ClassMiner: Mining medical video content structure and events towards efficient access and scalable skimming[*]

Xingquan Zhu[a], Jianping Fan[b], Walid G. Aref[a], Ahmed K. Elmagarmid[c]

[a]*Department of Computer Science, Purdue University, W. Lafayette, IN, USA*

[b]*Department of Computer Science, University of North Carolina at Charlotte, NC, USA*

[c]*Hewlett Packard, Palo Alto, CA, USA*

{zhuxq, aref}@cs.purdue.edu; jfan@uncc.edu; ahmed_elmagarmid@hp.com

Abstract

To achieve more efficient video indexing and access, we introduce a video content structure and event mining framework. A video shot segmentation and key-frame selection strategy are first utilized to parse the continuous video stream into physical units. Video shot grouping, group merging, and scene clustering schemes are then proposed to organize the video shots into a hierarchical structure using clustered scenes, scenes, groups, and shots, in increasing granularity from top to bottom. Then, audio and video processing techniques are integrated to mine event information, such as dialog, presentation and clinical operation, among the detected scenes. Finally, the acquired video content structure and events are integrated to construct a scalable video skimming tool which can be used to visualize the video content hierarchy and event information for efficient access. Experimental results are also presented to evaluate the performance of the proposed algorithms.

1. Introduction

As a result of decreased costs for storage devices, increased network bandwidth, and improved compression techniques, digital videos are more accessible than ever. To help users find and retrieve relevant video effectively and to facilitate new and better ways of entertainment, advanced technologies must be developed for indexing, filtering, searching, and mining the vast amount of videos now available on the web. While numerous papers have appeared on video analysis and retrieval, few deal with video database management and mining [1-6]. There has recently been much interest in video database mining [7-9], however, most existing data mining techniques work on structured data, but video data are unstructured [7]. The existing data mining tools suffer from the following problems when applied to video database:

- **Database Model Problem:** Most traditional data mining techniques work on the relational database [1-3]. Unfortunately, video documents are generally unstructured in semantics and cannot be represented easily via the relational data model. A good video database model is necessary and critical to support more efficient video database management and mining.
- **Objective Problem:** Existing video retrieval systems first partition videos into a set of access units such as shots, objects, or regions [10, 17], and then follow the paradigm of representing video content via a set of feature attributes (i.e., metadata) such as color, texture, shape, motion and layout. Thus, video data mining can be achieved by performing the data mining techniques on the metadata directly. Unfortunately, there is a semantic gap between low-level visual features and high-level semantic concepts. The capability of *bridging the semantic gap* is the first requirement for existing data mining tools to be used for video data mining [7].

  There are several widely accepted data mining techniques [1-4], but most of them are unsuitable for video database mining because of the semantic gap. Classification via machine learning is an attractive technique for video database mining [7]. However, decision tree classifiers may consist of hundreds of thousands of internal nodes, which are consequently very difficult to comprehend and interpret. Moreover, the constructed tree structures do not make sense to the video database indexing. Detecting similar or unusual patterns is not the only objective for video data mining. The current challenge is to determine what type of outcome is most suitable for video data mining. The capability of *supporting more efficient video database indexing* is the second requirement for existing data mining tools to be applicable to video data mining.
- **Security Problem:** As more and more techniques are developed to access video data, there is an urgent need for video data protection [4, 11]. For example, one of the current challenges is to protect children from accessing inappropriate videos on the Internet. In addition, video data are often used in various environments with very different objectives. An effective video database management structure is needed to maintain data integrity and security. User-adaptive database access control is becoming an important topic in the areas of networks, database, national security, and social studies. Multilevel security is needed for access control of various video database applications. The capability of *supporting a secure and organized video access* is the third

[*] The support of *NSF* under Grants IIS-0093116, IIS-9974255, EIA-9972883, and EIA-9983249 are gratefully acknowledged.

requirement for the existing data mining tools to be applied to video data mining.

In this paper, we introduce our framework, *ClassMiner*, which makes some progress in addressing these problems. In Section 2, we present a database management model and our system architecture. A video content structure mining scheme is proposed in Section 3, and the event mining strategy among detected scenes is introduced in Section 4. Based on the acquired content structure and event information, a scalable video skimming tool is proposed in Section 5. Section 6 presents the results of algorithm evaluation and we conclude in Section 7.

2. Database management framework and system architecture

There are two widely accepted approaches for accessing video in databases: shot-based and object-based. In this paper, we focus on the shot-based approach. In order to meet the requirements for video data mining (i.e., bridging the semantic gap, supporting more efficient video database management, and access control), we classify video shots into a set of hierarchical database management units, as shown in Fig. 1. To support efficient video database mining, we need to address the following key problems: (a) How many levels should be included in the video database model, and how many nodes should be included in each level? (b) What kind of decision rules should be used for each node? (c) Do these nodes (i.e., database management units) make sense to human beings? In order to support hierarchical browsing and access control, the nodes in the database indexing tree must be meaningful to human beings.

We solve the first and third problems by deriving the database model from the concept hierarchy of video content. Obviously, the concept hierarchy is domain-dependent; a medical video domain is given in Fig. 2. This concept hierarchy defines the contextual and logical relationships between higher level concepts and lower level concepts. The lower the level of a node, the narrower is its coverage of the subjects; thus, database management units at a lower level characterize more specific aspects of the video content and units at a higher level describe more aggregated classes of video content. From the database model proposed in Fig.1 and Fig.2, we find that the most challenging task in solving the second problem is to determine how to map the physical shots at the lowest level with various predefined semantic scenes. In this paper, we will focus on mining video content structure and event information to attain this goal.

