MMM: A Stochastic Mechanism for Image Database Queries

Mei-Ling Shyu
Department of Electrical & Computer Engineering
University of Miami
Coral Gables
FL 33124, USA
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
{chens, mchen005, czhang02}@cs.fiu.edu

Chi-Min Shu
Department of Environmental and Safety Engineering
National Yunlin University of Science and Technology
Yunlin, Taiwan, R.O.C.
shucm@pine.yuntech.edu.tw

Abstract

In this paper, we present a mechanism called the Markov Model Mediator (MMM) to facilitate the effective retrieval for content-based image retrieval (CBIR). Different from the common methods in content-based image retrieval, our stochastic mechanism not only takes into consideration the low-level image content features, but also learns high-level concepts from a set of training data, such as access frequencies and access patterns of the images. The advantage of our proposed mechanism is that it exploits the structured description of visual contents as well as the relative affinity measurements among the images. Consequently, it provides the capability to bridge the gap between the low-level features and high-level concepts. Our experimental results demonstrate that the MMM mechanism can effectively assist in retrieving more accurate results for user queries.

1. Introduction

The availability of today's digital devices and techniques offers users more opportunities than ever to create their own digital images. Moreover, the Internet has become the biggest platform to get, distribute and exchange digital image data. The rapid increase in the amount of image data and the inefficiency of text-based image retrieval have created great demands for new approaches in image retrieval.

Currently, Content-Based Image Retrieval (CBIR) becomes an active research area. The objective of a CBIR system is to offer the user an efficient way in finding and retrieving those images that are qualified for the matching criteria of users' queries from the database. In contrast to the text-based approach, CBIR operates on a totally different principle, i.e., to retrieve the stored images from a collection of images by comparing the features that were automatically extracted from the images themselves. It involves a matching process between a query image and the images stored in the database. The first step of the process involves automatically extracting a feature vector for the unique characteristics of each image. A quantified similarity value between two images is obtained by comparing their feature vectors. The commonly used image features include color [19], shape [23] and texture [9]. Queries are issued through query-by-image example (QBE). A lot of research work has been done, which resulted in a number of systems and techniques, both in the academic and commercial domain. For example, IBM's QBIC system [5] and Virage's VIR engine [20] are two most notable commercial image retrieval systems, while VisualSEEk [18], Metaseek [1], PhotoBook [13] are well-known academic image retrieval systems. In addition, a prototype content-based image retrieval system, PicHunter, was presented in [4].

However, an impediment to research on CBIR is the lack of mapping between the high-level concepts and the low-level features. In order to overcome this problem and to better capture the subjectivity of human perception of the visual content, the concept of relevance feedback (RF) associated with CBIR was proposed in [15]. Relevance feedback is an interactive process in which the user judges the quality of the retrieval results performed by the system by marking those images that the user perceives as truly relevant among the images retrieved by the system. This information is then used to refine the original query. Most of the relevance feedback research work has been carried out in two approaches: query point movement and re-weighting [8]. However, this process should be dealt with in a real-time manner in the loop because the metric dynamically depends upon the user's feedback and the context. The real-time learning of distance metric or feature space transformations based on the users' interactions remains an open issue. On the other hand, in order to get better results, the user may be asked to browse a bunch of images through iterations and to provide the detailed ranking of similarity for the images. The fact is that a heavy and unnecessary burden of responsibility is brought to the user. In addition, it is highly probable that this burden will have a negative effect on user's perception of the effectiveness and efficiency of the system.

