Image Database Retrieval Utilizing Affinity Relationships

Mei-Ling Shyu
Department of Electrical and
Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
1-305-284-5566
shyu@miami.edu

Shu-Ching Chen, Min Chen, Chengcui Zhang
Distributed Multimedia Information System Laboratory
School of Computer Science
Florida International University
Miami, FL 33199, USA
1-305-348-3480
{chens, mchen005, czhang02}@cs.fiu.edu

Kanoksri Sarinnapakorn
Department of Electrical and
Computer Engineering
University of Miami
Coral Gables, FL 33124, USA
1-305-284-5566
ksarin@miami.edu

ABSTRACT

Recent research effort in Content-Based Image Retrieval (CBIR) focuses on bridging the gap between low-level features and high-level semantic contents of images as this gap has become the bottleneck of CBIR. In this paper, an effective image database retrieval framework using a new mechanism called the *Markov Model Mediator* (MMM) is presented to meet this demand by taking into consideration not only the low-level image features, but also the high-level concepts learned from the history of user's access pattern and access frequencies on the images in the database. Also, the proposed framework is efficient in two aspects: 1) Overhead for real-time training is avoided in the image retrieval process because the high-level concepts of images are captured in the off-line training process. 2) Before the exact similarity matching process, Principal Component Analysis (PCA) is applied to reduce the image search space. A training subsystem for this framework is implemented and integrated into our system. The experimental results demonstrate that the MMM mechanism can effectively assist in retrieving more accurate results from image databases.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – Retrieval Models.

General Terms

Algorithms, Experimentation.

Keywords

Content-based Image Retrieval, Markov Model Mediator (MMM), and Principal Component Analysis.

1. INTRODUCTION

The explosion of image databases and the inefficiency of text-based image retrieval have created an urgent need for effective approaches in image database retrieval. Content-Based Image Retrieval (CBIR) approaches have been actively researched to address this need. These approaches utilize the features that are inherent in the images, such as color [1], shape [2] and texture [3], for retrieval purposes.

However, an impediment to research on CBIR is the lack of mapping between the high-level concepts and the low-level features. Currently, research efforts to address this issue fall into two major categories: region-based image retrieval and relevance feedback (RF). Region-based retrieval attempts to overcome the deficiencies of global feature matching processes by representing images at the object level, which is achieved by segmenting an image into a number of regions (i.e., a group of connected pixels which share common properties) and then the features of each region can be extracted and compared [4]. The object-level representation is considered a more natural way to capture human's perception. However, currently, there is hardly any unsupervised segmentation algorithm that can partition an image into regions/objects perfectly, especially when the image database contains a large collection of heterogeneous images. Therefore, the inaccurate object-level representation may result in poor retrieval performance. Another major approach to capture the semantic contents of images is relevance feedback (RF) [5]. Relevance feedback is an interactive process used in image retrieval systems, which refines the query performance based on the user's feedback on the quality of the previous query results performed by the system. Most of the relevance feedback research work has been carried out in two approaches: query point movement and re-weighting [6]. However, due to the fact that the metric dynamically depends upon the user's feedback and the context, how to provide real-time learning capability of the distance metric or feature space transformations based on the users' interactions is crucial and remains an open issue.

In this paper, a novel content-based image retrieval system that utilizes not only the low-level features of the images but also the high-level user concepts such as the affinity relationships among the images in the image retrieval process is proposed. The central core of the proposed CBIR system is the Markov Model Mediator (MMM) mechanism, which has been applied to multimedia database management [11] and document management on the

World Wide Web (WWW) [12]. The MMM mechanism adopts the mediator concept and the Markov model framework. A mediator is defined as a program that collects data from one or more sources, combines the data for processing, and yields the resulting information [7]. Markov model is a well-researched mathematical construct for analyzing complicated systems and has been used in lots of research work including Markov Random Field Models [8], and Hidden Markov Models (HMMs) [9]. Especially, HMMs have been integrated into multimedia domain [10].

In the proposed framework, the high-level user concepts are captured through the training process. Instead of learning and interpreting user's preferences through on-line user feedback, the relative affinity relationships among images in the image database are calculated off-line via the user access patterns and access frequencies from the training data, which serves as an indication of users' high-level concepts in our system. This approach provides the capability to reduce the gap between low-level features and high-level concepts without sacrificing the system's efficiency. In addition, for the sake of retrieval efficiency, Principal Component Analysis (PCA) is integrated into our proposed framework as the pre-filtering phase. Different from the approach proposed in [18], we avoid the expensive distance computation by utilizing the principal component scores to identify the images in the desired candidate image pool, and the distance computation is required only for those images in the candidate image pool. In order to test the performance of the proposed framework, a training subsystem has been implemented to collect the user access patterns and access frequencies, which is integrated into our system [20].

The remainder of this paper is organized as follows. In Section 2, the architecture of the proposed framework is introduced, followed by the detailed discussions for each important component, such as feature extraction process, training process, and retrieval process as well. Section 3 presents the system implementation and our experimental results in applying the proposed framework to content-based image retrieval. The experimental results demonstrate that our framework can assist in retrieving more accurate results for user queries. A brief conclusion is given in Section 4.

