Video Database Modeling and Temporal Pattern Retrieval using Hierarchical Markov Model Mediator

Na Zhao[1], Shu-Ching Chen[1], Mei-Ling Shyu[2]
[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
[1]*{nzhao002, chens}@cs.fiu.edu,* [2]*shyu@miami.edu*

Abstract

The dream of pervasive multimedia retrieval and reuse will not be realized without incorporating semantics in the multimedia database. As video data is penetrating many information systems, the need for database support for video data evolves. Hence, we propose an innovative database modeling mechanism called Hierarchical Markov Model Mediator (HMMM) which integrates low-level features, semantic concepts, and high-level user perceptions for modeling and indexing multiple-level video objects to facilitate temporal pattern retrieval. Different from the existing database modeling methods, our approach carries a stochastic and dynamic process in both search and similarity calculation. In the retrieval of semantic event patterns, HMMM always tries to traverse the right path and therefore it can assist in retrieving more accurate patterns quickly with lower computational costs. Moreover, HMMM supports feedbacks and learning strategies, which can proficiently assure the continuous improvements of the overall performance.

1. Introduction

To make multimedia information pervasively accessible and reusable, the "semantic gap" issue needs to be addressed. This denotes the gulf between the rich meaning and interpretation that the users anticipate the database systems to associate their queries for searching and browsing multimedia data. The other crucial problem is how to efficiently derive and facilitate semantic annotations which require knowledge and techniques from assorted disciplines and domains, even though many of them are outside of the traditional computer science fields. Furthermore, another emerging challenge is how to proficiently model, accumulate, and query multimedia data, along with the specific semantic events or event patterns.

To efficiently manage the large video archive, a promising solution is to architect the high-level semantic descriptions for multimedia content processing, management, and retrieval. In this paper, we present the Hierarchical Markov Model Mediator (HMMM) mechanism to efficiently store, organize, and manage low-level features, multimedia objects, and semantic events along with high-level user perceptions, such as user preferences, in the multimedia database management system (MMDBMS). In order to archive all valuable data, HMMM also adopts multi-disciplinary techniques, such as content-based image analysis, audio feature extraction, video shot detection and segmentation algorithms, machine learning methodologies, and relevance feedback techniques.

By employing the proposed HMMM mechanism, the high-dimensional multimedia databases can be efficiently organized, indexed and managed. Moreover, the temporal relationships between the video shots are naturally integrated in HMMM such that the proposed mechanism can offer the capabilities to execute not only the traditional event queries but also the complicated temporal pattern retrieval towards the large scale video database in a quick and accurate manner. In addition, the feedback and offline learning strategies are incorporated in the HMMM mechanism such that high-level user perceptions and preferences as well as the low-level visual/audio features can be considered. In general, HMMM helps to bridge the semantic gap between concept-based and the content-based retrieval approaches to the underlying multimedia database modeling.

The remainder of this paper is organized as follows. Section 2 reviews the current research projects and a variety of existing systems on multimedia database modeling, indexing, and retrieval. Section 3 introduces the temporal pattern query along with the architecture of the proposed framework. Section 4 provides the formalization and discussion of HMMM. Section 5 describes the algorithm for the temporal pattern retrieval processing upon the HMMM-based database structures, and presents our experimental example and result. Finally, the conclusions are summarized in Section 6.

2. Related Work

Traditional video databases basically store the textual descriptors along with the source videos such that the textual based video queries can be easily performed. However, it has limited capabilities in browsing and querying video data. Some of the later researches allow browsing and querying based on the video structure data. As an example, an interactive content-based video browser is presented in [11], which supports a hierarchical navigation of video over the Internet through multiple levels of key frames.

With the resurgence of research in multimedia database and retrieval systems, many approaches have been developed to support retrieval based on low-level or mid-level visual/audio features by using Query by Examples (QBE) [2][4][12]. In [2], a video-enhanced database system called VDBMS was proposed to support feature-based medical video data retrieval. [4] describes a system which applies image retrieval techniques to query videos by setting up the links between videos and images. In addition, IBM's video retrieval system MARVEL [12] supports QBE in both the low-level feature space and the high-level model-vector space. However, QBE approaches have their own limitations because the users may not have the image/video example at hand when issuing the queries. In addition, QBE would not perform well if the query example is not taken with an appropriate angle or scale.

Recently, there is a rapid proliferation of visual processing and analysis techniques, where the salient objects and their motions can be identified and utilized in video retrieval. For example, VideoQ was a web based video search system to search the video clips containing the object motion similar to the animated sketches provided by the user [3]. From the general users' points of view, it is expected to find videos on the basis of the semantic content of the video. Therefore, the most recent researches mainly focus on semantic events retrieval. Some of the existing video query approaches utilize event annotations, which are generally described as time-dependent information or values that are synchronous with the source data such as SMOOTH [14], GOALGLE [16], and IBM TRL's MEPG-7 authoring system [13]. These approaches either support semantic queries and some basic temporal queries, or deploy event-based indexing via the inclusion of the event name, start time, and end time.