This paper presents a content-based retrieval system that employs the Markov Model Mediator (MMM) mechanism to retrieve images, which functions as both the searching engine and image similarity arbitrator to facilitate the functionality of a multimedia database management system. In our previous studies, the MMM mechanism has been applied to multimedia database management [17] and document management on the World Wide Web (WWW) [16]. The MMM mechanism adopts the Markov model framework and the concept of the mediators. The Markov mechanism is one of the most powerful tools available to scientists and engineers for analyzing complicated systems, whereas a mediator is defined as a program to collect and combine information from one or more sources, and finally yield the resulting information [21]. Markov models have been used in many applications. Some well-known examples are Markov Random Field Models [6], and Hidden Markov Models (HMMs) [14]. Some research works have been done to integrate the Markov model into the field of image retrieval. Lin *et al*. [10] used a Markov model to combine the spatial and color information. In their approach, each image in the database is represented by a pseudo two-dimensional Hidden Markov Model (HMM) in order to adequately capture both the spatial and chromatic information about that image. [22] used the Hidden Markov Model (HMM) to parse video data. In [12], the Hidden Markov Model was employed to model the time series of the feature vector for the cases of events and objects in their probabilistic framework for semantic level indexing and retrieval.

In this paper, the MMM mechanism is applied to the dynamic content-based image retrieval process. Our method also builds an index vector for each image within the database and considers the relationship between the query image and the target image. However, unlike the common methods mentioned earlier, we capture the users' perceptions by using a set of training data, such as access patterns and access frequencies of the images in the image database. Also, instead of on-line learning user's preferences, interpretations or retrieval requirements in RF, we calculate the relative affinities among the images in the image database off-line, which serve as the factors of the high-level concepts in our system.

The remainder of this paper is organized as follows. Section 2 reviews the key components of the MMM mechanism and introduces the stochastic process for information retrieval. Section 3 presents our experiments in applying the MMM mechanism to content-based image retrieval. The experimental results demonstrate that the MMM mechanism can assist in retrieving more accurate results for user queries. A brief conclusion is given in Section 4.

2. The stochastic model

2.1. Markov Model Mediator (MMM) mechanism

Markov Model Mediator (for short, MMM) is a probabilistic-based mechanism that adopts the Markov model framework and the mediator concept. The MMM mechanism is defined as follows:

Definition 1: An MMM is represented by a 5-tuple $\lambda = (\mathcal{S}, \mathcal{F}, \mathcal{A}, \mathcal{B}, \Pi)$, where $\mathcal{S}$ is a set of images called states; $\mathcal{F}$ is a set of distinct features of the images; $\mathcal{A}$ denotes the states transition probability distribution, where each entry (i, j) actually indicates the relationship between image i and j captured through the off-line training processes; $\mathcal{B}$ is the feature matrix; and Π is the initial state probability distribution.

Each image database in our CBIR system is modeled by an MMM, where $\mathcal{S}$ consists of all the images in the image database and $\mathcal{F}$ includes all the distinct features for the images in $\mathcal{S}$. $\mathcal{A}$ represents the relationships among all the images in the database based on user's preference, and the relationships of the images are modeled by the sequences of the MMM states connected by transitions. $\mathcal{B}$ consists of the normalized image feature vectors for all the images. The last tuple Π indicates how likely an image would be accessed without knowing the query image. A training data set consisting of the access patterns and access frequencies of the queries issued to the database is used to train the model parameters $\mathcal{A}$, $\mathcal{B}$, and Π for an MMM.

2.2. Formulation of the model parameters

In each MMM, its model parameters $\mathcal{A}$, $\mathcal{B}$, and Π that are critical for the stochastic process need to be formulated and constructed. For this purpose, a set of training data is used.

2.2.1. Training data set. The training data set is used to generate the training concepts off-line for an MMM mechanism to construct its model parameters $\mathcal{A}$, $\mathcal{B}$, and Π matrices. Definition 2 gives the information available in the training data set.

Definition 2: The training data set consists of the following information:

- The value N that indicates the number of images in database d.
- A set of queries $Q = \{q_1, q_2, \ldots, q_q\}$ that are issued to the database in a period of time. Let $use_{m,k}$ denote the usage pattern of image m with respect to query q_k per time period, where the value of $use_{m,k}$ is 1 when m is accessed by q_k and zero otherwise. The value of $access_k$ denotes the access frequency of query q_k per time period.

The pair of user access pattern ($use_{m,k}$) and user access frequency ($access_k$) provides the capability to capture the user concepts in the training process, which can be seen from the following example.