2. ARCHITECTURE OF THE PROPOSED FRAMEWORK

The architecture of the proposed framework is shown in Figure 1. As can be seen from this figure, our proposed framework is divided into three major components based on their functionalities, namely image feature extraction process, training process, and retrieval process. In our framework, not only the low-level features (e.g., color) and the mid-level features (e.g., object locations), but also the high-level concepts learned from the off-line training process are used in the image retrieval process. Moreover, instead of conducting the exact similarity matching process in the whole database scope, a pre-filtering process using PCA is applied to reduce the search space. Hence, the retrieval process consists of the pre-filtering process and similarity matching process. In the following three subsections, each component and the relationships among the components are presented in details.
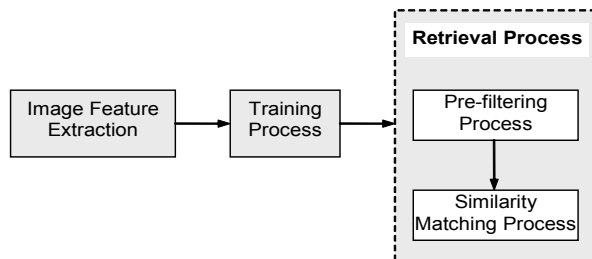

Figure 1: Architecture of the proposed framework

2.1 Image Feature Extraction Component

The features of color information and object location information for the images in the image database are both considered in our proposed framework. The HSV color space is used to obtain the color feature for each image due to the following two reasons: (1) it is perceptual and a proven color space particularly amenable to color image analysis [19], and (2) it has shown in the benchmark results that the color histogram in the HSV color space has the best performance [13]. For object location information, the SPCPE algorithm proposed in [14] is used, where each object is covered by a rectangle since the minimal bounding rectangle (MBR) concept in R-tree [15] is adopted. In addition, the centroid point of each object is used for space reasoning so that each object is mapped to a point object.

In this study, our main focus is not to explore the most appropriate features for image retrieval. Instead, we would like to evaluate the performance of the MMM mechanism and to reduce the feature space. Hence, each image has a feature vector of twenty-one elements, where twelve are for color descriptions and nine are for location descriptions. Based on the combinations of different ranges of the hue (H), saturation (S), and the intensity values (V), the color features 'black' (black), 'white' (w), 'red', 'red-yellow' (ry), 'yellow' (y), 'yellow-green' (yg), 'green' (g), 'green-blue' (gb), 'blue' (b), 'blue-purple' (bp), 'purple' (p) and 'purple-red' (pr) are considered. For any color whose number of pixels is less than 5% of the total number of pixels, its corresponding position in the feature vector has the value **0** since we treat it as non-important. Otherwise, the corresponding percentage of that color component is put in that position. For the location descriptions, each image is divided into 3×3 equal-sized reference regions that are ordered from left to right and top to bottom as L1, L2, L3, L4, L5, L6, L7, L8, and L9 (as shown in Figure 2). The value **1** is assigned to a region when there is an object in the image whose centroid falls into that reference region; and **0** otherwise. Moreover, an object will be ignored if its area is less than 2% of the total image area. When it is necessary, each image can be divided into a coarser or finer set of regions.
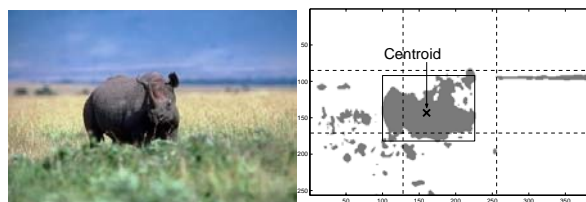

Figure 2: Object locations and their corresponding regions

In order to capture the appearance of a feature in an image, a feature matrix $\mathcal{B}$ is defined in the way that its rows are all the distinct images and columns are all the distinct features, where the

value in the (p, q) entry is greater than zero if feature q appears in image p, and zero otherwise. We consider that the color and object location information are of equal importance. That is, the color features are normalized and their sum equals 0.5, and the location features are normalized and their sum is 0.5. Hence, each value indicates the probability of a feature observed from an image and the sum of the probabilities of all the features of the image is 1. It is worth mentioning that our framework is flexible in terms of the update on the feature matrix $\mathcal{B}$. Any normalized vector-based image feature sets can be used to construct the $\mathcal{B}$ matrix. Table 1 illustrates the normalized feature vectors of the sample images in Figure 3.
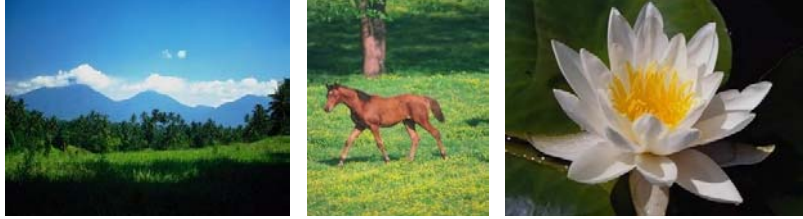

