An Integrated and Interactive Video Retrieval Framework with Hierarchical Learning Models and Semantic Clustering Strategy

Na Zhao[1], Shu-Ching Chen[1], Mei-Ling Shyu[2], Stuart H. Rubin[3]
[1]*Distributed Multimedia Information System Laboratory*
School of Computing and Information Sciences
Florida International University, Miami, FL 33199, USA
[2]*Department of Electrical and Computer Engineering*
University of Miami, Coral Gables, FL 33124, USA
[3]*Space and Naval Warfare Systems Center (SSC), San Diego, CA 92152-5001, USA*
[1]*{nzhao002, chens}@cs.fiu.edu,* [2]*shyu@miami.edu,* [3]*stuart.rubin@navy.mil*

Abstract

In this research, we propose an integrated and interactive framework to manage and retrieve large scale video archives. The video data are modeled by a hierarchical learning mechanism called HMMM (Hierarchical Markov Model Mediator) and indexed by an innovative semantic video database clustering strategy. The cumulated user feedbacks are reused to update the affinity relationships of the video objects as well as their initial state probabilities. Correspondingly, both the high level semantics and user perceptions are employed in the video clustering strategy. The clustered video database is capable of providing appealing multimedia experience to the users because the modeled multimedia database system can learn the user's preferences and interests interactively.

1. Introduction and Related Work

With the recent advances in multimedia technologies, the number of multimedia files and archives increase dramatically. Therefore, it becomes an important research topic to mine and cluster the multimedia data, especially to accommodate the requirements of video retrieval in a distributed environment. Since the multimedia databases may be distributed geographically through the local network or world-wide Internet, the associated workloads could be quite expensive when dealing with complicated video queries. In particular, semantic based video retrieval is multi-disciplinary and involves the integration of visual/audio features, temporal/spatial relationships, semantic events/event patterns, high-level user perceptions, etc. Therefore, it is expected to utilize a conceptual database clustering technique to index and manage the multimedia databases such that the related data can be retrieved together and furthermore the communication costs in the query processing can be significantly reduced.

Currently, there exist approaches focusing on the clustering techniques for the video data. For example, a hierarchical clustering method for sports video was presented [1]. Two levels of clusters are constructed where the top level is clustered by the color feature and the bottom level is clustered by the motion vectors. [2] describes a spectral clustering method to group video shots into scenes based on their visual similarity and temporal relationships. In [5], the algorithms are proposed for unsupervised discovery of the video structure by modeling the events and their stochastic structures in video sequences via using Hierarchical Hidden Markov Models (HHMM). Based on our best knowledge, most of the existing researches produce the clusters mainly on low-level and/or mid-level features, and do not consider high-level concepts or user perceptions in the clustering procedure. This brings the problem of "semantic gap". Relevance feedback is an effective method to narrow down this semantic gap. However, most of the existing relevance feedback systems are only capable of providing real-time updates on the retrieval results without any further improvement of the overall system performance. In addition, multimedia databases may not be efficiently modeled in these approaches even after the clustering technique is applied.

In this paper, an integrated and interactive video retrieval framework is proposed to efficiently organize, model, and retrieve the content of a large scale multimedia database. The core of our proposed framework is a learning mechanism called HMMM (Hierarchical Markov Model Mediator) [6] and an innovative video clustering strategy. HMMM models the video database; while the clustering strategy groups video data with similar characteristics into clusters that exhibit certain high level semantics. The HMMM structure is then extended by adding an additional level to represent the clusters and their relationships.

The proposed framework is designed to accommodate advanced queries via considering the high level semantic meaning. First of all, it is capable of searching semantic

0-7803-9788-6/06/$20.00

events or event patterns considering their popularity by evaluating their access frequencies in the large amount of historical queries. Second, the users can choose one or more example patterns with their anticipated features from the initial retrieved results, and then issue the next round of query. It can search and re-rank the candidate patterns which involve the similar aspects with the positive examples reflecting the user's interests. Third, video clustering can be conducted to further reduce the searching time especially when dealing with the top-*k* similarity retrievals. As the HMMM mechanism helps to traverse the most optimized path to perform the retrieval, the proposed framework can only search several clusters for the candidate results without traversing all the paths to check the whole database.

