examples are provided problem becomes more complex. Here we present a formulation for dealing with 3 query images. This can however be generalized for more query images.

The three query images forms a planar triangle in the IC space. Images lying within the triangle can be definitely represented through linear combination of query images. Our problem of finding similar images to the query images as well as images that are mixtures of the query images is equivalent to clipping the points inside the triangle formed. We also need to define a volume surrounding the finite plane formed by the three query images to find other similar images as well as their mixtures.

4.1. Projection, Clipping and defining an Approximate volume in IC space

We project the database image points in the IC-space to the infinite plane formed by the three query images. The point of intersection is checked to see whether it falls within the clip boundary formed by the three query images in the IC-space.

We define the distance of projection from the point to the plane as d_1 and if C is the centroid of the triangle, the distance of this projected point in the plane to the centroid as d_2. Our error in approximation is formed by taking

$$\mathbf{d_i} = \alpha \times \mathbf{d_{1_i}} + \beta \times \mathbf{d_{2_i}} \tag{13}$$

for the ith database image point in the IC-space. We define α as the coefficient of overall error and β as the coefficient of mixture. The $\mathbf{d_i}$ is then squashed using a sigmoidal squashing function and finally scores are presented to the user. The overall conceptual framework of clipping in the IC space is depicted in Figure 3.

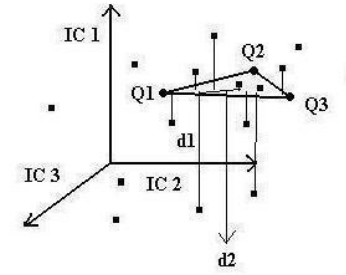

Figure 3. Conceptual framework

4.2. Comparison of the two approaches and performance evaluation

We note that the computational complexity of both the retrieval schemes is $O(n)$ where n is the number of database images. However, computational cost for each step in the second framework is less than that of the first because in the second approach IC computation for database images is done offline.

The first approach, on the other hand, provides opportunity of query adaptive learning as well as handling of local features using window based processing.

5 Experimental Results

In this section we analyze the retrieval results obtained with both the approaches as well as with and without learning. The images retrieved are sorted according to their score and then displayed.

5.1. Protocol

We experimented with the system in the Matlab environment on Windows 2000 running on Intel Xeon dual processor of 1.7 GHz and 2 GB RAM. We used the face database provided by California Institute of Technology [3]. The face database consisted of 450 images, each of size 128×128, of twenty different people in various background and lighting conditions. These images were scaled to 128×128 for further processing. Additionally, we test the system also with natural images provided by University of Washington[1] and also few natural images we had locally. There are 350 such natural images and were converted to 128×128 dimensions. We compute independent components by using Hyvarinen's fastICA algorithm [2].

We used subjective measures to evaluate the effectiveness of our retrieval system. We gathered feedback from eight independent subjects and took the average of the ratings provided by all subjects. We quantify the performance of our system by *precision* and *recall* measures. As we plot the *precision* and *recall* measures against the number of retrieved images, they intersect at a point which indicates the number of images required to be retrieved for a reasonable performance of the system. We refer this number as the *efficacy* point of the system. We have used this efficacy point to analyze performance of the system.

5.2. Retrieval Examples without Learning

Independent components of images capture most of the relevant information present in the images. We have shown this in [4] where a single query image of a face makes our system work as a face recognition system.

We provide results for some examples of mixture query for a system without relevance feedback to compare performance with our relevance feedback based learning technique. We show (in Figure 4) an example where we provide three different face images as the query. Here we used a window size of 32×32 such that we have a total of 16 windows for each image. For three query images we have a total of 48 windows. The basic intent of the query is to retrieve facial images similar to the feature of a 'french cut beard' (first two example images) or a 'mongoloid face' (indicated by the third query image of the girl) or a combination of both. Our objective here is to retrieve similiar images provided in the query images as well as possibly their mixtures. We observe that the images 1, 2, 3, 4, 7, 10, 11, 12, 14, 16 and 18 are various expressions of the subjects provided in the query images. Images 8 and 9 have mongoloid like features while image 14 is prominently mongoloid which exemplify variants of features in the query images. Mixture of features of the query images is also handled by our IC based scheme as shown in image 20 in the Figure but such images are ranked much behind. Thus images similar to those of the query images are indeed retrieved. The mixture of the features is also retrieved, but is not ranked highly. To improve the ranks of the similiar images as well as their mixtures we employ relevance feedback based learning. We also

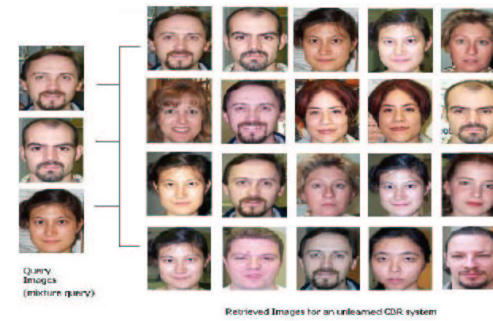

Figure 4. The results of facial image retrieval from an system without relevance feedback where three query images are used. The results are obtained with simple windowing where each image is divided into a grid of windows.