Figure 3: Three sample images (Image 1 – Image 3)

Table 1: $\mathcal{B}$ matrix - Normalized image feature vectors of the sample images

	black	w	red	ry	y	yg	g	gb	b	bp	p	pr	L1	L2	L3	L4	L5	L6	L7	L8	L9
Image 1	0.14	0.04	0	0	0	0	0	0.08	0	0.24	0	0	0	0.25	0	0	0	0	0	0.25	0
Image 2	0	0	0.04	0	0	0.17	0.29	0	0	0	0	0	0	0.25	0	0	0.25	0	0	0	0
Image 3	0.15	0.17	0	0.04	0	0.07	0.07	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0

2.2 Training Process Component

Markov Model Mediator (MMM) is a probabilistic-based mechanism which captures the probabilistic relationships among the states from the training data. In our framework, each image is called a state; while the probabilistic relationships among the states (images) are contained in the relative affinity matrix $\mathcal{A}$, which is obtained from the off-line training process and serves as the indications of users' high-level concepts for image retrieval. In particular, $\mathcal{A}$ is defined to represent the affinity relationships among the images in the database. In order to construct the MMM model, a training subsystem is implemented using a multi-threaded client/server architecture to collect the user access patterns and access frequencies. Figure 4 shows the flowchart of the training process.
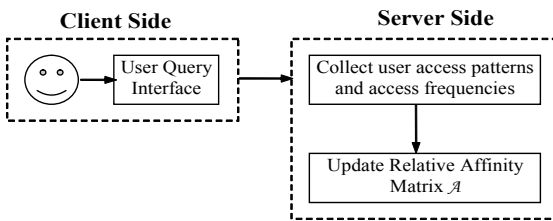

Figure 4: Flowchart of the training process

As can be seen from this figure, a user query interface is provided in the client side; while the server side collects the user access patterns and access frequencies from the client and updates the relative affinity matrix $\mathcal{A}$. The detailed implementation and operation of the training subsystem are described in Section 3.2.

2.2.1 Training Data Set

In our framework, the training subsystem is implemented to record the history of user access patterns and access frequencies on the image database during the training period. User access patterns denote the co-occurrence relationship among images accessed by user queries; while access frequencies denote how often a certain query was issued by the users. This training data set is used to bring in users' subjective concepts and to construct the relative affinity matrix $\mathcal{A}$ off-line. Definition 1 gives the information available in the training data set.

Definition 1: Assume N is the total number of images in the database. The training data set consists of the following information:

- A set of queries $Q = \{q_1, q_2, \ldots, q_q\}$ that are issued to the database in the training period. Let $use_{k,m}$ denote the access pattern of image m ($1 \leq m \leq N$) with respect to query q_k per time period, where

$$use_{k,m} = \begin{cases} 1 & \text{if image } m \text{ is accessed by } q_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The value of $access_k$ denotes the access frequency of query q_k per time period.

The pair of user access pattern ($use_{k,m}$) and user access frequency ($access_k$) provides the capability to capture the user concepts during the training process, as illustrated below.
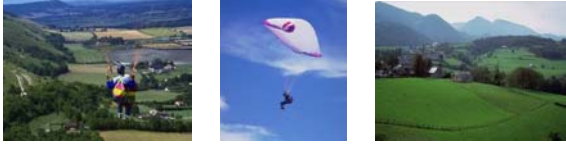
| | Image 4 | Image 5 | Image 6 |
Figure 5: Three sample images (Image 4 - Image 6)

Table 2: The user access patterns ($use_{k,m}$) for Image 4 - Image 6

	Image 4	Image 5	Image 6	…
q_1	1	1	0	…
q_2	0	0	1	…
q_3	1	0	1	…
…	…	…	…	…

Table 2 gives three example queries issued to our image database, where q_1 is a user-issued query related to retrieving some images with parachute jumping scenes, and q_2 and q_3 are two other queries with their interests in natural scenes with sky, mountain and grass. It is worth mentioning that in our training process, the queries issued to each image category (landscape, flower, etc.) are almost the same because the query images are randomly selected by the training system from different image categories. By recording different users' feedbacks during the training process, the corresponding access frequencies $access_k$ for q_1, q_2, and q_3 are 8, 7 and 1, respectively. In Table 2, the entry $(k, m) = 1$ indicates that the mth image is accessed by query q_k. For example, Image 4 and Image 5 (as shown in Figure 5) are accessed together in q_1 with their corresponding entries in the user access pattern matrix having value 1, and the access frequency $access_1$ for q_1 is 8. As for queries q_2 and q_3 that focus on retrieving natural scene images, $access_2$ equals to 7 while $access_3$ equals to 1. Notice that Image 4 and Image 6 are accessed together in q_3 but not in q_2, which is due to the different human perceptions (parachute jumping scene or landscape scene) on Image 4. However, since most of the users regard Image 4 as a parachute jumping scene rather than a landscape scene, the value of $access_1$ is significantly larger than that of $access_3$. Consequently, after the system training, Image 5 is more likely to be retrieved than Image 6, given Image 4 as the query image. Thus, the users' subjective concepts about the images are captured by the pair of user access pattern and user access frequency.