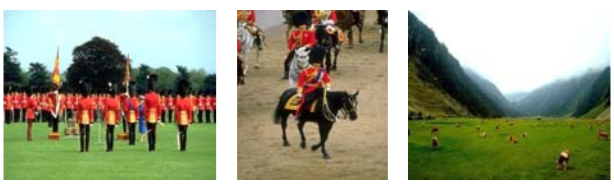

Image 1 Image 2 Image 3

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

Table 1. The user access pattern ($use_{m,k}$) for the sample images

	Image 1	Image 2	Image 3	…
q_1	1	1	0	…
q_2	0	0	1	…
q_3	1	0	1	…
…	…	…	…	…

Table 1 gives three example queries issued to our image database, where q_1 is a user-issued query related to retrieving some parade scenes, and q_2 and q_3 are two queries interested in the concept of natural scenes with green grass and blue sky. The corresponding access frequencies $access_k$ for q_1, q_2, and q_3, which are accumulated by recording different users' selections during the training process, are 21, 19 and 2, respectively. In Table 1, the entry (k, m) = 1 indicates that the m^{th} image is accessed by query q_k. For example, image 1 and image 2 are accessed together in q_1 with their corresponding entries in the user pattern matrix having value 1 and the access frequency $access_1$ is 21. For the query related to the natural scenes, two different access patterns are recorded with $access_2$ equals to 19 and $access_3$ equals to 2, which is due to the different user concepts (parade scene or landscape scene) existed for image 1. However, since most of the users regard image 1 as a parade scene more than a landscape scene, only image 3 is accessed in most cases, which is indicated by a larger $access_2$ value. Though it is possible that image 1 would be selected together with image 3 as natural scenes as recorded in q_3, obviously the value of $access_3$ (i.e., 2) is much smaller than the value of $access_2$ (i.e., 19). Consequently, after the system training, image 2 is more likely to be retrieved than image 3, given image 1 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. Matrix $\mathcal{A}$: the state transition probability distribution. Based on the information in the training data set, we can capture the relationships among the images in the database based on the high-level concepts. That is, the more frequently two images are accessed together, the more closely they are related. In order to capture the relative affinity measurements among all the images, a matrix $\mathcal{AF}$ is defined, which is constructed by having the $aff_{m,n}$ be the relative affinity relationship between two images m and n using the following definition.

Definition 3: The relative affinity measurement ($aff_{m,n}$) between two images m and n indicates how frequently these two images are accessed together, where

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

The state transition probability distribution (matrix $\mathcal{A}$) is constructed by having $a_{m,n}$ be the element in the $(m, n)^{th}$ entry in $\mathcal{A}$, where

$$a_{m,n} = \frac{aff_{m,n}}{\sum_{n \in d} aff_{m,n}} \quad (2)$$

As shown in this formulation, matrix $\mathcal{A}$ is obtained via normalizing $\mathcal{AF}$ per row and represents the conditional probability that refers to as the state transition probability for an MMM.

2.2.3. Matrix $\mathcal{B}$: the feature matrix. For the feature matrix $\mathcal{B}$, we consider the following features: color information and object location information for the images in the image database. Since the color feature is closely associated with image scenes and it is more robust to changes due to scaling, orientation, perspective and occlusion of images, it is the most widely used visual feature in image retrieval [11]. In our CBIR system, color information is obtained for each image from its HSV color space. The HSV color space is chosen for two reasons. First, it is perceptual, which makes HSV a proven color space particularly amenable to color image analysis [3]. Secondly, the benchmark results in [11] showed that the color histogram in the HSV color space performs the best. For object location information, the SPCPE algorithm proposed in [2] is used. The minimal bounding rectangle (MBR) concept in R-tree [7] is adopted so that each object is covered by a rectangle. The centroid point of each object is used for space reasoning so that any object is mapped to a point object.