demonstrate performance of our system without relevance feedback on images of natural database in Figure 5. In our query images we provide images of green trees and that of a rocky mountain, and the user is interested in images of the green trees or that of rocky mountains as well as the mixture of the two, namely, images of greeny mountains. We notice that the system approximately retrieves images of trees in images 2, 3, 4, 5 and of mountains in images 1 and 6. Image 8 contains scene of clouds which is also present in the query images. Images 7, 9 and 10 have textures similiar to those of mountains. The images having mixed characteristic are not apprarent as they are not ranked in the top order retrieved images. Thus here again a system without relevance feedback does capture the features of the query images and retrieves images similiar to it as well as their combinations but relevance feedback is needed to improve the ranks of the relevant images.

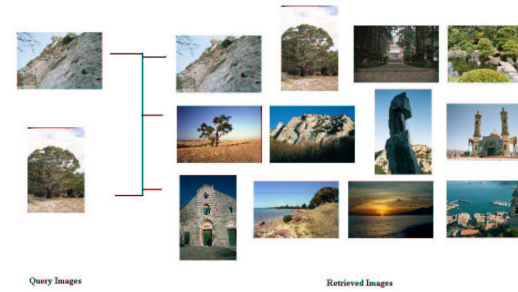

Figure 5. Results of our unlearned CBIR system on Natural Images

5.3. Performance with learning using user's relevance feedback

We found in the previous examples that our system without relevance feedback does retrieve similar images as well images consisting of the mixture of features. However, we would like to improve the ranks of these relevant images using a learning technique. In this section, we demonstrate the effectiveness of the system with relevance feedback based learning in improving the ranks of the intended images.

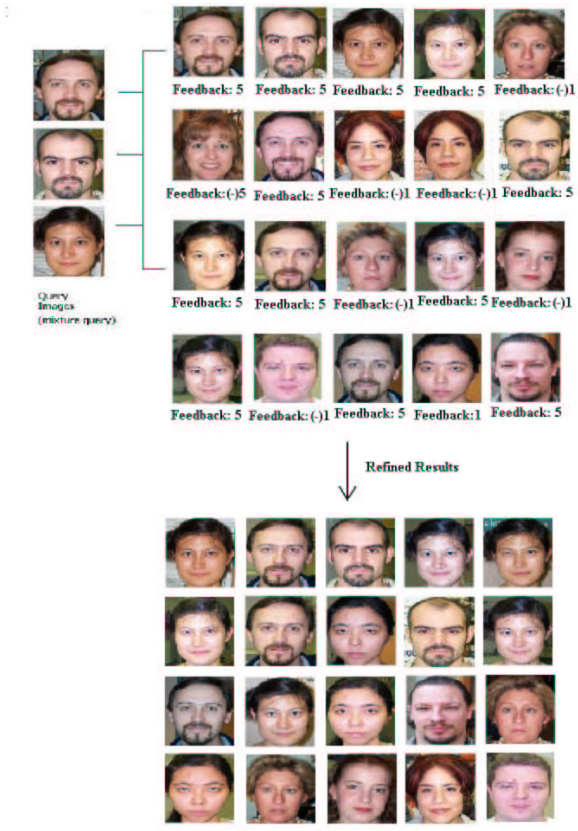

Figure 6. The results of facial image retrieval from our feedback system when three query images are used. Learning is performed by updating the weights according to user's response scores.

As shown in Figure 6, we provide three query images. We assume the user's true information need is to retrieve similar images to those of the query images as well as those having a mixture of features provided by them. In the proposed retrieval approach, the user is no longer required to explicitly map his information need to low-level features, but rather he or she can express his intended information need by marking the relevance scores of the returned images. We can see in this example that after refinement more and more relevant images are retrieved. Images 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12 after refinement are all various expressions of the subjects provided in the query images. We now see that image of the subject in image 8 after refinement, with proper mongoloid features, is provided a higher rank after dynamic updating of the weights based on user's feedback. We also see images of this subject in image 13 and 16. The improvement of the rank of similiar images is clearly due to 'online' learning of weights through passes of user's relevance feedback. We note that images 17, 18, 19 and 20 are given lower ranks due to user negative feedback. The image retrieved in image 6 before refinement is completely eliminated due to high negative relevance feedback of the user. The refined results displayed are actually after receiving 8 passes of user feedback to retrieved images and dynamic updating of system's weight based on the feedback. We note that the ranks of the images consisting of the mixture of features has also improved from image 20 before refinement to image 14 after refinement through relevance feedback passes.