2.2.2 Relative Affinity Matrix $\mathcal{A}$

Based on the information contained in the training data set, we can capture the affinity relationships among the images in the database. That is, the more frequently two images are accessed together, the more closely they are related to each other. The relative affinity matrix $\mathcal{A}$ is constructed in two steps as follows:

- Firstly, a matrix $\mathcal{AF}$ is defined to capture the affinity measurements among all the images using user access patterns and access frequencies as follows:

Definition 2: Each entry $aff_{m,n}$ in matrix $\mathcal{AF}$ indicates how frequently two images m and n are accessed together, and consequently how closely these two image are related to each other, where

$$aff_{m,n} = \sum_{k=1}^{q} use_{m,k} \times use_{n,k} \times access_k \qquad (2)$$

- Then the matrix $\mathcal{A}$ can be obtained via normalizing $\mathcal{AF}$ per row and represents the relative affinity relationships among all the images in the database. Let $a_{m,n}$ be the element in the (m, n) entry in $\mathcal{A}$, where

$$a_{m,n} = \frac{aff_{m,n}}{\sum_{n \in d} aff_{m,n}} \qquad (3)$$

For the sake of efficiency, instead of updating $\mathcal{A}$ matrix on-line, the training system only records all the user access patterns and access frequencies during a training period. Once the number of newly issued-queries reaches a threshold, the update of $\mathcal{A}$ matrix is triggered automatically. All the computations are done off-line.

2.3 Retrieval Process Component

After the training process, the low-level features and the high-level concepts of the images in the database are connected more closely, and the system is ready for image retrieval. However, as we mentioned earlier, efficiency is an important issue in CBIR. The proposed content-based image retrieval framework solves this problem by conducting the retrieval process in two steps:

1. A pre-filtering process is performed on the principal component subspace to reduce the search space.

2. Perform the actual retrieval process that computes the similarity functions on the original feature space only for those candidate images.

2.3.1 Pre-Filtering Process

The principal component analysis (PCA) [16] is integrated into our framework to generate a smaller feature space obtained by applying PCA on the original feature space (matrix $\mathcal{B}$). The resulting principal components will form a more compact feature space for the purpose of pre-filtering. By reducing the search space in this manner, the number of images that need to perform exact similarity matching is reduced, which in turn reduces the computation time since exact similarity matching is more expensive in terms of computation.

2.3.1.1 Principal Component Analysis (PCA)

The main capability of PCA is to reduce the dimensionality of the original feature space without losing the essential information [16]. Principal components are particular linear combinations of p features (X_1, X_2, …, X_p), which hold three important properties. First, principal components are uncorrelated. Second, the first principal component has the highest variance, while the second principal component has the second highest variance, and so on. Third, the total variation in all principal components combined is equal to the total variation in the original features X_1, X_2, …, X_p. Principal components can be easily obtained from the eigenanalysis of the covariance matrix or the correlation matrix of the original features. Principal components from these two matrices usually are not the same, and they are not simple functions of the others. It is better to perform PCA on the correlation matrix if the variables

are measured on scales with wildly different ranges or if the units of measurement are not commensurate [17].

Although it requires p principal components to reproduce the total system's variability, often a small number k of the principal components are sufficient to explain the variability. In such cases, the initial p features can be replaced by the first k principal components, and the original data set (consisting of n measurements on p features) is reduced to a data set consisting of n measurements on k principal components. There will be almost as much information in the k components as there is in the original p features, if k is chosen appropriately.

2.3.1.2 Obtain Principal Components
As mentioned earlier, the feature matrix $\mathcal{B}$ contains 12 color features and 9 spatial location features denoted by $X_1, X_2, ..., X_{21}$ that are normalized to lie in the range of 0.0 to 0.5. In this study, principal components are then obtained from the covariance matrix computed from $\mathcal{B}$. Only the first two principal components that account for 50.12% of the total variation in the data are used as image pre-filtering, which can reduce the search space and lower the processing time. Though we sacrifice some of the information in the original feature space in an exchange for search efficiency, these two principal components seem adequate since they provide reasonably good retrieval results (as illustrated in the experimental results in Section 3).

2.3.1.3 Pre-Filtering to Reduce the Search Space
Since the image retrieval process is usually computationally expensive in particular with a large number of features, the pre-filtering step is needed to reduce the search space. The principal component scores for each image in the database are computed and stored together with the image. The idea for pre-filtering is to get all the images that have principal component scores close to those of the query image in the principal component subspace, while filtering out those images that are unlikely to be similar to the query image.

For a large image database, the computation is very expensive. To achieve real time processing, we propose to use principal component scores 1 and 2 to construct a candidate image pool and only the images included in this candidate pool will require distance computations. Given a desired candidate pool size c (e.g., 100), the top c images that have principal component score 1 closest to that of the query image will be identified. Similarly, the top c images for principal component score 2 are also identified. The intersection of the two sets of images is the desired candidate image pool. If the size of the intersection set is less than c, then the scan scope in component scores 1 and 2 is increased until the candidate pool is filled. This simpler approach is reasonable due to the fact that principal components are uncorrelated and therefore the nearest neighbors found from individual principal component distributions will not be much different from the joint distribution.