Since our focus is to evaluate the performance of the MMM retrieval mechanism and to reduce the feature space rather than to explore the most appropriate features for image retrieval, in our study, each image has a feature vector of only twenty-one elements. Within the twenty-one features, twelve are for color descriptions and nine are for location descriptions. The color features considered are '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) according to the combinations of different ranges of the hue (H), saturation (S), and the intensity values (V). Colors with the number of pixels less than 5% of the total number of pixels are regarded as non-important and the corresponding positions in the feature vector have the value 0. Otherwise, we put the corresponding percentage of that color component to that position.

For the location descriptions, each image is divided into 3×3 equal-sized regions. The image can be divided into a coarser or finer set of regions if necessary. As shown in Figure 2, the nine regions are ordered from left to right and top to bottom: L1, L2, L3, L4, L5, L6, L7, L8, and L9. When there is an object in the image whose centroid falls into one of the nine regions, the value 1 is assigned to that region. Objects with their areas less than 8% of the total area are ignored.

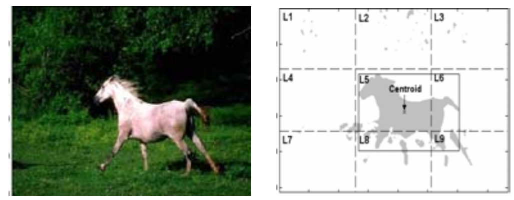

Image 4

Figure 2. Object locations and their corresponding regions

In order to capture the appearance of a feature in an image, we define a temporary matrix ($\mathcal{BB}$) whose rows are all the distinct images and columns are all the distinct features, where the value in the $(p, q)^{th}$ entry is greater than zero if feature q appears in image p, and zero otherwise. Table 2 illustrates the associated feature vectors of the sample images in Figures 1 and 2.

Then the feature matrix $\mathcal{B}$ can be obtained via normalizing $\mathcal{BB}$ per row as shown in Table 3. We consider that the color and location information are of equal importance, such that the sum of the feature values of the color features should be equal to that of the location features. In other words, the sum of the values that the features are observed from a given image should be 1, with 0.5 each for color features and location features. It is worth mentioning that our mechanism is robust in terms of the update to the feature matrix $\mathcal{B}$. Users are allowed to construct the matrix by using any normalized vector-based image feature set.

Table 2. $\mathcal{BB}$ matrix - 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	0.38	0.12	0	0	0	0.36	0	0	0	0	0	0	0	0	0	1	0	0	0	0
image 2	0.11	0.35	0.06	0.36	0.12	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
image 3	0.08	0.40	0	0	0.07	0.29	0.07	0	0.07	0	0	0	0	0	0	0	1	0	0	0	0
image 4	0.49	0.08	0	0	0	0	0.43	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Table 3. $\mathcal{B}$ 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	0.07	0.19	0.06	0	0	0	0.18	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0
image 2	0.05	0.18	0.03	0.18	0.06	0	0	0	0	0	0	0	0	0.5	0	0	0	0	0	0	0
image 3	0.04	0.20	0	0	0.03	0.15	0.04	0	0.04	0	0	0	0	0	0	0	0.5	0	0	0	0
image 4	0.25	0.04	0	0	0	0	0.21	0	0	0	0	0	0	0	0	0	0.5	0	0	0	0

2.2.4. Matrix Π: the initial state probability distribution. The preference of the initial states for queries can be obtained from the training data set. For any image $m \in d$, the initial state probability is defined as the fraction of the number of occurrences of image m with respect to the total number of occurrences for all the images in the image database d from the training data set.

$$\Pi = \{\pi_m\} = \frac{\sum_{k=1}^{q} use_{m,k}}{\sum_{l=1}^{N} \sum_{k=1}^{q} use_{l,k}} \quad (3)$$

2.3. Stochastic process for information retrieval

The need for efficient information retrieval from the databases is strong. However, as we mentioned earlier, usually the cost for query processing is expensive and time-consuming. In addition, the results may not be very satisfactory due to the lack of mapping between the high-level concepts and the low-level features. Probabilistic models offer a way to perform the searching process more efficiently and accurately. Based on the concepts learned from the training data set, we capture the most matched images through a dynamic programming algorithm that conducts a stochastic process in calculating the current edge weights and the cumulative edge weights.