This paper is organized as follows: Section 2 presents the overall framework of our proposed research. In Section 3, the detailed techniques are further expanded by introducing HMMM model and explaining the clustering strategy. Moreover, the retrieval algorithm and example are also included. Section 4 analyzes the experimental results. Finally, conclusions are summarized in Section 5.

2. Overall Framework

Figure 1 demonstrates the overall workflow of the proposed framework. In this framework, the soccer videos are first segmented into distinct video shots and their low-level video/audio features are extracted. A multimedia data mining approach is utilized to pre-process the video shots to get an initial candidate pool for the potential important events. After that, a set of initial event labels will be given to some of the shots, where not all of these labels are correct. All of these data and information will be fed into this framework for event pattern searching and video retrieval purposes. The videos included in the candidate pool are modeled in the 1st level of MMM (Markov Model Mediator) models, whereas the videos are modeled in the 2nd level. After initializing the 1st level and 2nd level of MMM models, the users are allowed to issue the event or event pattern queries. Furthermore, the users can select their interested event patterns in the initial results and re-issue the query to refine the retrieval results and their rankings. This step is also recognized as online learning. These user selected shot sequences are stored as positive patterns for the future offline training.

After a certain amount of queries and feedbacks, the proposed framework is able to perform the offline training. The historical queries and user access records are utilized to update the affinity relationships of the videos/video shots as well as their initial state probabilities. Thereafter, both the semantic events and the high level user perceptions are employed to construct the video clusters, which are then modeled by a higher level (3rd level) of the MMM model. In the meanwhile, the 2nd level MMM model are divided into a set of sub-models based on the clustered video groups.

The clustered database and the updated HMMM mechanism are capable of providing appealing multimedia experience to the users because the modeled multimedia database system learns the user's preferences and interests interactively via reusing the historical queries .

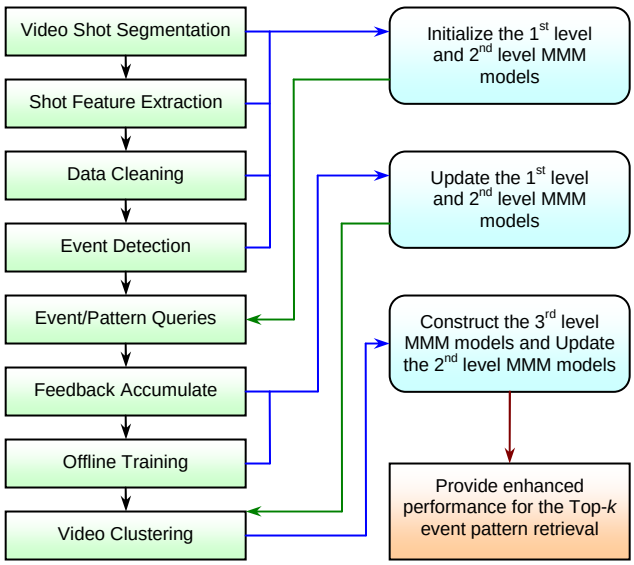


Figure 1. Overall workflow for the proposed approach

3. Video Database Clustering

3.1. Video Database Modeling

In our previous studies, we have successfully applied the MMM (Markov Model Mediator) model in image database clustering [4], and expanded MMM to HMMM [6] to model a video database which is formalized as an 8-tuple $\lambda = (d, S, F, A, B, \Pi, O, L)$. Let n denote the level number, where $1 \le n \le d$ and d is the number of levels in an HMMM. The n^{th} level of HMMM may contain one or more MMM models to represent the sets of distinct multimedia objects and their associate features. In [6], d is set as 2. S_1 in the lowest-level MMM represents the set of video shots, and the feature set F_1 consists of visual/audio features. While in the 2nd level MMM, S_2 describes the set of videos in the database, and F_2 contains the semantic events detected in the video collection. Each of the MMM models incorporates a set of matrices for affinity relationships (A_n), feature values (B_n), and initial state probability distributions (Π_n). In addition, O and L are designed for the relationship description between two adjacent levels. O ($O_{1,2}$) includes the weights of importance for the low-level features (F_1) when describing the high-level semantic events (F_2). L ($L_{1,2}$) describes the link conditions between the videos (S_2) and the video shots (S_1). HMMM model carries out a stochastic and

dynamic process in both search and similarity calculation, where it always tries to traverse the path with the largest possibility. Therefore, it can assist in retrieving more accurate patterns quickly with lower computational costs. The specific design of HMMM helps not only the general event queries, but also the retrieval of temporal based semantic event patterns.