2.3.2 Similarity Matching Process
After the pre-filtering process, the size of the candidate image pool for a certain query can be reduced dramatically. Then the exact retrieval process can be carried out to extract the most matched images based on both the image features and the relative affinities among image learned from the training process. Let C be the total number of images in the candidate image pool, p be the total number of distinct features of the images in the database, and the non-zero features of the query image q be denoted as $\{o_1, o_2, ..., o_T\}$, where T is the total number of non-zero features in the query ($1 \leq T \leq p$).

Definition 3: $W_t(i)$ is defined as the edge weight from the image i to the query image q at the evaluation of the tth feature (o_t) in the query, where $1 \leq i \leq C$ and $1 \leq t \leq T$.

Based on the definition, the retrieval algorithm is given as follows.

At $t = 1$,

$$W_1(i) = a_{q,i}(1 - |b_i(o_1) - b_q(o_1)| / b_q(o_1)) \tag{4}$$

The values of $W_{t+1}(i)$, where $1 \leq t \leq T\text{-}1$, are calculated by using the values of $W_t(i)$.

$$W_{t+1}(i) = W_t(i)(1 - |b_i(o_{t+1}) - b_q(o_{t+1})| / b_q(o_{t+1})) \tag{5}$$

Then the similarity function is defined as:

$$S(i) = \sum_{t=1}^{T} W_t(i) \tag{6}$$

Here, $a_{q,i}$ ($a_{q,i} \in \mathcal{A}$) is the relative affinity relationship to indicate how closely the query image q is related to image i, and $b_i(o_k)$ ($b_i(o_k) \in \mathcal{B}$) is the value of the kth feature extracted from image i. The $S(i)$ value for image i in the candidate image pool represents the similarity score between images q and i, where a larger score suggests the more similarity between images q and i.
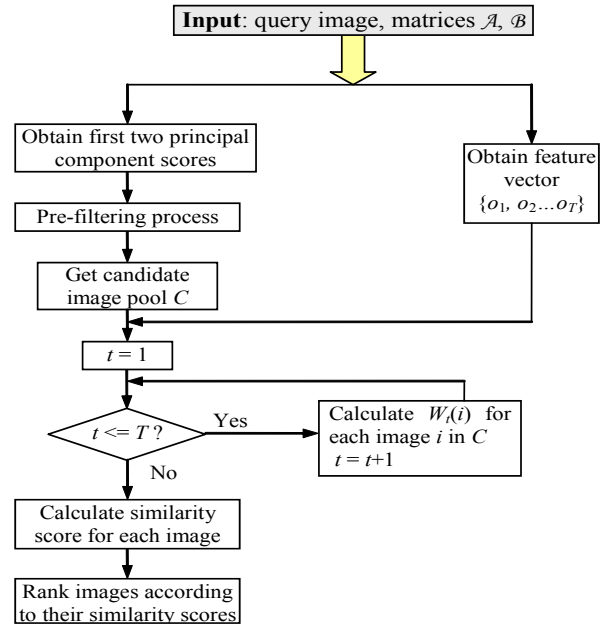
Figure 6: Flowchart of our proposed image retrieval process

Figure 6 gives the flowchart of our proposed image retrieval process. Unlike other methods that either have difficulties to capture the high-level concepts or try to learn the concepts in real-time, our proposed framework captures the user concepts off-line from the training data set and achieves high efficiency in terms of storage and retrieval due to the following reasons: 1) Before the exact matching process, the pre-filtering process

generates a small set of candidate images at low cost and with high accuracy, which normally accounts for 4%-8% of the total number of images in the database, and 2) Given a query image q issued by a user, in the retrieval process, only the data in the row q of matrix $\mathcal{A}$ are used. In addition, normally the features contained in one query image is no more than six, which enables us to load less than half of the whole feature matrix. Thus we can retrieve the results more accurately and efficiently.

3. EXPERIMENTS

3.1 Image Database

Our image database consists of 10,000 color images of 72 semantic categories with various dimensions. In our experiments, the color information and object location information of the images are considered and the query-by-example strategy is used. For the purpose of supporting high-level meaning in the queries, a training subsystem is implemented to collect the user access patterns and query access frequencies information for the training data set.