As shown in Fig. 3, we first utilize the general video shot segmentation and key-frame selection scheme to parse the video stream into physical units. Then, the video group detection, scene detection and clustering strategies are executed to mine the video content structure. Various visual and audio feature processing techniques are utilized to detect slides, face and speaker changes, etc. within the video, and these detection results are joined together to mine three types of events (presentation, dialog, clinical operation) from the detected video scenes. Finally, a scalable video skimming tool is constructed by utilizing mined video content structure and event information to help the user visualize and access video content effectively.

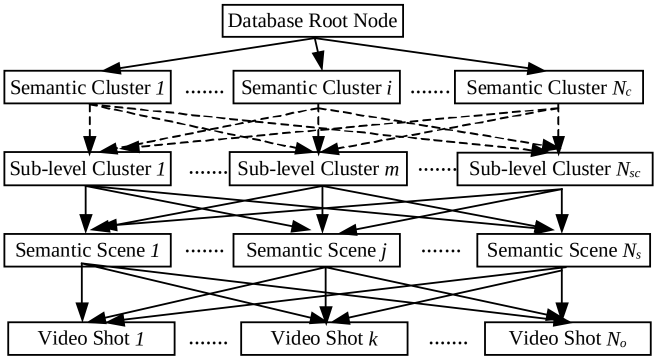

Figure 1. The proposed hierarchical video database model, where the cluster may include multiple levels according to the concept hierarchy, and a video scene consists of sequence of shots.

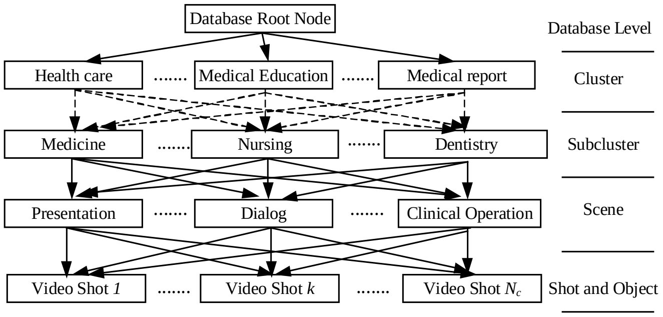

Figure 2. The concept hierarchy of video content in the medical domain, where the subcluster may consist of several levels

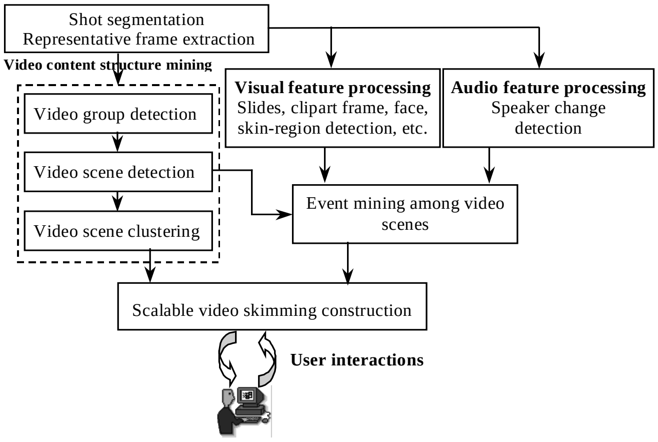
Shot segmentation
Representative frame extraction
Video content structure mining
Video group detection
Video scene detection
Video scene clustering
Visual feature processing
Slides, clipart frame, face, skin-region detection, etc.
Audio feature processing
Speaker change detection
Event mining among video scenes
Scalable video skimming construction
User interactions

Figure 3. System architecture

3. Video content structure mining

The simplest way to parse video data for efficient browsing, retrieval and navigating is to segment the continuous video sequence into physical shots, and then select one or more representative frames for each shot to depict its content information [12-13]. We use the same approach in our strategy. Video shots are first detected from a video using our shot detection techniques [10]. For the sake of simplicity, the 10^{th} frame of each detected shot is selected as its representative frame.

As we know, a video shot is a physical unit, and is usually incapable of conveying independent semantic information. Hence, various approaches are proposed to parse video content or scenario information. *Zhong et. al* [12] proposes a strategy which clusters visually similar shots and supplies the viewers with a hierarchical structure for browsing. However, since spatial shot clustering strategies consider only the visual similarity among shots, the video context information is lost. To address this problem, *Rui et. al* [14] presents a method which merges visually similar shot into groups, then constructs a video content table by considering the temporal relationships among groups. The same approach is reported in [16]. In [15], a time-constrained shot clustering strategy is proposed to cluster temporally adjacent shots into clusters, and a Scene Transition Graph is constructed to detect the video story unit by utilizing the acquired cluster information. A temporally time-constrained shot grouping strategy has also been proposed [17].

Generally, most videos from daily life can be represented using a hierarchy of five levels (video, scene, group, shot and frame)*, increasing in granularity from top to bottom. Hence, the most efficient way to address video content is to construct a video content hierarchy. As shown in Fig. 1, our video content structure mining is executed in three steps: (1) group detection, (2) scene detection, and (3) scene clustering. The video shots are first grouped into semantically richer groups. Then, similar neighboring groups are merged into scenes. Beyond the scene level, a pairwise cluster scheme is applied to eliminate repeated scenes in the video. And finally, the video content structure is constructed successfully.