Another significant advantage of HMMM is its interactive feedback and learning strategies, which can proficiently assure the continuous improvements of the overall performance. The users are capable of providing their own feedbacks such that the system can be trained either online or offline. The online learning mechanism creates an individual MMM instance by using the video shots that a user prefers. All of the users' feedbacks are efficiently accumulated and ready for the offline system training process. In this research, the large amount of user feedbacks will be reused in video database clustering to further improve the overall retrieval performance and reduce the searching space and time.

3.2. Conceptual Video Clustering

3.2.1. Similarity Measurement

In this proposed framework, a video is treated as an individual database in a distributed multimedia database system, where its video shots are the data instances in the database. Accordingly, a similarity measure between two videos is defined as a value indicating the likeness of these two videos with respect to their conceptual contents. It is calculated by evaluating their positive events and event patterns in the historical queries. That is, if two videos consist of the same event(s) and/or event pattern(s) and are accessed together frequently, it is considered that these two videos are closely related and their similarity score should be high.

Assume there are H user queries issued through the video retrieval framework, where the set of all the query patterns is denoted as QS. In order to refine their retrieved results in real-time, the users mark their preferred event patterns as "positive" before making the next query. By evaluating the issued query sets and their associated positive patterns, the similarity measure is defined as follows.

Let v_i and v_j be two videos, and $X=\{x_1, \ldots, x_m\}$ and $Y=\{y_1, \ldots, y_n\}$ be the sets of video shots belonging to v_i and v_j ($X \subseteq v_i$, $Y \subseteq v_j$), where m and n are the numbers of annotated video shots in v_i and v_j.

Denote a query with an observation sequence (semantic event pattern) with C semantic events as $Q^k=\{e_1^k, e_2^k, ..., e_C^k\}$, where $Q^k \in QS$. Let R^k be the set of G positive patterns that a user selected from the initial retrieval results for query Q^k. This can be represented by a matrix of size $G \times C$, $G \geq 1$, $C \geq 1$. As shown in Equation (1), each row of R^k represents an event shot sequence that the user marked as positive, and each column includes the candidate event shots which correspond to the requested event in the query pattern.

$$R^k = \begin{Bmatrix} \{s_1^1, s_2^1, ..., s_C^1\} \\ \{s_1^2, s_2^2, ..., s_C^2\} \\ ... \\ \{s_1^G, s_2^G, ..., s_C^G\} \end{Bmatrix}. \quad (1)$$

Based on the above assumptions, the video similarity function is defined as below.

***Definition 1*: $SV(v_i, v_j)$**, the similarity measure between two videos, is defined by evaluating the probabilities of finding the same event pattern Q^k from v_i and v_j in the same query for all the query patterns in QS.

$$SV(v_i, v_j) = \left(\sum_{Q^k \in QS} P(Q^k | v_i) P(Q^k | v_j) \right) \times FA(H), \quad (2)$$

where $1 \leq k \leq H$, and $FA(H)$ is an adjusting factor. $P(Q^k | v_i)$ and $P(Q^k | v_j)$ represent the occurrence probabilities of finding Q^k from v_i and v_j, where the occurrence probability can be obtained by summing the joint probabilities over all the possible states [3]. In order to calculate this value, we need to select all the subsets with C event shots from the positive pattern set R^k, which also belong to v_i or v_j. That is, $X'=\{x_1', x_2', ..., x_C'\}$ and $Y'=\{y_1', y_2', ..., y_C'\}$, where $X' \subseteq X$, $X' \in R^k$, $Y' \subseteq Y$, $Y' \in R^k$. If these patterns do not exist, then the probability value is set as 0 automatically.

$$P(Q^k | v_i) = \sum_{all\ X'} P(Q^k, X' | v_i) = \sum_{all\ X'} P(Q^k | X', v_i) P(X' | v_i). \quad (3)$$

Assume the statistical independence of the observations, and given the state sequence of $X'=\{x_1', x_2', ..., x_C'\}$, Equation (4) gives the probability of X' given v_i.