3.2 Implementation of Training System

The training subsystem for this framework is implemented and integrated into our system [20], a prototype multimedia management system developed by our research group aiming to support a comprehensive set of functionalities and components for multimedia database management systems. Figure 7 shows the interface of the training subsystem.
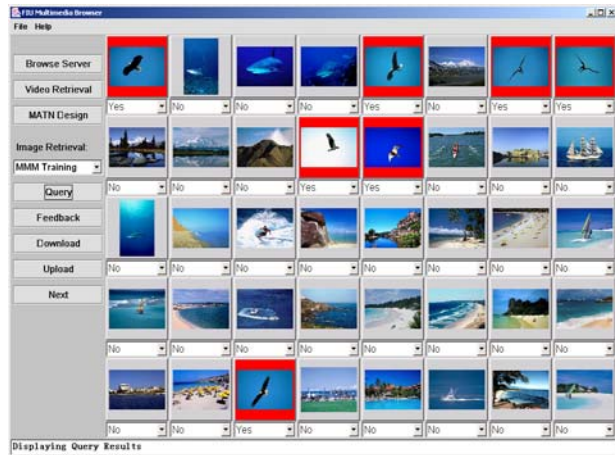

Figure 7: The interface of the training subsystem

The detailed training process is described as follows: Firstly, the user selects one query image. After clicking the "*Query*" button, a query message is sent to the server through UDP. The query results will be sent back after the server fulfills the query process. It is worth mentioning that for training purposes, any available image retrieval methods can be implemented on the server side. Upon receiving the results, the user selects the images that he/she thinks are related to the query image by right-clicking on the image canvases, and clicks the "*Feedback*" button to send the feedbacks back to the server. When the server receives and identifies this feedback message, it updates the user access patterns and access frequencies accordingly. Then the user can continue the training process or exit.

Totally, there are 1400 queries issued to the database which cover nearly 50% images in the database. As shown in our experimental results, this training data set can improve the query results dramatically. The more images covered in the training process, the higher retrieval accuracy can be achieved.

3.3 Experimental Results and Analysis

We use 80 randomly chosen query images that belong to the "landscape", "flower", "animal", "vehicle", and "human" categories (16 images per category) to test the performance and efficiency of our proposed framework. For each query image, our proposed retrieval process is executed to find a set of similar images. The degrees of matching between the query image and other images in the database are determined by the similarity values according to the S function (as shown in Equation (6)). In the following subsections, a query-by-image example is first given to demonstrate the effectiveness of our proposed approach, and then the complete performance analysis and comparison are conducted to show the performance improvement brought by the training process as well as the scalability of this framework.

3.3.1 Query-by-Image Example

As shown in Figure 8, the retrieved images are ranked and displayed in the descending order of their similarity scores from the top left to the bottom right, where the upper leftmost image is the query image marked with a red box. In this example, the query image belongs to the 'vehicle' category and contains complicated scenes. Though it contains an object (boat) with clear semantics, it is difficult to extract the foreground objects from the uneven background scenes for most of the existing region-based image retrieval systems due to the inaccuracy of the object segmentation. Figure 8 shows the snapshot of the most qualified twenty images retrieved for this query image. As can be seen from this figure, the perceptions contained in these returned images are quite similar and the ranking is reasonably good. This query example demonstrates the advantages and potentials brought by utilizing user access patterns and access frequencies.
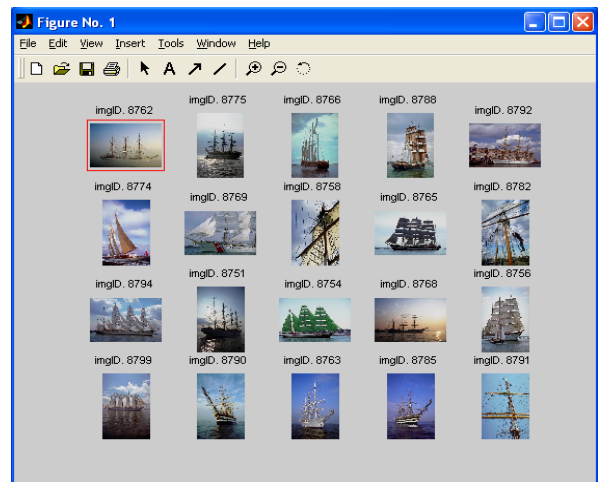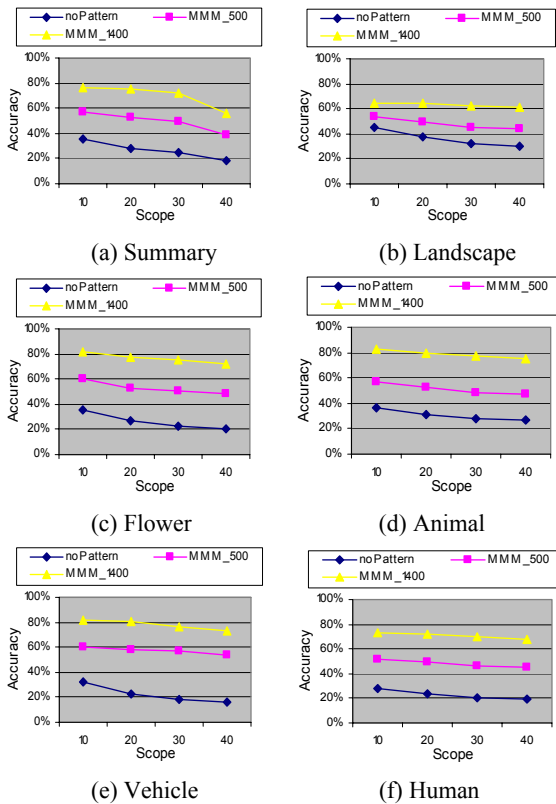

Figure 8: The snapshot of the retrieval results

3.3.2 Performance Comparison

Off-line training processes can improve the query results dramatically by capturing users' perceptions. The more images that are covered in the training process, the more accurate the retrieval results would be. In this section, we use the accuracy-scope curve to analyze the performance of our proposed framework. In the accuracy-scope curve, the scope specifies the number of images returned to the users and the accuracy is defined as the percentage of the retrieved images that are semantically related to the query image.