***Remark:** In this paper, the video group and scene are defined as follows: (1) A *video group* is an intermediate entity between the physical shots and semantic scenes; examples of groups are temporally related shots or spatially related shots. (2) A *video scene* is a collection of semantically related and temporally adjacent groups depicting and conveying a high-level concept or story.

3.1 Video group detection

The shots in one group generally share a similar background or have a high correlation in time series. Therefore, to segment the spatially or temporally related video shots into groups, a given shot is compared with shots that precede and succeed it (using no more than 2 shots) to determine the correlation between them, as shown in Fig.4. Since closed caption and speech information is not available in our strategy, visual features such as color and texture play a more important role in determining the similarity between shots. We adopt a 256-bin dimensional *HSV* color histogram and 10-bin dimensional tamura coarsness texture for visual features. Suppose $H_{i,j}$, $j \in [0,255]$ and $T_{i,j}, j \in [0,9]$ are the normalized color histogram and texture of the key frame *i*. The similarity between shot *i*, *j* is defined by Eq. (1).

$$StSim\ (S_i, S_j) = W_c \sum_{k=0}^{255} \min(\ H_{i,k}, H_{j,k}) + W_T\,(1 - \sqrt{\sum_{k=0}^{9} (T_{i,k} - T_{j,k})^2}\,) \quad (1)$$

where S_i, S_j denote shot *i* and *j* respectively, W_C and W_T indicate the weight of color and tamura texture. For our system, we set W_C=0.7, W_T=0.3.

In order to detect the group boundary by using the correlation among adjacent video shots, we define the following similarity distances:

$$CL_i = Max\{\ StSim(S_i,S_{i-1}),\ StSim(S_i,S_{i-2})\} \quad (2)$$

$$CR_i = Max\{\ StSim(S_i,S_{i+1}),\ StSim(S_i,S_{i+2})\} \quad (3)$$

$$CL_{i+1} = Max\{\ StSim(S_{i+1},S_{i-1}),\ StSim(S_{i+1},S_{i-2})\} \quad (4)$$

$$CR_{i+1} = Max\{\ StSim(S_{i+1},S_{i+2}),\ StSim(S_{i+1},S_{i+3})\} \quad (5)$$

Given video shot S_i, if it is the first shot of a new group, it will have larger correlations with shots on its right side (as shown in Fig. 4) than shots on its left side, since we assume the shots in the same group usually have large correlations with each other. A *separation factor R(i)* for shot S_i is defined by Eq. (6) to evaluate a potential group boundary.

$$R(i)=(CR_i+CR_{i+1})/(CL_i+CL_{i+1}) \quad (6)$$

The shot group detection procedure then takes the following steps:

1. Given any shot S_i, if CR_i is larger than T_2-0.1:
   a. If $R(i)$ is larger than T_1, claim that a new group starts at shot S_i..
   b. Otherwise, go to step 1 to process other shots.
2. Otherwise:
   a. If both CR_i and CL_i are smaller than T_2, claim that a new group starts at shot S_i.
   b. Otherwise, go to step 1 to process other shots.
3. Iteratively execute step 1 and 2 until all shots are parsed successfully.

As the first shot of a new group, both CR_i and *R(i)* of shot S_i are generally larger than predefined thresholds. Step 1 is proposed to handle this situation. Moreover, there may be shot that is dissimilar with groups on its both sides, with itself acting as a group separator (like the anchor person in a News program.) Step 2 is used to detect such boundaries.

The threshold T_1 and T_2 can be automatically determined via a fast entropy technique [10].

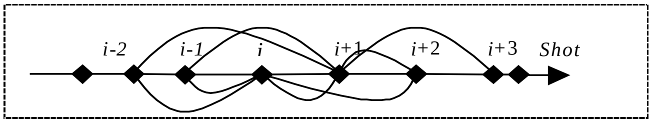

Figure 4. Correlations among video shots

With the strategy above, two kinds of shots are absorbed into a given group: (1) shots related in temporal series, where similar shots are shown back and forth. Shots in this group are referred to as *temporally related*, and (2) shots similar in visual perception, where all shots in the group are similar in visual features. Shots in this group are referred to as *spatially related*.

3.1.1 Group classification and representative shot selection

Given any detected group, G_i, we will classify it in one of two categories: temporally *vs.* spatially related group. Assume that there are T shots (S_i, $i=1,..,T$) contained in G_i. The group classification strategy is as follows:

Input: Video group G_i and shots S_i ($i=1,..,T$) in G_i. **Output:** Clusters (C_{Nc}, $N_c=1,..U$) of shots in G_i.

Procedure:

1. Initially, set variant $N_c=1$; cluster C_{Nc} has no members.
2. Select the shot (S_k) in G_i with the smallest shot number as the seed of cluster C_{Nc}, and subtract S_k from G_i. If there are no more shots contained in G_i, go to step 5.
3. Calculate the similarity between S_k and shot S_j in G_i, If $StSim(S_k,S_j)$ is larger than threshold T_h, absorb shot S_j into cluster C_{Nc}, and subtract S_j from G_i.
4. Iteratively execute step 3, until there are no more shots that can be absorbed into the current cluster C_{Nc}. Increase N_c by 1 and go to step 2.
5. If N_c is larger than 1, we claim G_i is a *temporally related* group, otherwise it is a *spatially related* group.