$$P(X' | v_i) = \prod_{t=1}^{C-1} P(x_t' | x_{t+1}') P(x_1') = \prod_{t=1}^{C-1} A_1(x_t', x_{t+1}') \pi_1(x_1'). \quad (4)$$

Here, $P(x_t' | x_{t+1}')$ represents the probability of retrieving a video shot x_{t+1}' given that the current video shot is x_t'. It corresponds to the $A_1(x_t', x_{t+1}')$ entry in the relationship matrix. $P(x_1')$ is the initial probability for video shot x_1', i.e., $\pi_1(x_1')$. Equation (5) gives the probability of an observation sequence (semantic event pattern) Q^k.

$$P(Q^k | X', v_i) = \prod_{t=1}^{C} P(e_t^k | x_t'), \quad (5)$$

where $P(e_t^k \mid x_t')$ indicates the probability of observing a semantic event e_t^k from a video shot x_t'. This value is computed using a similarity measure by considering low-level and mid-level features. However, in this approach, since the users already marked these video shots as the events they requested and preferred, the probabilities of observing the semantic events are simply set to 1.

3.2.2. Clustering Strategy

Considering a large scale video database, it is a significant issue to cluster similar videos together to speed up the similarity search. As we stated before, a two-level HMMM has been constructed to model video and video shots. Furthermore, a video database clustering strategy which is traversal-based and greedy is proposed.

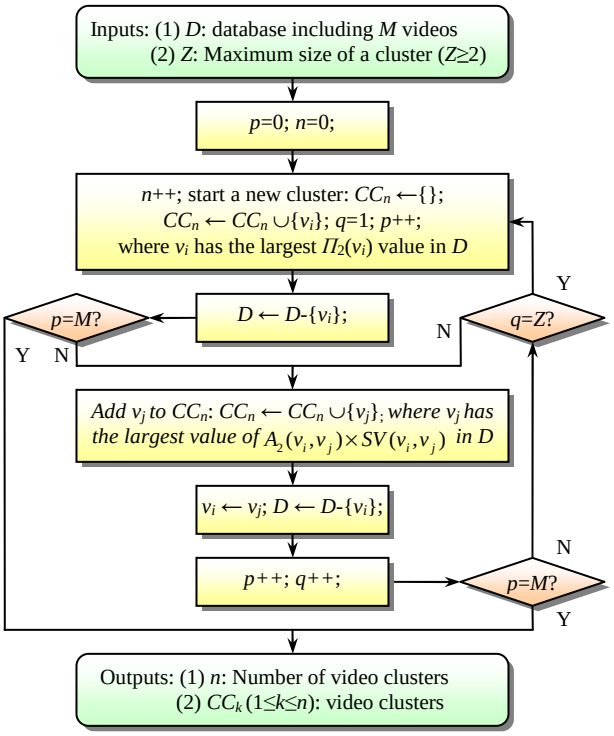


Figure 2. The proposed conceptual video database clustering procedure

As illustrated in Figure 2, the proposed video database clustering technique contains the following steps. Given the video database D with M videos and the maximum size of the video database cluster as Z ($Z \geq 2$), the mechanism:

a) Initialize the parameters as $p=0$; $n=0$, where p denotes the number of videos being clustered, and n represents the cluster number.

b) Set $n=n+1$. Search the current video database D for the video v_i with the largest stationary probability $\Pi_2(v_i)$, and then starts a new cluster CC_n with this video ($CC_n = \{\}$; $CC_n \leftarrow CC_n \cup \{v_i\}$). Initialize the parameter as $q=1$, where q represents the number of videos in the current cluster.

c) Remove v_i from database D ($D \leftarrow D-\{v_i\}$). Check if $p=M$. If yes, output the clusters. If no, go to step d).

d) Search for v_j, which has the largest $A_2(v_i,v_j) \times SV(v_i,v_j)$ in D. Add v_j to the current cluster CC_n ($CC_n \leftarrow CC_n \cup \{v_i\}$).

e) $v_i \leftarrow v_j$, where v_i represents the most recent clustered video. Every time when a video is assigned to a cluster, it is automatically removed from D ($D \leftarrow D - \{v_i\}$).

f) p++ and q++. Check if $p=M$. If yes, output the clustering results. If no, check if $q=Z$. If yes, goes to step b) to start a new cluster. If no, goes to step d) to add another video in the current cluster.

g) If there is no un-clustered video left in the current database, output the current clusters.