Assume N is the total number of images in the database, and the features of the query image S_{query} is denoted as $\{o_1, o_2, \ldots, o_T\}$, where T is the total number of non-zero features of the query image S_{query}. In our case, $1 \leq T \leq 21$ since there are 21 features in total.

Definition 4: $W_t(i)$ is defined as the edge weight from the state S_i to S_{query} at the evaluation of the t^{th} feature (o_t) in the query, where $1 \leq i \leq N$ and $1 \leq t \leq T$.

Definition 5: $D_t(i)$ is defined as the cumulative edge weight from the state S_i to S_{query} at the evaluation of the t^{th} feature (o_t) in the query, where $1 \leq i \leq N$ and $1 \leq t \leq T$.

Based on definitions 4 and 5, the dynamic programming algorithm is given as follows.

At $t = 1$,

$$W_1(i) = \pi_{S_i} b_{S_i}(o_1) \quad (4)$$

$$D_1(i) = W_1(i) \quad (5)$$

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

$$W_{t+1}(i) = D_t(i) a_{S_{query} S_i} b_{S_i}(o_{t+1}) \quad (6)$$

$$D_{t+1}(i) = (\max_i \; D_t(i)) + W_{t+1}(i) \quad (7)$$

As we mentioned before, $\mathcal{A} = \{a_{S_i S_j}\}$ denotes the states transition probability distribution, $\mathcal{B} = \{b_{S_i}(o_k)\}$ is the feature matrix, and is the initial state probability distribution. The image retrieval steps using the dynamic programming algorithm in the stochastic process are shown in Table 4.

Table 4. Image retrieval steps using our proposed stochastic model

1. Given the query image *q*, obtain its feature vector $\{o_1, o_2, \ldots, o_T\}$, where T is the total number of non-zero features of the query image *q*.
2. Upon the first feature o_1, calculate $W_1(i)$ and $D_1(i)$ according to Equations (4) and (5).
3. Move on to calculate $W_2(i)$ and $D_2(i)$ according to Equations (6) and (7).
4. Continue to calculate the next values for the *W* and *D* vectors until all the features in the query have been taken care of.
5. Upon each feature in query, we can obtain a pair of vectors: $W_t(i)$ and $D_t(i)$. We then sum up each value at the same position in the vectors $W_1(i)$, $W_2(i)$, …, $W_T(i)$. Namely, $sumW_T(i) = \Sigma_T W_t(i)$ is calculated.
6. Find the candidate images by sorting their corresponding values in $sumW_T(i)$. The bigger the value is, the stronger the relationship exists between the candidate image and the query image.

In Step 3, since we already obtained matrices $W_1(i)$ and $D_1(i)$ from Step 2, and the second feature o_2 is known, the content of $W_2(i)$ and $D_2(i)$ can be determined. Following the same way, all the pairs of W and D vectors can be obtained. The value of $sumW_T(i)$ (obtained in Step 5) is the sum of the edge weights $W_1(i)$, $W_2(i)$, … , $W_T(i)$. In other words, it indicates the matching percentage of the i^{th} image in the image database to the query image S_{query} with respect to the features $\{o_1, o_2, \ldots, o_T\}$.

In contrast to the common methods which either have difficulties to capture the high-level concepts or try to learn the concepts in real-time, our method provides the capability of training the data set off-line. On the other hand, because given a query image *q* issued by a user, only the data in the row *q* of matrix $\mathcal{A}$ are used, most of the values in the entries of the $\mathcal{B}$ matrix are zeros, and normally the features contained in one query image is no more than six, we can retrieve the results more accurately and efficiently.

3. Experiments

3.1. Experimental image database system

In our image database, there are 10,000 color images of 72 semantic categories with various dimensions that are used to carry out the experiments. Both the color information and object location information of the images are considered and the query-by-example strategy is used to issue queries in our experiments. In addition, with the purpose of supporting high-level meaning in the queries, first we need to construct the model parameters for the MMM mechanism based on the training data set (i.e., access patterns and access frequencies) obtained from the 150 training queries issued to the image database.