In order to support hierarchical video database indexing and summarization, the representative shot(s) of each group are selected to represent and visualize the content information in G_i. We denote this procedure as *SelectRepShot().*

[SelectRepShot]

The representative shot of group G_i is defined as the shot that represents the most content in G_i. Since semantic content is not available, we use visual features in our strategy. With the technique above, all shots in G_i are merged into N_c clusters, and these clusters will help us to select the representative shots for G_i. Given group G_i with N_c clusters (C_i) , we denote by $ST(C_i)$ the number of shots contained in cluster C_i. The representative shot of G_i is selected as follows:

1. Given N_c clusters C_i ($i=1,..,N_c$) in G_i, use steps 2, 3 and 4 to extract one representative shot for each cluster C_i. In all, N_c representative shots will be selected for G_i.
2. Given any cluster C_i which contains more than 2 shots, the representative shot of C_i (denote by $R_S(C_i)$) is obtained from Eq. (7)

$$R_s(C_i) = \arg\max_{S_j} \{\frac{1}{ST(C_i)} \sum_{k=1}^{ST(C_i)} StSim(S_j, S_k); \quad S_j \subset C_i, S_k \subset C_i\} \quad \underset{1 \le j \le ST(C_i)}{} \tag{7}$$

3. If there are 2 shots contained in C_i, the shot with larger time duration usually conveys more content information, and hence is selected as the representative shot of C_i.
4. If there is only 1 shot contained in cluster C_i, it is selected as the representative shot for C_i.

3.2 Video group similarity evaluation

As we stated above, video scenes consist of semantically related adjacent groups. To merge video groups for scene detection, the similarity between video groups must be determined. We first consider the similarity between the shot and group. Based on Eq. (1), given shot S_i and group G_j, the similarity between them is defined by Eq. (8).

$$StGpSim(S_i, G_j) = \underset{S_j \in G_j}{Max}\{StSim(S_i, S_j)\} \tag{8}$$

This implies that the similarity between S_i and G_j is the similarity between S_i and the most recent shot in G_j.

In general, when we evaluate the similarity between two groups using the human eye, we usually take the group with fewer shots as the benchmark, and then determine whether there are any shots in the second group similar to shots in the benchmark group. If most shots in the benchmark group are similar enough to the other group, they are treated as similar. Given group G_i and G_j, assuming $\hat{G}_{i,j}$ represents the group containing fewer shots, and $\tilde{G}_{i,j}$ denotes the other group. Suppose $NT(x)$ denotes the number of shot in group x, then, the similarity between G_i and G_j is given by Eq. (9).

$$GpSim(G_i, G_j) = \frac{1}{NT(\hat{G}_{i,j})} \sum_{i=1; S_i \in \hat{G}_{i,j}}^{NT(\hat{G}_{i,j})} StGpSim(S_i, \tilde{G}_{i,j}) \tag{9}$$

That is, the similarity between G_i and G_j is the average similarity between shots in the benchmark group and their most similar shots in the other group.

3.3 Group merging for scene detection

Since our shot grouping strategy places more emphasis on the details of the scene, one scene may be parsed into several groups. However, groups in the same scene generally have higher correlation with each other when compared with other groups in different scenes. Hence, we introduce a group merging method as follows:

1. Given groups G_i, $i=1,..,M$, calculate similarities between all neighboring groups (SG_i, $i=1,..,M-1$) using Eq. (10), where $GpSim(G_i,G_j)$ denotes the similarity between group G_i and G_j (given in Eq. (9)
$$SG_i=GpSim(G_i, G_{i+1}) \qquad i=1,..,M-1 \tag{10}$$
2. Collect all similarities SG_i, $i=1,..,M-1$, and apply the fast entropy technique [10] to determine the merging threshold T_G.
3. Adjacent groups with similarity larger than T_G are merged into a new group. If there are more than 2 sequentially adjacent groups with similarities larger than T_G, all are merged into a new group.
4. Those reserved and newly generated groups are formed as video scenes. Scenes containing less than three shots are eliminated, since they usually convey less semantic information than scenes with more shots. The *SelectRepGroup()* strategy is then used to select the representative group for each scene.

[SelectRepGroup]

For any scene, SE_i, the representative group is defined as the group in SE_i that contains the most content information of SE_i. As noted previously, the low-level features associated with each group are used in our strategy:

1. For any scene SE_i that contains three or more groups, G_j ($j=1,..,N_i$), the representative group of SE_i, $R_p(SE_i)$, is given by Eq. (11)

$$R_p(SE_i) = \arg\max_{G_j} \{ \frac{1}{N_i} \sum_{k=1}^{N_i} GpSim(G_j, G_k); \quad G_k \subset SE_i, G_j \subset SE_i \} \quad \underset{1 \le j \le N_i}{} \tag{11}$$

That is, $R_p(SE_i)$ is the group in SE_i which has the largest average similarity to all other groups.

2. If there are only two groups in SE_i, we use the number of shots and time duration in the group as the measurement. Usually, a group containing more shots will convey more content information, hence it is chosen as the representative group. If more than one group is selected, the group with longer time duration is selected as the representative group.
3. If there is only one group in SE_i, this group is selected as the representative group for SE_i.

In the sections below, the selected representative group $R_p(SE_i)$ is also taken as the centroid of SE_i.