3.3. Interactive Retrieval upon Video Clusters

3.3.1. Inter-Cluster Relationships

In this research, the HMMM model is extended by the 3rd level MMM to improve the overall retrieval performance. In the 3rd level MMM ($d = 3$), the states (S_3) denotes the video clusters. Matrix A_3 describes the relationships between each pair of clusters.

***Definition 2*:** Assume CC_m and CC_n are two video clusters in the video database D. Their relationship is denoted as an entry in the affinity matrix A_3, which can be computed by Equations (6) and (7). Here, SC is the function that calculates the similarity score between two video clusters.

$$SC(CC_m, CC_n) = \sum_{v_i \in CC_m} (\Pi_2(v_i) \times \max_{v_j \in CC_n} (A_2(v_i,v_j) \times SV(v_i,v_j))) / M \,,$$
$$\text{where } CC_m \in D, CC_n \in D \,. \quad (6)$$

$$A_3(CC_m, CC_n) = \frac{SC(CC_m, CC_n)}{\sum_{CC_j \in D} SC(CC_m, CC_j)} \,. \quad (7)$$

Table 1. 3-Level HMMM Model

	1st Level MMM	2nd Level MMM	3rd Level MMM
S	State set of video shots	State set of Videos	State set of video clusters
F	Low level visual/audio features	Semantic events (concepts)	-
A	Temporal based state transition probability between video shots	Affinity relationship between videos	Affinity relationship between video clusters
B	Formalized feature values	Annotated event numbers	-
Π	Initial state probability distribution for video shots	Initial state probability distribution for videos	Initial state probability distribution for video clusters

The matrix Π_3 can be constructed to represent the initial state probability of the clusters. The calculation of Π_3 is similar to the ones for Π_1 and Π_2. In addition, matrix $L_{2,3}$ can also be constructed to illustrate the link conditions between the 2nd level MMMs and the 3rd level MMM. As demonstrated in Table 1, the MMM models in different levels of the 3-level HMMM describe distinct objects and represent different meanings.

3.3.2. Retrieval through Clustered Video Database

Given an example shot sequence $Q'=\{s_1,s_2,...,s_C\}$ which represents the event pattern as $\{e_1,e_2,...,e_C\}$ such that s_i describes e_i ($1 \le i \le C$), and they follow the temporal sequence as $T_{s_1} \le T_{s_2} \le ... \le T_{s_C}$. Assume that a user wants to find top-k related shot sequences which follow the similar patterns. In our proposed retrieval algorithm, a recursive process is conducted to traverse the HMMM database model and find the top k candidate results. As shown in Figure 3, a lattice based structure for the overall video database can be constructed. Assume the transitions are sorted based on their edge weights [6], and the retrieval algorithm will traverse the edge with a higher weight each time. For example, in Figure 4, we assume that the edge weights satisfy $w(s_1, s_2) \ge w(s_1, s_4) \ge w(s_1, s_7)$. The retrieval algorithm can be described as below.

1. Search for the first candidate cluster, first candidate video and first candidate video shot by checking matrices Π_3, Π_2, B_2, Π_1 and B_1.
2. If the pattern is not complete, continue search for the next event (video shot) via computing the edge weights by checking A_1.
3. If the candidate pattern has been completed, the system goes back state by state and checks for other possible paths. The system also checks if there are already k candidate patterns being retrieved. If yes, the system stops searching and goes to Step 6.
4. If there is no more possibilities in the current video, then mark this video with a "searched" flag and check A_2 and B_2 to find next candidate video.
5. If all the videos are "searched" in the current cluster, then mark the current cluster as "searched" cluster and check A_3 to find the next candidate video cluster.
6. Once k patterns are retrieved, or there are no more possibilities in the database, the system ranks the candidate patterns via calculating the similarity scores [6] and outputs the candidate patterns.

As shown in the Figure 4, the yellow cells include the paths the algorithm traversed. Furthermore, we designed a function to fill in the missed cells by copying the correspondent shots in the previous candidate patterns. Finally, 6 complete candidate patterns are generated. Once k candidate patterns are generated, the system does not need to traverse any other clusters or videos. Therefore, it significantly reduces the searching spaces and accelerates the searching speed.