3.2. Constructions of the model parameters

Each MMM has three probability distributions ($\mathcal{A}$, $\mathcal{B}$, and Π). The state transition probability distribution $\mathcal{A}$ can be obtained according to Equations (1) to (2) given in Section 2. In order to calculate $\mathcal{B}$, first we need to construct $\mathcal{BB}$ based on the images and their features in the experimental database. Based on $\mathcal{BB}$, $\mathcal{B}$ can be obtained using the procedure illustrated in Section 2. The initial state probability distribution for experimental database can be determined by using Equation (3). The constructions of these model parameters can be performed off-line.

Once the model parameters of the MMM for the image database are constructed, the stochastic process shown in Table 4 is used for image retrieval.

3.3. Experiments

To test the performance and efficiency of our proposed mechanism, 80 randomly chosen images belonging to 5 distinct categories: landscape, flower, animal, vehicle and human, are used as the query images, with 16 images per category. For a given query image issued by a user, the stochastic process with the proposed dynamic programming algorithm will be carried on to dynamically find the matched images for the user's query. The qualifying degrees of the images with respect to the certain query image are determined by the values in the resulting $sumW_T$ vectors according to the rules described in Table 4. In the following subsections, a query-by-image example is first used to demonstrate the effectiveness of our stochastic model. As mentioned earlier, the off-line training process can improve the query results dramatically by capturing users' perceptions. In addition, the feature matrix $\mathcal{B}$ can be altered without affecting other components in our MMM model. In order to demonstrate the performance improvement contributed by the training process and the flexibility of our model, we use the accuracy-scope curve to compare the performance of our proposed mechanism with the 'noPattern' method. 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.

3.3.1. Query-by-image example. In this experiment, 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. In this example, the query image belongs to the 'Landscape' category and is complicated to analyze because, though it contains clear semantic meanings, it is hard to extract the foreground object from the background. Figure 3 shows the snapshot of the retrieval result screen containing the most qualified twelve images to this query image from the database. As can be seen from this figure, the perceptions contained in these returned images are quite similar and the ranking is reasonably good.

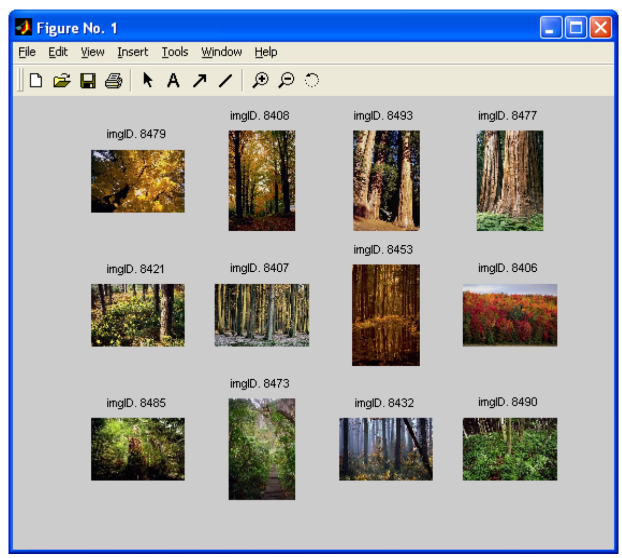

Figure 3. The snapshot of the retrieval results

3.3.2. Performance comparison. In our second experiment, we compare the overall performance of our proposed MMM mechanism with the so-called noPattern method. The noPattern method does not integrate the information of user access pattern and access frequency, and performs the full sequential search through the image database based on the feature matrix $\mathcal{B}$. Then in our third experiment, we alter the feature matrix $\mathcal{B}$ by assigning different weights, 0.1 and 0.9 respectively, to the color features and location features instead of 0.5 each. It is worth mentioning that the weights are chosen randomly and in fact any normalized vector-based image feature set can be plugged into the matrix $\mathcal{B}$. Then the performance comparison is conducted in the same way as the second experiment.