In our first experiment, two different training data sets are used. The first training data set includes 500 queries which cover 2034 images in our database (denoted as MMM_500); while the other set is called MMM_1400 which contains 1400 queries with 4890 images being covered. We compare the overall performance of of our proposed MMM framework with the "noPattern" method that does not integrate the information of user access patterns and access frequencies, and performs the full sequential search through the image database based on the feature matrix $\mathcal{B}$. For the pre-filtering step, we choose the size of candidate pool as 400 after PCA search space reduction, which means there are only 400 images that need to do exact similarity matching. Thus the candidate image pool constitutes only 4% of the total amount of images in database. In fact, the larger the candidate pool is used, the more accurate results can be achieved as will be shown in our second experiment.

represent the accuracy results of the noPattern method, MMM_500, and MMM_1400, respectively. From this figure, we have the following two observations: 1) Our proposed framework (MMM_500 or MMM_1400) outperforms the noPattern method in all cases, which proves that the user access patterns and access frequencies obtained from the off-line training process can capture the subjective aspects of the user concepts; and 2) With more queries issued and more images covered in the training process, the more accurate retrieval results can be achieved. The reason is that the relative affinity relationships among images can be revealed more completely, and the bias caused by individual users can be corrected (more or less) as more user feedbacks are received.

In the second experiment, using the same training data set MMM_1400, we select three different sizes for the candidate pool as 200, 400 and 800, called PCA_200, PCA_400, and PCA_800, respectively. It can be easily seen from Figure 10 that more accurate retrieval results can be achieved with a larger size of the PCA candidate pool. On the other hand, the size 400 is a reasonably good choice in terms of time and space because, using only 4% of the total images in the database, the accuracy in the top 20 retrieved images can reach 80%. Also, in most cases, the accuracies of PCA_400 are quite close to those of PCA_800.

(a) Summary   (b) Landscape

(c) Flower   (d) Animal
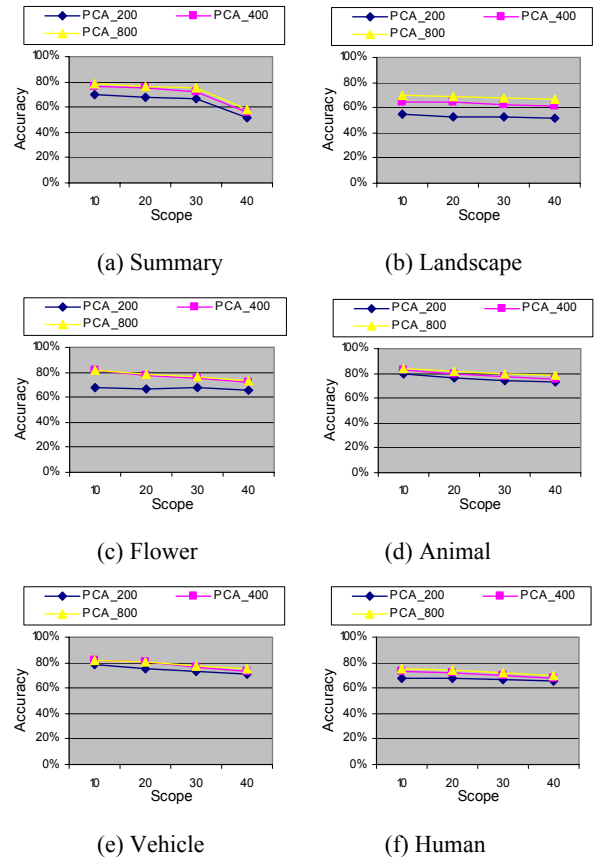
(e) Vehicle   (f) Human

Figure 9: Accuracy comparison between the proposed framework and the noPattern method

Figure 9 shows the results of the performance evaluation. In Figures 9(a)-(f), 'noPattern', 'MMM_500' and 'MMM_1400'

(a) Summary   (b) Landscape

(c) Flower   (d) Animal

(e) Vehicle   (f) Human

Figure 10: Accuracy comparison of the proposed framework against three different PCA candidate pool sizes (200, 400, and 800)

It is worth mentioning that, besides the training process, users can also provide feedback during the actual retrieval process,

which enables the accumulative learning and can further boost the system performance.