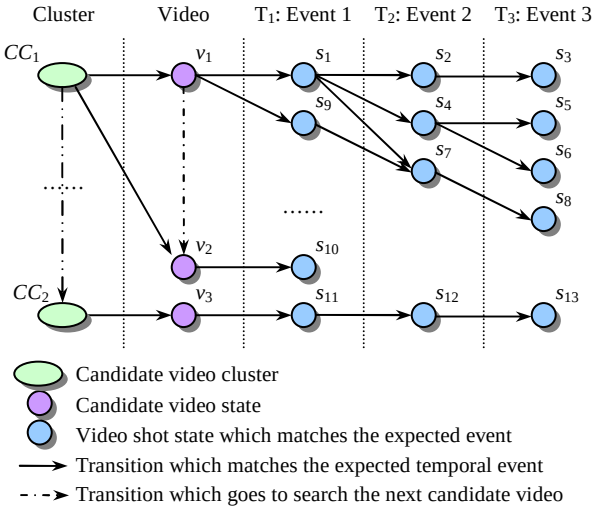


Figure 3. Lattice structure of the clustered video database

R	Cluster	Video	Event 1	Event 2	Event 3
1	CC_1	v_1	s_1	s_2	s_3
2	CC_1	v_1	s_1	s_4	s_5
3	CC_1	v_1	s_1	s_4	s_6
4	CC_1	v_1	s_1	s_7	s_8
5	CC_1	v_1	s_9	s_7	s_8
6	CC_2	v_3	s_{11}	s_{12}	s_{13}

Figure 4. Result patterns and the traverse path

4. EXPERIMENTAL RESULTS

We have built up a soccer video database with totally 45 videos, which contains 8977 video shots. A retrieval system has also been implemented for the system training and experimental tests. Totally 150 sets of historical queries were issued and user feedbacks were returned with their preferred patterns, which cover all of the 45 videos and 259 distinct video shots. In the clustering process, we define the cluster size as 10 and the expected result pattern number as k=60. As shown in Figure 5, we use letters "G", "F", and "C" to represent "Goal", "Free kick", "Corner kick" events, respectively. Therefore, the x-axis represents different query patterns, e.g., "G" means a query to search for "Goal" events; "FG" means a query to search for the event pattern where a "Free kick" followed by a "Goal"; and "CGF" means a query pattern of a "Corner kick" event, followed by a "Goal" and then a "Free kick", etc. For each query pattern, we issued 10 queries to compute the average execution time in milliseconds. As illustrated in Figure 5, the query patterns with fewer event numbers will be executed in less time as expected. In addition, the execution time of the system with clusters is less than that of the system without clusters, indicating that our proposed approach effectively

groups relevant videos in the video clusters so that only the relevant clusters and their member videos will need to be searched. Therefore, the searching space is dramatically decreased, and the execution of the queries becomes faster.