The existing event-based and object-based video retrieval applications may encounter the problem that event detection and object segmentation require manual annotations of video events, salient objects, and their boundaries. [1] describes a system for content-based and model-based detection and retrieval of events and other concepts, where the semantic concept models are built by training the classifiers using labeled training videos. VideoQA was proposed to query a news video archive, where the speech recognition techniques are utilized for the semantic meaning retrieval [17]. Ideally, the semantic contents of the video data can be mined automatically by utilizing various machine interpretation techniques and therefore the videos can be automatically annotated. However, based on the state of the art in computer vision, this kind of complicated data abstractions may not be feasible in practice. Instead, the computer may perform automatic annotation with limited semantic interpretation.

Due to the rapid propagation of multimedia applications that require video data management, it becomes more desirable to provide proper video database modeling and indexing techniques capable of representing the rich semantics in video data. The approach proposed in [9] employs high-dimensional points to represent the pixel regions. Another multilevel video indexing approach called ClassView was proposed in [10], which a set of hash tables are included in the database indexing structure for different visual concept levels, and a root hash table is utilized to integrate the information about all semantic clusters.

Our research group has proposed semantic event mining methodologies to identify the "goal" and "corner kick" events from soccer videos [6][7], and a temporal query model related graphical query language for soccer event queries with the support of temporal relationships [8]. In this paper, we mainly aim at developing a comprehensive database modeling mechanism for video event pattern retrieval by considering temporal relationships with the help of user feedbacks and machine learning technologies.

3. Overall Framework

In general, multimedia data and metadata can be categorized into three groups: entities, attributes, and values, where the description of an entity is composed of the combinations of attributes and their corresponding values. One of the significant characteristics of video data is that video entities may pose various temporal or spatial relationships. Accordingly, the users are normally interested in specific semantic concepts and the associated temporal-based event patterns when querying a large scale video archive. However, some of the current computer vision and video/audio analysis techniques only offer limited query processing techniques on textual annotations or primitive low-level or mid-level features. Although a variety of researches start to consider the retrieval on semantic events and the salient objects, there lacks a comprehensive database modeling technique to support the access and query on the temporal based event patterns.

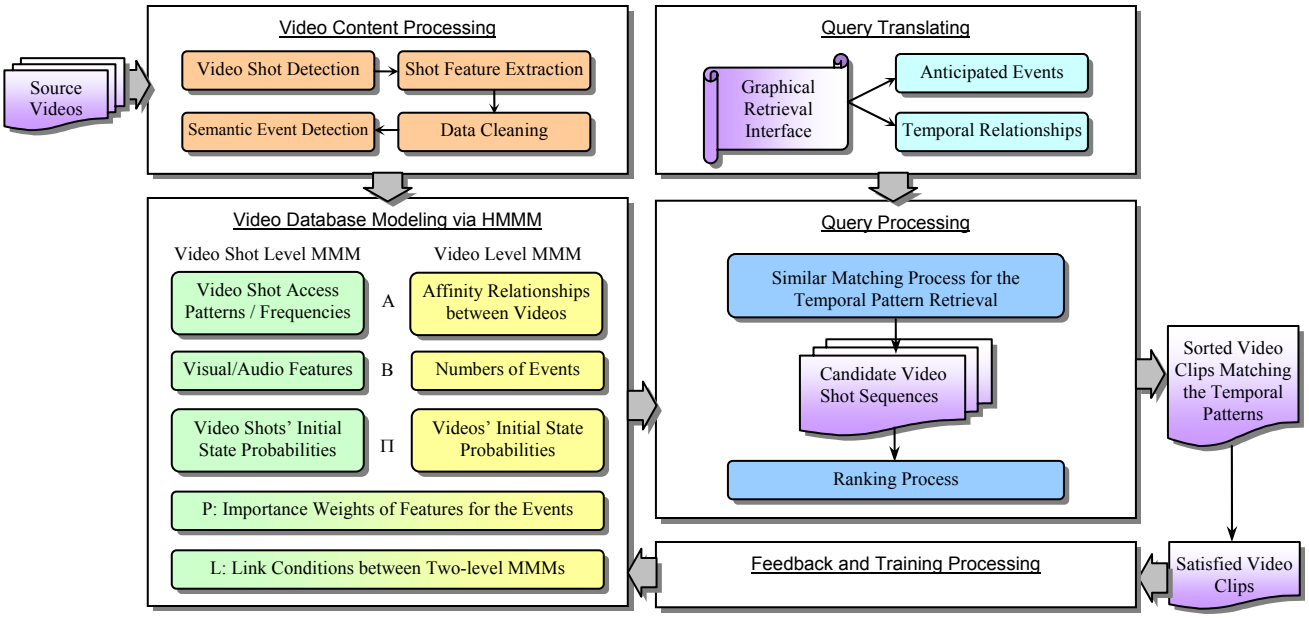

Figure 1. Overall framework of video database modeling and temporal pattern retrieval utilizing HMMM

In this study, a temporal event pattern is defined as a sequence of semantic events which follow some specific temporal relations. Here, a semantic event annotation is used to mark real-world situations of the video shot, also referred to as events. For instance, in soccer video, the events such as "goal", "corner kick", "free kick", "foul", "goal kick", "yellow card", and "red card" are considered. An example temporal pattern query can be expressed as follows: "A user wants to search for a specific soccer video segment with the following temporal patterns. At first, a goal is resulted from a free kick. After that, a corner kick occurs at some point in time, followed by a player change, and finally another goal shot follows the player change." In our proposed approach, the Markov Model Mediator (MMM) mechanism is extended to Hierarchical MMM such that the multiple-level video entities and their associated temporal or affinity relationships can be efficiently modeled to answer this type of temporal pattern queries.