4. CONCLUSIONS

Currently, Content-Based Image Retrieval (CBIR) technology is still far from mature although it has demonstrated great potentials. In this paper, a framework based on the Markov Model Mediator (MMM) mechanism and PCA pre-filtering process is proposed for efficient content-based image database retrieval. In our proposed framework, both the low-level features (e.g., color) and mid-level features (e.g., object location) are supported, and the utilization of user access patterns and access frequencies enables the system to integrate the high-level user concepts into the retrieval process and to produce the more accurate results for user queries. Experiments were conducted to demonstrate the effectiveness and efficiency of our proposed framework, and the experimental results exemplify these advantages and the overall retrieval performance of the presented system is promising. Currently, this framework is tested within a single database. In our future work, we will apply the MMM mechanism to facilitate efficient search across multiple image databases in a distributed environment, where the mediator concept is used to improve the query processing performance by analyzing and discovering the summarized knowledge at the database level.

5. ACKNOWLEDGEMENT

6. REFERENCES

[1] Stehling, R.O., Nascimento, M.A., and Falcao, A.X. On Shapes of Colors for Content-Based Image Retrieval. ACM International Workshop on Multimedia Information Retrieval (ACM MIR'00), Los Angeles, CA, USA, 2000, 171-174.

[2] Zhang, D.S. and Lu, G. Generic Fourier Descriptors for Shape-Based Image Retrieval. Proceedings of IEEE International Conference on Multimedia and Expo (ICME'02), Vol. 1, Lausanne, Switzerland, August 2002, 425-428.

[3] Kaplan, L.M., et al. Fast Texture Database Retrieval Using Extended Fractal Features. Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Media Databases, 1998, 162-173.

[4] Jing, F., Li, M., Zhang, H.-J., and Zhang, B. An Effective Region-based Image Retrieval Framework. Proceedings of the 8th ACM International Conference on Multimedia (MM'00), Juan-les-Pins, France, December 1-6, 2002, 456-465.

[5] Rui, Y., Huang, T.S., and Mehrotra, S. Content-based Image Retrieval with Relevance Feedback in MARS. Proceedings of the 1997 International Conference on Image Processing (ICIP'97) (3-Volumn Set), 1997, 815-818.

[6] Ishikawa, Y., Subramanya, R., and Faloutsos, C. Mindreader: Query Databases through Multiple Examples. Proceedings of the 24th VLDB Conference (VLDB'98), New York City, USA, 1998, 218-227.

[7] Wiederhold, G. Mediators in the architecture of future information systems. IEEE Computers, March 1992, 38-49.

[8] Frank, O., and Strauss, D. Markov graphs. Journal of the American Statistical Association, Vol. 81, 1986, 832-842.

[9] Rabiner, L. R., and Huang, B. H. An Introduction to Hidden Markov Models. IEEE ASSP Magazine, Vol. 3, No. 1, January 1986, 4-16.

[10] Naphade, M. R., and Huang, T. S. A Probabilistic Framework for Semantic Video Indexing, Filtering and Retrieval. IEEE Transactions on Multimedia, Vol. 3, No. 1, March 2001, 141-151.

[11] Shyu, M.-L., Chen, S.-C., and Kashyap, R. L. A Probabilistic-Based Mechanism for Video Database Management Systems. Proceedings of IEEE International Conference on Multimedia and Expo (ICME'00), New York City, USA, July 30-August 2, 2000, 467-470.

[12] Shyu, M.-L., Chen, S.-C., Haruechaiyasak, C., Shu, C.-M., and Li, S.-T. Disjoint Web Document Clustering and Management in Electronic Commerce. Proceedings of the 7th International Conference on Distributed Multimedia Systems (DMS'01), Tamkang University, Taipei, Taiwan, September 26-28, 2001, 494-497.

[13] Ma, W.-Y., and Zhang, H. J. Content-Based Image Indexing and Retrieval. Handbook of Multimedia Computing, CRC Press, Chapter 13, 1999.

[14] Chen, S.-C., Sista, S., Shyu, M.-L., and Kashyap, R.L. An Indexing and Searching Structure for Multimedia Database Systems. Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Media Databases, 2000, 262-270.

[15] Guttman, A. R-tree: A Dynamic Index Structure for Spatial Search. Proceedings of ACM SIGMOD, June 1984, 47-57.

[16] Jobson, J. D. Applied Multivariate Data Analysis Volume II: Categorical and Multivariate Methods. Springer-Verlag Inc., NY, 1992.

[17] Johnson, R. A., and Wichern, D. W. Applied Multivariate Statistical Analysis. 4th Ed., Prentice-Hall, NJ, 1998.

[18] Su, Z., Li, S., and Zhang, H. Extraction of Feature Subspaces for Content-Based Retrieval Using Relevance Feedback. Proceedings of the 9th ACM International Conference on Multimedia (MM'01), Ottawa, Canada, September 30 - October 5, 2001, 98-106.

[19] Cheng, H. D., and Sun, Y. A Hierarchical Approach to Color Image Segmentation Using Homogeneity. IEEE Transactions on Image Processing, Vol. 9, No. 12, December 2001, 2071-2082.

[20] Chen, S.-C., Shyu, M.-L., Zhao, N., and Zhang, C. An Affinity-Based Image Retrieval System for Multimedia Authoring and Presentation. Accepted for publication, the 11th ACM International Conference on Multimedia (MM'03), Berkeley, CA, USA, November 2-8, 2003.