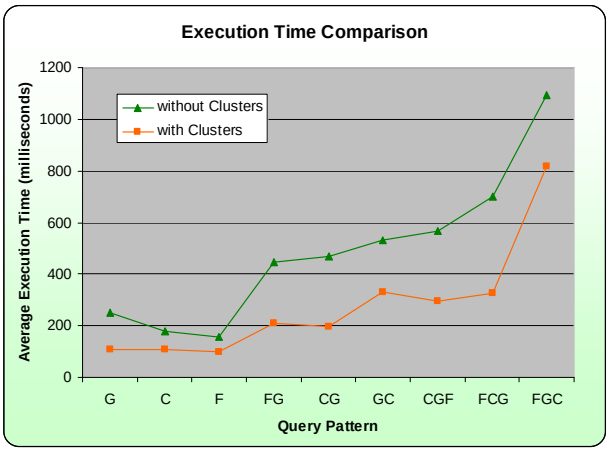


Figure 5. Comparison of the average execution time

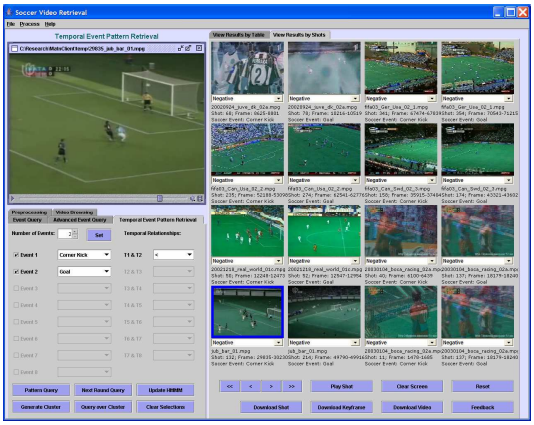
For the query pattern ("Corner kick" followed by a "Goal"), Figure 6(a) demonstrates the first screen of retrieval results over the non-clustered soccer video database; while Figure 6(b) shows the query results over the clustered database. It can be clearly seen that the query results in the same cluster represent the similar visual clues, which are mined from the historical queries and feedbacks, and accordingly represent user preferences.

5. CONCLUSIONS

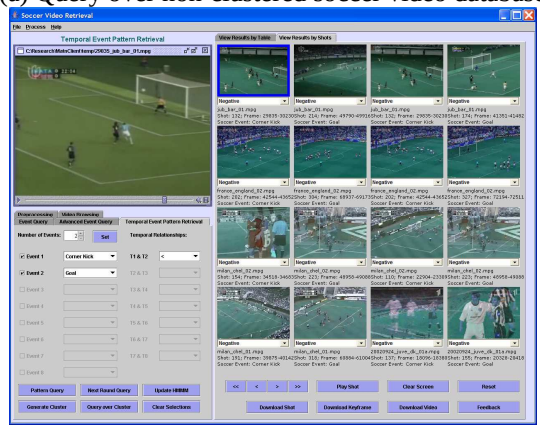
In this paper, an interactive video retrieval system is proposed which incorporates the conceptual video clustering strategy and the HMMM hierarchical learning mechanism. This proposed framework is able to reuse the cumulated user feedbacks to cluster the videos, such that the overall system not only learns the user perceptions, but also gets a good database structure via adopting the clustering technique. The HMMM-based database model is constructed to support the conceptual video database clustering. In the meanwhile, the clustering technique helps to further improve the database structure via adding a new level to model the video clusters. The experiments show that our proposed approach helps accelerate the retrieval speed with providing decent retrieval results.

6. ACKOWLEDGEMENTS

For Shu-Ching Chen, this research was supported in part by NSF EIA-0220562 and HRD-0317692. For Mei-Ling Shyu, this research was supported in part by NSF ITR (Medium) IIS-0325260. For Stuart Rubin, this research was supported in part by an ONR ILIR grant.



(a) Query over non-clustered soccer video database



(b) Query over clustered soccer video database

Figure 6. Soccer video retrieval system interfaces

7. REFERENCES

[1] C.-W. Ngo, T.-C. Pong and H.-J. Zhang, "On Clustering and Retrieval of Video Shots," In *Proc. of the 9th ACM International Conference on Multimedia*, Ottawa, Canada, 2001, pp. 51-60.

[2] J.-M. Odobez, D. Gatica-Perez, and M. Guillemot, "Video Shot Clustering using Spectral Methods," In *Proc. of 3rd International Workshop on Content-Based Multimedia Indexing (CBMI)*, Rennes, France, 2003, pp. 94-102.

[3] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*, Prentice Hall, 1993, ISBN: 0130151572.

[4] M.-L. Shyu, S.-C. Chen, M. Chen, and C. Zhang, "Affinity Relation Discovery in Image Database Clustering and Content-based Retrieval," In *Proc. of ACM Multimedia 2004 Conference*, New York, USA, pp. 372-375.

[5] L. Xie, S.-F. Chang, A. Divakaran, and H. Sun, "Unsupervised Discovery of Multilevel Statistical Video Structures Using Hierarchical Hidden Markov Models," In *Proc. of IEEE International Conference on Multimedia and Expo (ICME)*, vol. 3, pp. 29-32, July 2003.

[6] N. Zhao, S.-C. Chen and M.-L. Shyu, "Video Database Modeling and Temporal Pattern Retrieval Using Hierarchical Markov Model Mediator," In *Proc. of the First IEEE International Workshop on Multimedia Databases and Data Management (MDDM), in conjunction with IEEE International Conference on Data Engineering (ICDE)*, Atlanta, USA, 2006.