As illustrated in Figure 1, our proposed framework consists of five major components. The first step is to process the video data by utilizing the multi-disciplinary techniques to detect the video shot boundaries and extract the shot features. After the data cleaning procedure, data mining techniques are deployed to detect the semantic events. The detailed algorithms for soccer event detection can be found in [7]. Second, as shown in the lower-left box, HMMM is deployed to model the extracted features, detected events, segmented video shots and original source data, etc. Once a temporal pattern query is issued via the graphical retrieval interface, the third component (query translator) analyzes the user requirements and encodes the query to a set of expected events and their associated temporal patterns. These requirements are sent to the query processing component as inputs. The similarity matching process is then executed to achieve the candidate video shot sequences and finally they are sorted according to the similarity scores. Moreover, the HMMM mechanism can be trained via considering user feedbacks for continuous system learning.

4. The Hierarchical Stochastic Model

4.1. Definition of HMMM

Markov Model Mediator (MMM) is a well-established mathematical construct capable of modeling complicated multimedia databases and can efficiently collect and report information periodically. MMM has been successfully applied in several applications such as content-based image retrieval [15]. In this study, MMM is extended to multiple level descriptions and utilized for video database modeling, storage and retrieval purposes. The formal description of an HMMM is defined as below.

Definition 1: An HMMM is represented by an 8-tuple $\lambda = (d, S, F, A, B, \Pi, P, L)$, where

- d is the number of levels in an HMMM. The hierarchy index of the lowest level with the smallest multimedia object is 1 and the index for the highest level state is d.
- S (S_n) represents distinct multimedia objects in different levels. For example, in the proposed

framework, S_1 is the set of states for the shot-level video clips; while S_2 is a set of video states. A state of an HMMM is denoted by $S_n(i)$, where i is the state index and n is the hierarchy index. Here, we also denote the number of the sub-states of an internal state $S_n(i)$ as $|S_n(i)|$.

- F (F_n) is defined as the sets of distinct features of the specific multimedia objects. The entry $F_n(i)$ denotes the i^{th} feature in the n^{th}-level MMM. In this paper, the lower-level MMM model for video shots incorporates the feature sets as low-level or mid-level visual/audio features, while the higher level MMM model for the videos integrates the features as the semantic events.
- A ($A_n \rightarrow S_n \times S_n$) indicates the state transition probability distributions, where A_n denotes the n^{th} level matrix for state transition probability. The entry $A_n(i, j)$ actually indicates the relationship between the i^{th} and j^{th} multimedia objects in the n^{th} level MMM. The higher this entry is, the tighter the relationship exists between these two objects. In the retrieval process, the possibility is also higher to traverse from $S_n(i)$ to $S_n(j)$.
- B ($B_n \rightarrow S_n \times F_n$) denotes the feature matrices of different level MMMs by incorporating the associated feature values for each multimedia object. In this paper, B_1 includes the visual/audio feature values for each of the video shots; while B_2 consists of the numbers of semantic events associated with each of the videos.
- Π (Π_n) incorporates the information for the initial state probability distributions, where $\Pi_n(i)$ represents the possibility that the i^{th} multimedia object of the n^{th} level MMM is chosen as the initial state in the retrieval process.
- P ($P_{n,n+1} \rightarrow F_n \times F_{n+1}$) represents the weights of importance of the lower-level features (F_n) when describing the higher level feature concepts (F_{n+1}).
- L ($L_{n,n+1} \rightarrow S_n \times S_{n+1}$) denotes the link conditions between the higher level states (S_{n+1}) and the lower level states (S_n).

4.2. Construction of two-level HMMM

In this paper, a two-level HMMM model is deployed to model the source video and their associated video shots. More specifically, the fundamental level of the MMM model consists of a series of consecutive video shots. It needs to be noted that the events are referred to as shot-level video clips in this paper. It is merely a choice of representation rather than a statement about the actual duration of a specific event. Thus, one local MMM model is architected for each video in the database; while an integrated MMM model is constructed to model all the source videos and incorporate all the lower level MMM models.

4.2.1. Video shot level MMM

As we stated above, the matrices for affinity relationship, feature, and initial state probability distributions at different levels may hold slightly dissimilar meanings although the general depiction is the same. In the most fundamental level, the states (S_1) represent the video shots, which are the elementary units in the video database and describe the continuous action between the start and end of a camera operation. The feature set (F_1) for the video shot level MMM consists of low-level or mid-level visual/audio features.

4.2.1.1. A_1: temporal-based relative affinity matrix

A_1 represents the temporal-based affinity relationship between the video shots in the video shot-level MMM. Let $S_1(i)$ and $S_1(j)$ (where $0 < i < j$) represent two specific semantic events, and if they are frequently accessed together in one temporal event pattern, they are said to have a higher affinity relationship. Hence, their temporal based affinity relationship value from $S_1(i)$ to $S_1(j)$ will be larger.