3.4 Video scene clustering

Using the results of group merging, the video scene information is constructed. In most situations, many scenes are shown several times in the video. Clustering similar scenes into one unit will eliminate redundancy and produces a more concise video content summary. Since the general *K*-meaning cluster algorithm needs to seed the initial cluster center, and furthermore the initial guess of cluster centroids and the order in which feature vectors are classified can affect the clustering result, we therefore introduce a seedless *Pairwise Cluster Scheme* (*PCS*) for video scene clustering.

Input: Video scenes (SE_j, $j=1,..,M$) and all member groups (G_i, $i=1,..,NG$); **Output:** Clustered scene structure (SE_k, $k=1,..,N$).

Procedure:

1. Given video groups G_i, $i=1,..,NG$, we first calculate the similarities between any two groups G_i and G_j ($i=1,..,NG-1; j=1,..,NG-1$). The similarity matrix SM_{ij} for all groups is then constructed using Eq. (12).

   $$SM_{ij}(G_i,G_j)=GpSim(G_i,G_j),\ i=1,..,NG-1;\ j=1,..,NG-1 \tag{12}$$

   where $GpSim(G_i,G_j)$ denotes the similarity between G_i and G_j which is given by Eq. (9). Since any scene SE_j consists of one or more groups, the similarity matrix of all scenes (SM'_{ij}) can be derived from the group similarity matrix (SM_{ij}) using Eq. (13)

   $$SM'_{ij}(SE_i, SE_j) = GpSim(R_p(SE_i), R_p(SE_j));\ i, j \in [0, M], i \neq j \tag{13}$$

2. Find the largest value in matrix SM'_{ij}. Merge the corresponding scenes into a new scene, and use *SelectRepGroup()* to find the representative group (scene centroid) for the newly generated scene.
3. If we have obtained the desired number of clusters, go to the end. If not, go to step 4.
4. Based on the group similarity matrix SM_{ij} and the updated centroid of the newly generated scene, update the scene similarity matrix SM'_{ij} with Eq. (13) directly, then go to step 2.

To determine the end of scene clustering at step 3, the number of clusters *N* must be explicitly specified. Our experimental results suggest that for a significant number of interesting videos, if we have *M* video scenes, then using a clustering algorithm to reduce the number of scenes by 40% produces a relatively good result with respect to eliminating redundancy and reserving important scenario information. However, a fixed threshold often loses the adaptive ability of the algorithm. Therefore, to find an optimal number of clusters, we employ cluster validity analysis [21]. The intuitive approach is to find clusters that minimize intra-cluster distance while maximizing the inter-cluster distance. Assuming *N* denotes the number of clusters. Then the optimal cluster would result in measuring $\rho(N)$ with the smallest value, where $\rho(N)$ is defined in Eq. (14)

$$\rho(N) = \frac{1}{C_{\max} - C_{\min}} \sum_{i=C_{\min}}^{C_{\max}} \max_{C_{\min} \le j \le C_{\max}} \{ \frac{\varsigma_i + \varsigma_j}{\xi_{ij}} \} \tag{14}$$

where
$$\varsigma_i = \frac{1}{N_i} \sum_{k=1}^{N_i} (1 - GpSim(C_i^k, u_i)); \qquad \xi_{ij} = 1 - GpSim(u_i, u_j) \tag{15}$$

and N_i is the number of scenes in cluster *i*, and u_i is the centroid of the i^{th} cluster (C_i). Hence, ς_i is the intra-cluster distance of the cluster *i*, while ξ_{ij} is the inter-cluster distance of cluster *i* and *j*, and C_{min}, C_{max} are the ranges of the cluster numbers we seek for optimal values. We set these two numbers $C_{min}=[M \cdot 0.5]$ and $C_{max}=[M \cdot 0.7]$, where *M* is the number of scenes for clustering, and the operator [*x*] indicates the greatest integer which is not larger than *x*. That is, we seek optimal cluster number by eliminating 30% to 50% of the original scenes. Hence, the optimal number of cluster $\hat{N}$ is selected as:

$$\hat{N} = \underset{C_{\min} \le N \le C_{\max}}{Min} (\rho(N)) \tag{16}$$

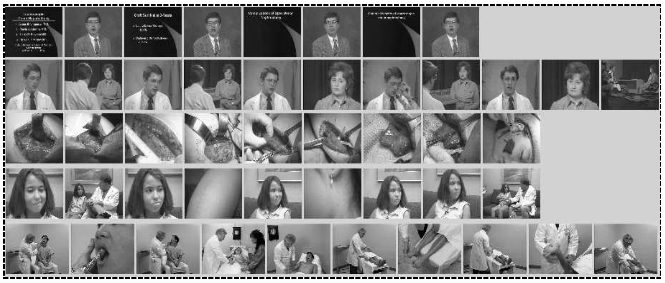

Figure 5. Video scene detection results

Fig. 5 presents the experimental results of video scene detection strategy. By utilizing the shot grouping and group merging, most scenes can be correctly detected.

4. Event mining among video scene

After video shots have been parsed into scenes, the event mining strategy is applied to detect the event information among the scenes. A successful result would satisfy a query such as “Show me all dialogs within the video.” Since medical videos are mainly used for educational purposes, the video content is usually recorded or edited using the style formats described below:

- Using presentations of the doctor or experts to express the general topics about the video.
- Using clinical operations (such as the diagnosis, surgery, organ pictures, etc.) to present details of the disease, their symptoms, comparisons and surgeries, etc.
- Using dialog between the doctor and patients to acquire other knowledge about the disease.