Figure 4 shows the results of the performance evaluation. In Figures 4(a)-(f), 'MMM' and 'noPattern' indicate the accuracy results of MMM mechanism and the noPattern method in our second experiment, while 'MMM-Update' and 'noPattern-Update' are for the third experiment. The results in Figure 4(a) are calculated using the averages of all the 80 query images, while Figures 4(b)-(f) show the results for each category.

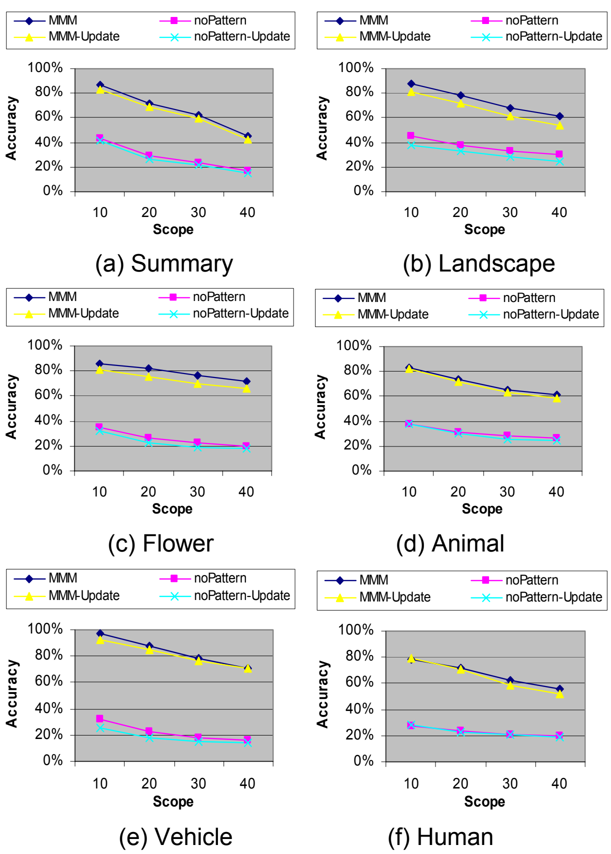

Figure 4: Accuracy comparison between the MMM mechanism and noPattern method

From Figure 4, we have the following observations. First, our proposed MMM mechanism outperforms the noPattern method in all cases. This proves that the use of the user access patterns and access frequencies obtained from the off-line training process can capture the subjective aspects of the user concepts. Second, our model is flexible enough to use different feature vector sets. Third, the MMM model and the noPattern method share almost the same trend, which implies that the more effective feature vectors are extracted from the images, the higher accuracy our model can achieve.

4. Conclusion

Currently, Content-Based Image Retrieval (CBIR) technology is still immature but has great potential. In this paper, a review of the recent efforts and techniques in CBIR is given, followed by the discussion of the current problems in the CBIR systems from the concern of lacking the mapping between the high-level concepts and the low-level features. Though Relevance Feedback has been proposed to overcome this problem recently, it requires the concepts to be trained in real-time and the users are required to take heavy responsibilities during the retrieval process. In response to these issues, the Markov Model Mediator (MMM) mechanism is applied to the image databases for content-based image retrieval. A stochastic process based on the MMM mechanism is proposed to traverse the database and find the similar images with respect to the query image. Our proposed stochastic-based mechanism provides the capability to learn the concepts off-line based on the training data set, such as access patterns and access frequencies. Then it performs the similarity comparison between the query image and the target image based on not only the low-level features, such as color and location features, but also the concepts obtained by training in the previous steps among all the images in the image database. The fact that the proposed stochastic content-based CBIR system utilizes the MMM mechanism and supports both spatial and color information offers more flexible and accurate results for user queries. The experimental results exemplify this point, and the overall retrieval performance of the presented system is promising.

Acknowledgement

For Shu-Ching Chen, this research was supported in part by NSF HRD-0317692, NSF EIA-0220562, and the office of the Provost/FIU Foundation. For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260.

References

[1] M. Beigi, A. Benitez, and S.-F. Chang, "Metaseek: A Content-Based Meta Search Engine for Images," *Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases*, San Jose, CA, Jan. 1998, pp. 118-128.