1) Initialization of A_1

Let N be the number of video shots $\{s_1, s_2, \ldots, s_N\}$ annotated in the initial video database. Assume these video shots follow the normal temporal sequence, i.e., $T_{s_1} < T_{s_2} < \ldots < T_{s_N}$, where T_{s_i} is the occurrence time of s_i. The number of event annotations in s_i is represented as $NE(s_i)$. Accordingly, A_1 can be initialized as follows. Basically, $A_1(i,j)$ is set as 0 when $i > j$. When $1 \le i < N$ and $i < j \le N$, $A_1(i,j) = NE(s_j) / ((\sum_{k=i}^{N} NE(s_k)) - 1)$. If $i = j$ and $i \ne N$, $A_1(i,j) = (NE(s_j) - 1) / ((\sum_{k=i}^{N} NE(s_k)) - 1)$. When $i = j = N$, $A_1(i,j)$ value will be set as 1, which denotes that s_i is the last event of this video.

Given an example, if in a soccer video V, there are originally three shots to be annotated. The first shot (s_1) is denoted as "Free Kick"; the second shot (s_2) is annotated as two events "Free Kick" and "Goal"; while the last shot (s_3) is a "Corner Kick" shot. Based on the above descriptions, $NE(s_1)$=1, $NE(s_3)$=1, and $NE(s_2)$=2.

Therefore, we can calculate the values in matrix A as: $A_1(1,2) = 2/((1+2+1)-1) = 2/3$, $A_1(1,3) = 1/3$, $A_1(2,2) = 1/2$, $A_1(2,3) = 1/2$, $A_1(3,3) = 1$.

2) Update of A_1

By adopting HMMM, the users are allowed to provide their feedbacks to the system. Those video shot sequences that are similar to the anticipated temporal event pattern will be marked as "Positive" patterns. This information is used to capture the user preferences to refine the system retrieval capability for the future update. A matrix AF_1 is defined to capture the temporal-based affinity relationships among all the annotated video shots using user access patterns and access frequencies. For the k^{th} pattern R_k, $access_1(k)$ represents its access frequencies, and $use_1(i, k)$ equals 1 if s_i (the i^{th} video shot) was accessed in pattern R_k. Moreover, both s_m and s_n should belong to the "Positive" temporal pattern R_k and s_m should occur before s_n or they should occur at the same time. Let q be the number of positive patterns on the shot level.

$$aff_1(m,n) = A_1(m,n) \times \sum_{k=1}^{q} use_1(m,k) \times use_1(n,k) \times access_1(k), \quad iff\ s_m \in R_k,\ s_n \in R_k,\ T_{s_m} \le T_{s_n}. \tag{1}$$

Each entry of $aff_1(m,n)$ in AF_1 indicates the frequency of s_m and s_n being accessed together in the first level MMM, and consequently the probability of these two video shots being accessed together in the temporal patterns. A_1 can then be updated via normalizing AF_1 per row and thus MMM represents the relative affinity relationships among all the video clips in the database. Let $A_1(m,n)$ be the element in the (m, n) entry in the first level MMM model and N be the video shot number of this model, then

$$A_1(m,n) = aff_1(m,n) / \sum_{j=1}^{N} aff_1(m,j), \quad where\ 1 \le m \le N\ and\ 1 \le n \le N. \tag{2}$$

For the sake of efficiency, the training system can only record all the user access patterns and access frequencies during a training period, instead of updating A_1 matrix on-line every time. Once the number of newly achieved feedbacks reaches a certain threshold, the update of A_1 matrix can be triggered automatically. All the computations should be done offline.

4.2.1.2. B_1: visual/audio feature matrix

We consider both the visual and audio features in the feature matrix B_1 for the video shot level MMM construction. As shown in Table 1, there are a total of 5 visual and 15 audio features [6].

TABLE I. FEATURE LIST FOR THE VIDEO SHOTS

Category	Feature Name	Feature Description
Visual Features	*grass_ratio*	Average percent of grass areas in a shot
	pixel_change_per cent	Average percent of the changed pixels between frames within a shot
	histo_change	Mean value of the histogram difference between frames within a shot
	background_var	Mean value of the variance of background pixels
	background_mea n	Mean value of the background pixels
Audio Features	*volume_mean*	Mean value of the volume
	volume_std	Standard deviation of the volume, normalized by the maximum volume
	volume_stdd	Standard deviation of the difference of the volume
	volume_range	Dynamic range of the volume, defined as (max(v)-min(v))/max(v)
	energy_mean	Mean RMS energy
	sub1_mean	Average RMS energy of the first sub-band
	sub3_mean	Average RMS energy of the third sub-band
	energy_lowrate	Percentage of samples with RMS power less than 0.5 times the mean RMS power
	sub1_lowrate	Percentage of samples with RMS power less than 0.5 times the mean RMS power of the first sub-band
	sub3_lowrate	Percentage of samples with RMS power less than 0.5 times the mean RMS power of the third sub-band
	sub1_std	Standard deviation of the mean RMS power of the first sub-band energy
	sf_mean	Mean value of the Spectrum Flux
	sf_std	Standard deviation of the Spectrum Flux, normalized by the maximum Spectrum Flux
	sf_stdd	Standard deviation of the difference of the Spectrum Flux, which is normalized too
	sf_range	Dynamic range of the Spectrum Flux.