In this section, visual/audio features and rule information are integrated to mine these three types of events.

4.1 Visual feature processing

Visual feature processing is executed among all representative frames to extract semantically related visual cues. Currently, five types of special frame and regions are detected: slides or clip art frame, black frame, frame with face, frame with large skin area and frame with blood-red regions. Due to lack of space, we describe only the main idea; the algorithm details can be found in [18-20]. Since the slides, clip art frames and back frames are man-made frames, they contain less motion and color information when compared with other natural frame images. They also generally have very low similarity with other natural frames, and their number in the video is usually small. These features are utilized to detect slides, clip art and black frames. Following this step, the videotext and gray information are used to distinguish the slides, clip art and black frames from each other. To detect the faces, skin and blood-red regions, the Gaussian models are first utilized to segment the skin and blood-red regions, and then a general shape analysis is executed to select those regions that have considerable width and height. For skin-like regions, the texture filter and morphological operations are implemented to process the detected regions. A facial feature extraction algorithm is also introduced. Finally, a template curve-based face verification strategy is utilized to verify whether a face is in the candidate skin region.

4.2 Audio feature processing

Audio signals are a rich source of information in the video. It can be used to separate different speakers, detect various audio events, etc. In this paper, our objective is to verify whether speakers in different shots are the same person. The entire classification can be separated into two steps: (1) select the representative audio clip for each shot, and (2) compare whether representative clips of different shots belong to the same speaker.

For each video shot, we will separate the audio stream into adjacent clips, such that each is about 2 seconds long (a video shot of length less than 2 seconds is discarded), and then compute 14 audio features from each clip [22]. We classify each clip using the Gaussian Mixture Model (*GMM*) classifier into two classes: clean speech *vs* non-clean speech, and select the clip most like the speech clip as the audio representative clip of the shot. Given any audio representative clip of the shot S_i, a set of 14 dimensional mel frequency coefficients (*MFCC*) $X_i=\{x_1,..,x_{N_i}\}$ are extracted from 30 *ms* sliding windows with an overlapping of 20 *ms*. Then, the Bayesain Information Criterion (*BIC*) procedure is performed for comparison [23].

The *BIC* is a likelihood criterion penalized by the model complexity. Given $\chi=\{x_1,..,x_n\}$, a sequence of N_χ acoustic vectors, and $L(\chi,M)$, the likelihood of χ for the model *M*, the *BIC* value is determined by: $BIC(M)=\log L(\chi,M)-\lambda\frac{m}{2}\log N_\chi$, where *m* is the number of parameters of the model *M* and λ is the penalty factor. We assume that χ is generated by a multi-Gaussian process. Given shot S_i, S_j and their acoustic vectors $X_i=\{x_1,..,x_{N_i}\}$ and $X_j=\{x_1,..,x_{N_j}\}$, we consider the following hypothesis test for speaker change between S_i and S_j:

$$\begin{aligned} * \quad & H_0:(x_1,..,x_{N_\Re})\rightarrow N(u_\chi,\Sigma_\chi) \\ * \quad & H_1:(x_1,..,x_{N_i})\rightarrow N(u_{\chi_i},\Sigma_{\chi_i}) \\ & \textit{and} \quad (x_1,..,x_{N_j})\rightarrow N(u_{\chi_j},\Sigma_{\chi_j}) \end{aligned} \tag{17}$$

The variation of *BIC* between hypothesis H_0 (no speaker change) and H_1 (speaker change) is defined by Eq. (18):

$$\Delta BIC=\lambda P-(\frac{N_\Re}{2}\log|\Sigma_\chi|-\frac{N_i}{2}\log|\Sigma_{\chi_1}|-\frac{N_j}{2}\log|\Sigma_{\chi_j}|) \tag{18}$$

where $N_\Re=N_i+N_j$, $\Sigma_\chi,\Sigma_{\chi_i}$ and Σ_{χ_j} are, respectively the covariance matrices of the feature sequence $\{x_1,..,x_{N_i},..,x_{N_i+N_j}\}$, $\{x_1,..,x_{N_i}\}$ and $\{x_1,..,x_{N_j}\}$. The penalty is given by $P=\frac{1}{2}(p+\frac{1}{2}p(p+1))\log N_\Re$, where *p* is the dimension of the acoustic space, and λ is the penalty factor. If ΔBIC is less than zero, we claim a change of speaker between shots S_i and S_j.

4.3. Event mining strategy

Given any mined scene SE_i, our objective is to verify whether it belongs to one of the following event categories:

1. A "Presentation" scene is defined as a group of shots that contain slides or clip art frames. At least one group in the scene should consist of temporally related shots. Moreover, at least one shot should contain a close-up face (human face with size larger than 10% of the total frame size), and there is no speaker change between adjacent shots.
2. A "Dialog" scene is a group of shots containing both face and speaker changes. Moreover, at least one group in the scene should consist of spatially related shots. The speaker change should take place at adjacent shots, which both contain the face. At least one speaker should be duplicated more than once.
3. The "Clinical operation" scene includes three kinds of medical events, such as surgery, diagnosis, symptoms, etc. In this paper, we define the "Clinical operation" as a group of shots without speaker change, where at east one shot in SE_i contains blood-red or a close-up of a skin region (skin region with its size is larger than 20% of the total frame

size) or where more than half of shots in SE_i contain skin regions.