[2] S.-C. Chen, S. Sista, M.-L. Shyu, and R.L. Kashyap, "An Indexing and Searching Structure for Multimedia Database

Systems," *Proceedings of IS&T/SPIE Conference on Storage and Retrieval for Media Databases*, 2000, pp. 262-270.

[3] H.D. Cheng and Y. Sun, "A Hierarchical Approach to Color Image Segmentation Using Homogeneity," *IEEE Transactions on Image Processing*, vol. 9, no. 12, 2000, pp. 2071-2082.

[4] I.J. Cox, M.L. Miller, T.P. Minka, T.V. Papathornas, and P.N. Yianilos, "The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments," *IEEE Transactions on Image Processing*, vol. 9, issue 1, Jan. 2000, pp. 20-37.

[5] M. Flickner, et al., "Query by Image and Video Content: The QBIC System," *IEEE Computer*, vol. 28, no. 9, Sep. 1995, pp. 23-31.

[6] O. Frank and D. Strauss, "Markov Graphs," *Journal of the American Statistical Association*, 81, 1986, pp. 832-842.

[7] A. Guttman, "R-tree: A Dynamic Index Structure for Spatial Search," *Proceedings of ACM SIGMOD*, Jun. 1984, pp. 47-57.

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

[9] L.M. Kaplan, et al, "Fast Texture Database Retrieval Using Extended Fractal Features," *Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Database*, 1998, pp. 162-173.

[10] H. C. Lin, et al, "Color Image Retrieval Based On Hidden Markov Models," *IEEE Transactions on Image Processing*, vol. 6, no. 2, 1997, pp. 332-339.

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

[12] M. R. Naphade and T. S. Huang, "A Probabilistic Framework for Semantic Video Indexing, Filtering and Retrieval," *IEEE Transactions on Multimedia*, vol. 3, no. 1, March 2001, pp. 141-151.

[13] A. Pentland, R.W. Picard, and S. Sclaroff, "Photobook: Tools for Content-Based Manipulation of Image Databases," *Proceedings of IS&T/SPIE Storage and Retrieval for Image and Video Databases II*, vol. 2185, Bellingham, Washington, 1994, pp.34-47.

[14] L. R. Rabiner and B. H. Huang, "An Introduction to Hidden Markov Models," *IEEE ASSP Magazine*, vol. 3, no. 1, Jan. 1986, pp. 4-16.

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

[16] M.-L. Shyu, S.-C. Chen, and C.-M. Shu, "Affinity-Based Probabilistic Reasoning and Document Clustering on the WWW," *Proceedings of the 24th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC)*, Taipei, Taiwan, Oct. 2000, pp. 149-154.

[17] M.-L. Shyu, S.–C. Chen, and R. L. Kashyap, "A Probabilistic-Based Mechanism for Video Database Management Systems," *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2000)*, New York City, USA, Jul. 2000, pp. 467-470.

[18] J. R. Smith and S. F. Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System," *Proceedings of ACM International Conference on Multimedia*, Boston, Nov. 1996, pp. 87-98.

[19] R.O. Stehling, M.A. Nascimento, and A.X. Falcao, "On Shapes of Colors for Content-Based Image Retrieval," *Proceedings of the 2000 ACM workshops on Multimedia*, Los Angeles, California, USA, 2000, pp. 171-174.

[20] http://www.virage.com

[21] G. Wiederhold, "Mediators in the Architecture of Future Information Systems," *IEEE Computers*, Mar. 1992, pp. 38-49.

[22] W. Wolf, "Hidden Markov Model Parsing of Video Programs," *Proceedings of the International Conference of Acoustics, Speech and Signal Processing*, Munich, Germany, Apr. 1997, pp. 2609-2611.

[23] D.S. Zhang, and G. Lu, "Generic Fourier Descriptors for Shape-Based Image Retrieval," *Proceedings of IEEE International Conference on Multimedia and Expo (ICME2002)*, Lausanne, Switzerland, Aug. 2002, pp. 425-428.