1) Normalization of B_1

The initial values of the features need to be normalized to achieve more accurate similarity measures. To capture the original value of a feature in a video shot, we define a temporal matrix BB_1 whose rows represent the distinct video shots while the columns denote all the distinct features. The entry of $BB_1(i,j)$ denotes the original value of the j^{th} feature in the i^{th} video shot, where $1 \le j \le K$, K is number of features, and $1 \le i \le N$, N is the number of video shots. Our target is to normalize all of the features to fall between [0, 1]:

$$B_1(i,j) = \frac{BB_1(i,j) - \min_{k=1}^{N}(BB_1(k,j))}{\max_{k=1}^{N}(BB_1(k,j)) - \min_{k=1}^{N}(BB_1(k,j))}, \quad \text{where } 1 \le i \le N,\ 1 \le j \le K. \tag{3}$$

4.2.1.3. Π_1: initial state probability matrix for shots

The preference of the initial states for queries can be achieved from the training data set. For any video shot state $s_m \in S_1$, the initial state probability is defined as the fraction of the number of occurrences of video shot s_m as the initial state can traverse with respect to the total

number of occurrences for all the initially traversed video shot states in the video database (D) from the training data set.

$$\Pi_1 = \{\pi_m\} = \frac{\sum_{k=1}^{N} use_1(m,k)}{\sum_{l \in S_1} \sum_{k=1}^{N} use_1(l,k)}. \quad (4)$$

4.2.2. Video-level MMM

The purpose of constructing video-level MMM is to cluster the videos describing similar events. A large video archive may contain various kinds of videos, such as news videos, movies, advertisement videos, and sports videos. The integrated MMM is constructed such that the system is able to learn the semantic concepts and then cluster the videos into different categories.

4.2.2.1. A_2: relative affinitive matrix for videos

Based on the information contained in the training data set, the affinity relationships among the video sets in the database can be captured, i.e., the higher the frequency of two videos being accessed together, the closer they are related to each other. The relative affinity matrix A_2 is constructed in two steps as follows:

First, a matrix AF_2 is defined to capture the affinity measures among all the videos using user access patterns and access frequencies. After that, each entry $aff_2(m,n)$ in AF_2 indicates the frequency of the two videos v_m and v_n being accessed together in the 2^{nd} level MMM, and consequently how closely these two videos are related to each other. Let q' be the number of queries on the video level.

$$aff_2(m,n) = \sum_{k=1}^{q'} use_2(m,k) \times use_2(n,k) \times access_2(k). \quad (5)$$

The matrix A_2 can then be obtained via normalizing AF_2 per row and thus represents the relative affinity relationships among all the M videos in the database (D).

$$A_2(m,n) = aff_2(m,n) / \sum_{j=1}^{M} aff_2(m,j), \quad where\ 1 \le m \le M\ and\ 1 \le n \le M. \quad (6)$$

Please note that A_1 and A_2 are different since A_1 considers the temporal relationships as well, while A_2 does not.

4.2.2.2. B_2: event number matrix for videos

Matrix B_2 includes the event numbers of each video, where each row represents a video and each column denotes one semantic event. Assume there are totally M videos in the database, where the video v_i ($1 \le i \le M$) contains the set of C events denoted as $\{e_1, e_2, \ldots, e_C\}$, and $B_2(i,j)$ means the number of the j^{th} event (e_j) in v_i. B_2 does not need to be normalized and the integer values are kept.

4.2.2.3. Π_2 : initial state probability matrix for videos

In the video-level, the access patterns and access frequencies for videos in use_2 (instead use_1) are used to construct the matrix Π_2.

4.2.3. Connections between local MMM and integrated MMM

4.2.3.1. $P_{1,2}$: weight importance matrix

Since there are only two levels of the MMM models in the proposed approach, only one weight importance matrix ($P_{1,2}$) is required to denote the relationship between the features for video shots and the specific semantic events. This matrix is utilized to adjust the characteristic influences by learning the features of the annotated events. In $P_{1,2}$, each row represents an event concept, while each column represents one of the visual or audio features. The value in $P_{1,2}$ means the weight of the importance of the corresponding feature for the specific event concept.