Based on the above definitions, the event mining is executed as follows.

1. Input all shots in SE_i and their visual/audio preprocessing results.
2. Test whether SE_i belongs to a "Presentation" scene:
   a. If there is no slide or clip art frame contained in SE_i, go to step 3. If there is no close-up face contained in SE_i, go to step 3.
   b. If all groups in SE_i consist of spatially related shots, go to step 3.
   c. If there is any speaker change between adjacent shots of SE_i, go to step 3,
   d. Assign the current group to the "Presentation" category; go to end or process other scenes.
3. Test whether SE_i belongs to "Dialog":
   a. If there is either no face or no adjacent shots which both contain faces in SE_i, go to step 4.
   b. If all groups in SE_i consist of spatially related shots, go to step 4.
   c. If there is no speaker change between all adjacent shots which both contain faces, go to step 4.
   d. Among all adjacent shots which both contain face and speaker change, if there are two or more shots belonging to the same speaker, SE_i is claimed as a "Dialog", otherwise, go to step 4.
4. Test whether SE_i belongs to "Clinical Operation":
   a. If there is a speaker change between any adjacent shots, go to step 5.
   b. If there are any close-up skin region or blood-red regions detected, SE_i is assigned to "Clinical Operation".
   c. If more than half of representative frames of all shots in SE_i contain skin regions, then SE_i is assigned as "Clinical Operation." Otherwise, go to step 5.
5. Claim the event in SE_i cannot be determined and process another scene.

5.Scalable video skimming system

Based on mined video content structure and events, a scalable video skimming tool has been developed to visualize an overview of the video and help the user access the video content effectively, as shown in Fig. 6. Currently, a four layer video skimming is constructed, with the level 4 to level 1 video skimming consisting of representative shots of clustered scenes, all scenes, all groups, and all shots respectively. Hence, the granularity of video skimming increases from level 4 to level 1. A user can change to different levels of video skimming by clicking the up or down arrow. While video skimming is playing, only those selected skimming shots are shown, and all other shots are skipped. A scroll bar indicates the position of the current skimming shot among all shots in the video. The user can drag the tag of the scroll bar to fast-access the interesting video unit.

To help users visualize the mined events, understand video content structure, and access the video more effectively, a color bar with each color representing one event type has been constructed, as shown in Fig. 6.

6. Experimental results

To illustrate the performance of the proposed strategies, two types of experimental results, video scene detection and event mining, are presented in this section. Our dataset consists of approximately 6 hours of *MPEG-I* encoded medial videos which describe face repair, nuclear medicine, laparoscopy, skin examination, and laser eye surgery. Fig.7 presents the experimental results and comparisons between our scene detection algorithm and other strategies [14, 17]. To judge the quality of the detected results, the following rule is applied: the scene is judged to be rightly detected if and only if all shots in the current scene belong to the same semantic unit (scene), otherwise the current scene is judged to be falsely detected. Thus, the scene detection precision (*P*) in Eq. (19) is utilized for performance evaluation.

$$P = \text{Rightly detected scenes / All detected scenes} \quad (19)$$

Clearly, without any scene detection (that is, treating each shot as one scene), the scene detection precision would be 100%. Hence, a *compression rate factor* (*CRF*) is defined in Eq. (20).

$$CRF = \text{Detected scene number / Total shot number} \quad (20)$$

To show both *CRF* and *P* in the same figure, we multiply *CRF* by 10. We denote our method as *A*, and the two methods from the literature [14] and [17] as *B* and *C* respectively. From the results in Fig. 6, some observations can be made: (1) our scene detection algorithm achieves the best precision among all three methods, about 65% shots are assigned to the appropriate semantic unit, (2) method *C* achieves the highest compression rate, unfortunately the precision of this method is also the lowest, and (3) as a tradeoff with precision, the compression ratio of our method is the lowest (*CRF*=8.6%, each scene consists of about 11 shots). We believe that in semantic unit detection, it is worse to fail to segment distinct boundaries than to over-segment a scene. From this point of view, our method is better than other two methods.

After the video content structure has been mined, we manually select scenes which distinctly belong to one of the following event categories: presentation, dialog and clinical operation, and use them as a benchmark. We then apply the event mining algorithm to automatically determine their event category. The experimental results are shown in Table 1, where *PR* and *RE* represent the precision and recall which are defined in Eq. (21) and Eq. (22), respectively. On average, our system achieves relatively good performance (72% in precision and 71% in recall) when mining these three types of events.

$$PR = \text{True Number / Detected Number} \quad (21)$$

$$RE = \text{True Number / Selected Number} \quad (22)$$

7.Conclusion

In this paper, we have addressed video mining techniques for efficient video database indexing and access. To achieve this goal, a video database management framework is first proposed. Then, a video content structure mining strategy is adopted to parse the video shots into a hierarchical structure using shots, groups, scenes, and clustered scenes by applying a shot grouping and clustering strategy. Meanwhile, both visual and audio feature processing techniques are proposed to extract the semantic cues within each scene. Afterward, a video event mining algorithm is then introduced by integrating visual and audio cues to detect three types of events: presentation, dialog and clinical operation. Finally, by integrating the mined content structure and events information, a scalable video skimming and content access prototype system is constructed to help the user visualize the overview and access video content more efficiently.