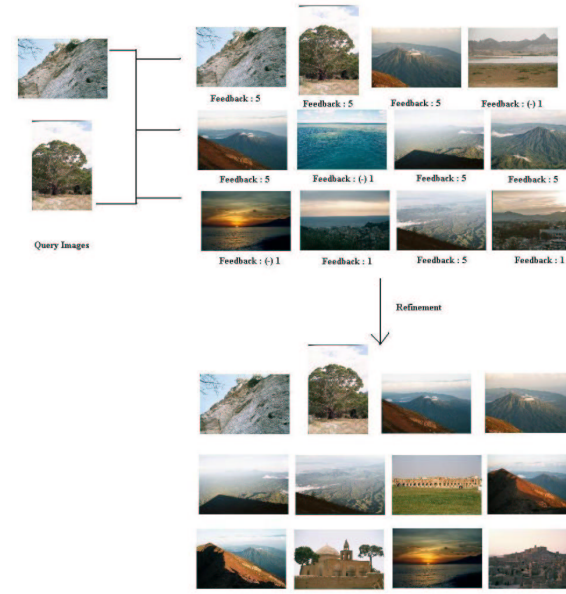

Figure 7. Results of Relevance feedback based CBIR on Natural Images

We also provide the results of a multi-exemplar query consisting of natural images when it is presented to our system in Figure 7. We note that the system not only retrieved images such as image 7 after refinement which contains greenness feature of a tree and are similar to the query image is ranked highly, but also the system has ranked highly images consisting of mixture of features in image 3, 4, 5, 6 and 9 after refinement, which are actually pictures of greeny mountains. Image 10 matched the texture of the rock with that of the greenness feature of the tree. Thus after learning, images similar to query images are retrieved at higher ranks.

Also ranks of images retrieved due mixture of features provided by query images have also increased significantly.

5.4. Performance of the System when we use Clipping in the IC-space for Indexing

We demonstrate the performance of our system when our Model II computation is used with ICA to retrieve relevant images as well as images consisting of mixture of features of the query images, as shown in Figure 8. We demonstrate performance of this model of computation on the same natural image database. We find that here also different variants of the query images (which consist of a images of mountains and trees) as well as their mixtures are among the retrieved images. Images of mountains are retrieved in images 1, 9, 10 and 11. Images of green trees and their variants are found in images 2, 3, 4, 7, and 12. Typical mixture of features of the query images are found in images 9 and 11.

We note that in this approach the average precision and recall values when about 20-25 images were retrived were 0.665 and 0.522 respectively which is much lower than the corresponding values found using relevance feedback(0.72 and 0.635 resp.). This shows that Model II results in faster searches at the expense of accuracy of retrieval.

Figure 8. Results of our CBIR system using the technique of clipping in IC space

5.5. Performance Comparison

We discuss overall performance of the system in this section. We observe that the efficacy for the system with relevance feedback is greater than that for the system having no relevance feedback. This is evidenced by comparing the efficacies of the system using relevance feedback with that of the system without relevance feedback. In Table 1, we report the efficacy of the system without relevance feedback and our relevance feedback based system for different number of refinement passes. We have considered 180 different queries and the average recall and precision performance is computed.

	No. of passes of the feedback		
	5	10	20
Relevance feedback	0.582	0.633	0.697
Unlearned	0.478		

Table 1. The efficacy of our relevance feedback based system along with that of the system without relevance feedback.

6. Conclusions

In this paper we have proposed a technique for retrieving images using multiple example based query. Our methodology leads to retrieval of images similar to the given query images as well as their combination. We also explored relevance feedback based learning paradigm, that is suitable for image domains like face databases and natural image databases in particular, to learn the window weights required for efficient retrieval.

References

[1] Computer vision test images. http://www.cs.washigton.edu/research/imagedatabase/.

[2] Fastica matlab implementation. http://www.cis.hut.fi/projects/ica/fastica/code/dlcode.html.

[3] Frontal face dataset. http://www.vision.caltech.edu/html-files/archive.html.

[4] J. Basak, K. Bhattacharya, and S. Chaudhury. Multiple exemplar based image retrieval using independent component analysis. *Communicated [now under revision] to IEEE Transactions in System, Man and Cybernetics (PartB: Cybernetics)*, May, 2004.

[5] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36:287 –314 , 1994.

[6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, and B. Dom. Query by image and video content: The qbic system. *IEEE Computer*, 28:23 –32 , 1995.

[7] W. Y. Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. volume 1, pages 568 –571 , 1997.

[8] Y. Rui and T. Huang. Image retrieval :current techniques, promising directions and open issues. *Journal of visual Communication and Image Representation*, 10:39 –62 .

[9] Y. Rui, T. S. Huang, and S. Mehrotra. Relevance feedback techniques in interactive content-based image retreival. pages 26 –29 , 1997.

[10] H. H. Yang and S. Amari. Adaptive on-line learning algorithms for blind separation - maximum entropy and minimum mutual information. *Neural Computation*, 9:1457 –1482 , 1997.