1) Initialization of $P_{1,2}$

Let each multimedia object have K features $\{f_1, f_2, \ldots, f_K\}$ and C events $\{e_1, e_2, \ldots, e_C\}$. We define the initial value for each feature in an event concept to be $1/K$, which means they carry the same weight importance.

$$P_{1,2}(i,j) = \frac{1}{K}, \text{ where } 1 \le i \le C, 1 \le j \le K. \quad (7)$$

2) Update of $P_{1,2}$

Once a group of N video shots $\{s_1, s_2, \ldots, s_N\}$ consisting of the same event concept e_i ($1 \le i \le C$) are known, the standard deviations of the K features for all the N video shots can be calculated as $\{Std_{i,1}, Std_{i,2}, \ldots, Std_{i,K}\}$, where $Std_{i,j}$ represents the standard deviation of the i^{th} event and j^{th} feature ($1 \le i \le C$, $1 \le j \le K$). Equations (8)-(10) can be employed to compute $P_{1,2}$. As can be seen from Equation (10), the larger the $P_{1,2}$ value is, the more important this feature is when calculating the similarity score with the specified event.

$$P'(i,j) = \frac{1}{Std_{i,j}}, \text{ where } 1 \le i \le C, 1 \le j \le K \quad (8)$$

$$P_{1,2}(i,j) = \frac{P'(i,j)}{\sum_{j=1}^{K} P'(i,j)}; \text{ and} \quad (9)$$

$$P_{1,2}(i,j) = \left(\frac{1}{Std_{i,j}}\right) / \left(\sum_{j=1}^{K} \frac{1}{Std_{i,j}}\right). \quad (10)$$

4.2.3.2. B_1': mean value of the features per event

In matrix B_1', the row represents an event (concept), and the column denotes the visual and audio features. Assume that for the event e_i ($1 \le i \le C$), a set of N video shots $\{s_1, s_2, \ldots, s_N\}$ are identified e_i. The mean value of the features f_j ($1 \le j \le K$) for e_i can be calculated as follows.

$$B_1'(i,j) = \frac{\sum_{k=1}^{N} B_1(k,j)}{N}, \text{ where } 1 \le i \le C,\ 1 \le j \le K. \quad (11)$$

4.2.3.3. $L_{1,2}$: link conditions matrix

To facilitate the connections between the local MMM model and the integrated MMM model, the link conditions matrix $L_{1,2}$ is designed. Let $\{v_1, v_2, \ldots, v_M\}$ be the M videos and $\{s_1, s_2, \ldots, s_N\}$ be the N video shots. If s_i belongs to v_i, $L_{1,2}(i, j) = 1$ (where $1 \le i \le M$, $1 \le j \le N$). Otherwise, $L_{1,2}(i,j) = 0$.

5. Temporal Pattern Retrieval Process

Given a temporal pattern with C events $R = \{e_1, e_2, ..., e_C\}$ sorted by the temporal relationships such that $T_{e_1} \le T_{e_1} \le ... \le T_{e_C}$, Figure 2 presents our proposed retrieval process. Here, we assume there are M videos $\{v_1, \ldots, v_M\}$ in the multimedia database archive, and there are total K non-zero features $\{f_1, f_2, \ldots, f_K\}$ of the query sample. Here, $1 \le K \le 20$ since 20 features are used. As shown in Figure 2, our proposed retrieval process includes the following steps.

Step 1. Initialize the flag parameters as i=1, j=1, and k=1.

Step 2. The system checks matrix B_2 and/or matrix A_2 to search for video v_i which contains event e_j. This video should have a close affinity relationship with the previous video if it is available.

Step 3. Checks the link condition matrix $L_{1,2}$ and/or matrix A_1 to find the specified video shot which is annotated as event e_j or similar to event e_j. This video shot should also have a strong connection to the previous video shot.

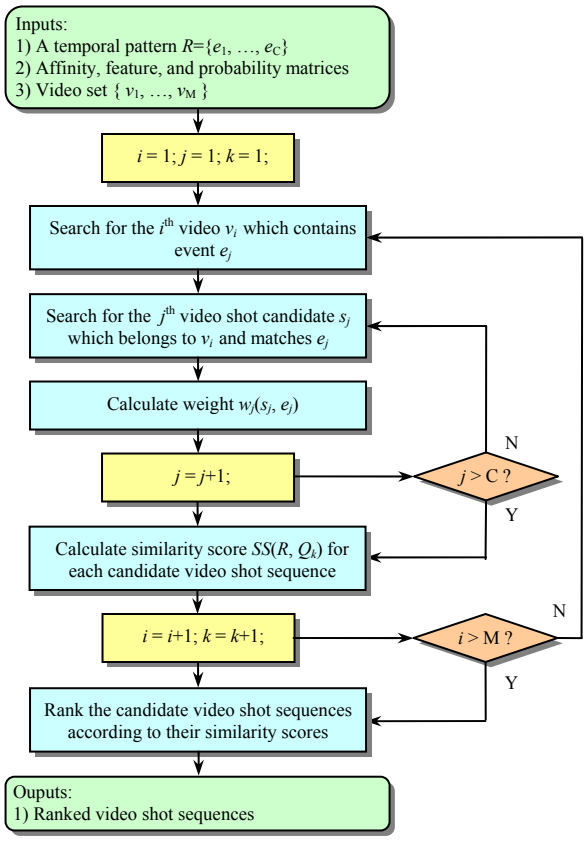

Figure 2. Flowchart of the video shot retrieval process

Step 4. Calculates the edge weight $w_j(s_j, e_j)$ using Equations (12) and (13), which is defined as the edge weight from the current state s_j to the target event e_j at the evaluation of the y^{th} feature (f_y) in the query, where $1 \le y \le K$ and $1 \le j \le C$.