References

1. R. Agrawal, T. Imeielinski, and A. Swami, "Data mining: A performance perspective", *IEEE TKDE, 5(6), p.914-925, 1993.*
2. U. Fayyad and R. Uthurusamy, "Data mining and knowledge discovery in database", *Communication of ACM, 39*(*11*)*, 1996.*
3. M.S. Chen, J. Han and P.S. Yu, "Data mining: An overview from a database perspective", *IEEE TKDE, 8(6), 1996.*
4. B. Thuraisingham, "Managing and mining multimedia database", *CRC Press, 2001.*
5. J. Han and M. Kamber, "Data Mining: Concepts and techniques", *Morgan Kaufmann Publishers, 2001.*
6. O.R. Zajane, J. Han Z.N. Li and J. Hou, "Mining multimedia data", *Proc. of SIGMOD, 1998.*
7. J. Fan, X. Zhu and X. Lin, "Mining of video database", *Book chapter in Multimedia data mining, Kluwer, 2002.*
8. J. Y. Pan, C. Faloutsos, "VideoGraph: A new tool for video mining and classification", *JCDL June, 2001, Virginia, USA.*
9. S.C. Chen, M.L. Shyu, C. Zhang, J. Strickrott, "Multimedia data mining for traffic video sequence*", MDM/KDD workshop 2001, San Francisco, USA.*
10. J. Fan, W.G. Aref, A.K. Elmagarmid, M.S. Hacid, M.S. Marzouk and X. Zhu, "Multiview: multilevel video content representation and retrieval", *Journal of electronic imaging, vol.10, no.4, pp.895-908, 2001.*
11. E. Bertino, J. Fan, E. Ferrari, M.S. Hacid and A.K. Elmagarmid, "A hierarchical access control model for video database system", *ACM Trans. on Info. Syst., vol.20, 2002.*
12. H.J. zhang, A. Kantankanhalli, and S.W. Smoliar, "Automatic partitioning of full-motion video", ACM Multimedia system, vol.1, no.1, 1993.
13. D. Zhong, H. J. Zhang and S.F. Chang, "Clustering methods for video browsing and annotation", *Technical report, Columbia Univ.,1997.*
14. Y. Rui, T.S. Huang, S. Mehrotra, "Constructing table-of-content for video"*, ACM MSJ, Vol.7, No.5, pp 359-368. 1999.*
15. M.M. Yeung, B.L. Yeo, "Time-constrained clustering for segmentation of video into story units", *Pro. of ICPR'96.*
16. J.R. Kender, B.L. Yeo, "Video scene segmentation via continuous video coherence", *Proc. Of CVPR 1998.*
17. T. Lin, H.J. Zhang "Automatic Video Scene Extraction by Shot Grouping", *Proc. ICPR 2000.*
18. J.P. Fan, X. Zhu, L.D. Wu, "Automatic model-based semantic object extraction algorithm", *IEEE CSVT, 11(10), pp.1073-1084, Oct., 2000.*
19. X. Zhu, J. Fan, A.K. Elmagarmid, W.G. Aref, "Hierarchical video summarization for medical data", *Proc. of IST/SPIE storage and retrieval for media database, pp.395-406, 2002.*
20. X. Zhu, J. Fan, A.K. Elmagarmid, "Towards facial feature localization and verification for omni-face detection in video/images", *Prof. of IEEE ICIP, 2002.*
21. A.K. Jain, "Algorithms for clustering data", *Prentice Hall, 1998.*
22. Z. Liu and Q. Huang, "Classification of audio events in broadcast News", *MMSP-98, pp.364-369, 1998.*
23. P. Delacourt, C, J Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing", *Speech communication, vol.32, p.111-126, 2000.*

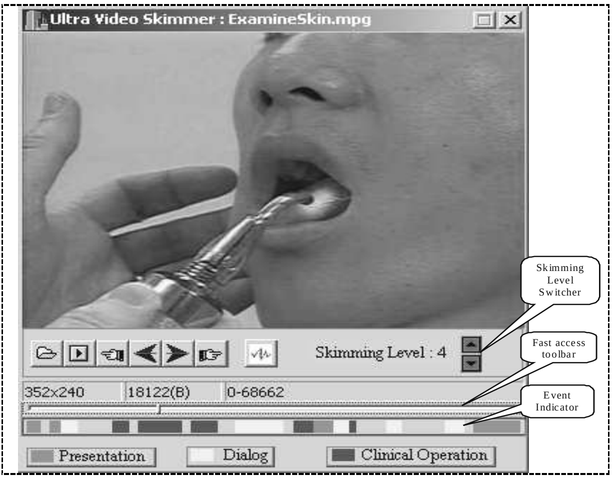

Figure 6. Scalable video skimming tool

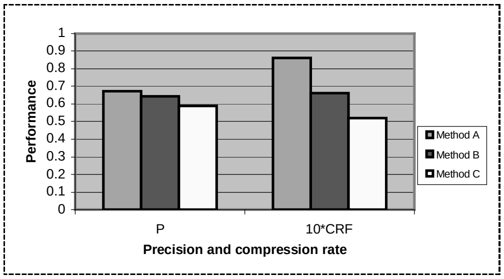

Figure 7. Scene detection performance

Table 1. Video event mining results

Events	Selected number	Detected number	True number	*PR*	*RE*
presentation	15	16	13	0.81	0.87
dialog	28	33	24	0.73	0.85
clinical operation	39	32	21	0.65	0.54
average	*82*	*81*	*58*	*0.72*	*0.71*