At $j = 1$, $w_1(s_1, e_1) = \Pi_1(s_1) \times sim(s_1, e_1)$. (12)

When $1 \le j < C$,

$$w_{j+1}(s_{j+1}, e_{j+1}) = w_j(s_j, e_j) \times A_1(s_j, s_{j+1}) \times sim(s_{j+1}, e_{j+1}). \quad (13)$$

Equation (14) defines the similarity function to measure the similarity between s_j and e_j based on all of the non-zero features in $\{f_1, f_2, ..., f_K\}$.

$$sim(s_j, e_j) = \sum_{y=1}^{K} \left(P_{1,2}(e_j, f_y) \times \frac{(1 - |B_1(s_j, f_y) - B_1'(e_j, f_y)|)}{B_1'(e_j, f_y)}\right),$$
$$\text{where } s_j \in S_1, 1 \le y \le K, 1 \le j \le C. \quad (14)$$

Figure 3 shows the traversal paths when querying the target temporal pattern. In each traversal, the system will choose the optimized path to access the next possible video shot states similar to the anticipated events. At the

end of one video, the next possible video candidate will be selected by checking the higher-level affinity and feature matrices.

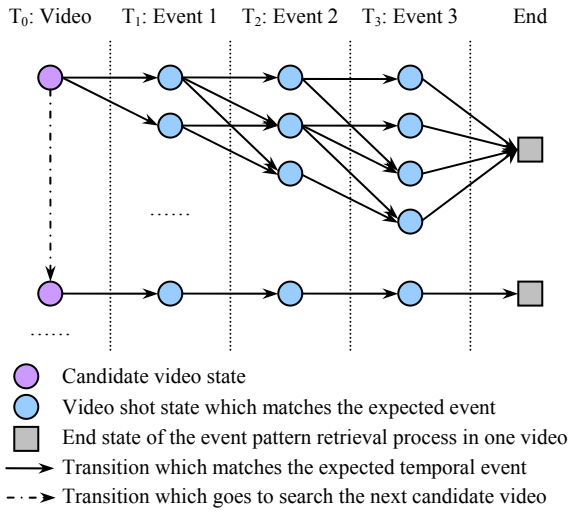

Figure 3. The lattice architecture for the temporal pattern retrieval process

Step 5. j=j+1. If j > C, it means that all the events in this pattern have been traversed and therefore the similarity score of the whole candidate pattern should be computed as indicated in Step 6. Otherwise, the system goes to Step 3 to continue checking the next video shot candidate which most closely matches the next event. Note that the traversal path should be recorded in the whole process.

Step 6. Assume a candidate video shot sequence is defined as $Q_k = \{s_1, s_2, ..., s_C\}$, the final similarity score can be calculated as:

$$SS(R, Q_k) = \sum_{j=1}^{C} w_j (s_j, e_j) . \quad (15)$$

Step 7. i = i+1; k = k+1. Check if i > M. If yes, it means that all the candidate video sets are checked and the system goes to Step 8. If no, the system goes to Step 2 and checks matrices A_2 and B_2 to find the next video candidate.

Step 8. There are k-1 candidate patterns. The system ranks the candidate video shot sequences according to the similarity scores.

Step 9. Finally, a list of k-1 sorted video shot sequences are retrieved as the outputs.

Here one query example is used to demonstrate the results of the retrieval mechanism. As illustrated in Figure 4, each temporal query pattern can be represented as a Multimedia Augmented Transition Network (MATN) [5], which is originally designed to describe multimedia presentations. The key frames of a set of retrieved temporal event patterns are displayed below the MATN model to show an example of the results.

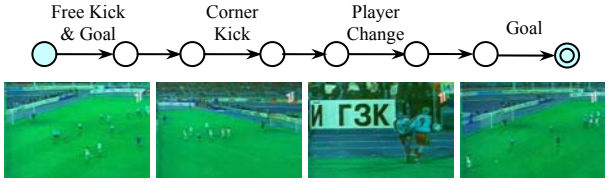

Figure 4. MATN-based query model and the result of a temporal pattern query

Figure 5. HMMM-based soccer video retrieval interface

A soccer video retrieval system has been developed for the evaluation of the proposed approach. In the current approach, the proposed two-level HMMM mechanism is utilized to model the multimedia database, where 54 soccer videos are segmented into 11,567 video shots. Among these video shots, 506 of them are annotated as semantic events. Figure 5 shows the client-side interface of the system, where the left-bottom part shows the interactive panels where a user can issue the queries. The right side panel demonstrates the resulting patterns sorted by their similarity scores. In this case, the target pattern is issued with a goal shot followed by a free kick, and therefore 8 patterns (including 16 shots) are displayed, where the magenta box marked the 3rd pattern. The left-upper panel displays the video shot which is chosen by the user. Finally, by using the drop down menu below the key frames, users are able to select their preferred video shots/patterns, and their feedbacks can be sent back to the server side for further improvement of the retrieval performance.

6. Conclusions

In this paper, we exemplify our efforts in the design of a generic mechanism for high-dimensional video database modeling and retrieval. A probabilistic-based approach called HMMM is proposed to integrate both low-level visual/audio features and high-level user perceptions along with the original multimedia data. Furthermore, the ranked temporal pattern query algorithm is proposed by considering the visual/audio features, temporal relationships, and user preferences. The retrieval procedure to search the specified temporal patterns becomes a stochastic process which always tries to traverse the most optimal path, thus guaranteeing the most efficient retrieval performance even in a large scale video database. Additionally, the overall system supports feedbacks and learning strategies since the user preferences can be proficiently analyzed and stored.

7. Acknowledgements

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.

8. References

[1] A. Amir, S. Basu, G. Iyengar, C.-Y. Lin, M. Naphade, J. R. Smith, S. Srinivasan and B. L. Tseng, "A Multi-modal System for the Retrieval of Semantic Video Events," *Journal of Computer Vision and Image Understanding*, vol. 96, no. 2, 2004, pp. 216-236.

[2] W. G. Aref, A. Catlin, J. Fan, A. K. Elmagarmid, M. A. Hammad, I. F. Ilyas, M. S. Marzouk and X. Zhu, "A Video Database Management System for Advancing Video Database Research," in *Proc. of the International Workshop on Multimedia Information Systems (MIS)*, Tempe, Arizona, USA, 2002, pp. 8-17.

[3] S.-F. Chang, W. Chen, H. Meng, H. Sundaram and D. Zhong, "A Fully Automated Content Based Video Search Engine Supporting Spatio-Temporal Queries," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 8, no. 5, 1998, pp. 602-615.

[4] L. Chen, M. T. Özsu and V. Oria, "Modeling Video Data for Content Based Queries: Extending the DISIMA Image Data Model," in *Proc. of the 9th International Conference on Multi-Media Modeling (MMM)*, Taiwan, 2003, pp. 169-189.

[5] S.-C. Chen, R. L. Kashyap and A. Ghafoor, *Semantic Models for Multimedia Database Searching and Browsing*, Springer, ISBN 0-7923-7888-1, September 2000.

[6] S.-C. Chen, M.-L. Shyu, C. Zhang, L. Luo, and M. Chen, "Detection of Soccer Goal Shots Using Joint Multimedia Features and Classification Rules," in *Proc. of the Fourth International Workshop on Multimedia Data Mining (MDM/KDD), in conjunction with the ACM International Conference on Knowledge Discovery & Data Mining (SIGKDD)*, Washington, DC, USA, 2003, pp. 36-44.

[7] S.-C. Chen, M.-L. Shyu, M. Chen and C. Zhang, "A Decision Tree-based Multimodal Data Mining Framework for Soccer Goal Detection," in *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)*, Taipei, Taiwan, R.O.C., 2004, pp. 265-268.

[8] S.-C. Chen, M.-L. Shyu and N. Zhao, "An Enhanced Query Model for Soccer Video Retrieval Using Temporal Relationships," in *Proc. of the 21st International Conference on Data Engineering (ICDE)*, Tokyo, Japan, 2005, pp. 1133-1134.

[9] D. DeMenthon and D. Doermann, "Video Retrieval using Spatio-Temporal Descriptors," in *Proc. of the 11th ACM International Conference on Multimedia (ACM MM)*, Berkeley, CA, USA, 2003, pp. 508-517.

[10] J. Fan , X. Zhu, A. K. Elmagarmid, W. G. Aref and L. Wu, "ClassView: Hierarchical Video Shot Classfication, Indexing, and Accessing," *IEEE Trans. on Multimedia*, vol. 6, no. 1, 2004, pp. 70-86.

[11] M. Guillemot, P. Wellner, D. Gatica-Perez and J.-M. Odobez, "A Hierarchical Keyframe User Interface for Browsing Video over the Internet," in *Proc. of the Ninth IFIP TC13 International Conference on Human-Computer Interaction (Interact)*, Zurich, Switzerland, 2003.

[12] IBM Marvel: MPEG-7 Multimedia Search Engine. http://www.research.ibm.com/marvel/

[13] IBM TRL's MPEG-7 Authoring System. http://www.trl.ibm.com/projects/digest/authoring_e.htm

[14] H. Kosch, L. Böszörményi, A. Bachlechner, B. Dörflinger, C. Hanin, C. Hofbauer, M. Lang, C. Riedler and R. Tusch, "SMOOTH - A Distributed Multimedia Database System," in *Proc. of the 27th International Conference on Very Large Database (VLDB)*, Rome, Italy, 2001, pp. 713-714.

[15] M.-L. Shyu, S.-C. Chen, M. Chen and C. Zhang, "A Unified Framework for Image Database Clustering and Content-based Retrieval," in *Proc. of the Second ACM International Workshop on Multimedia Databases* (ACM MMDB), Arlington, VA, USA , 2004, pp. 19-27.

[16] C.G.M. Snoek and M. Worring, "Multimedia Event based Video Indexing using Time Intervals," *IEEE Trans. on Multimedia*, vol. 7, no. 4, 2005, pp. 638-647.

[17] H. Yang, L. Chaisorn, Y. Zhao, S.-Y. Neo and T.-S. Chua, "VideoQA: Question Answering on News Video," in *Proc. of the 11th ACM International Conference on Multimedia (ACM MM)*, 2003, USA, pp. 632-641.