Medical Video Mining for Efficient Database Indexing, Management and Access

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

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

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

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

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

Abstract

To achieve more efficient video indexing and access, we introduce a video database management framework and strategies for video content structure and events mining. The video shot segmentation and representative 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, from 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 framework and 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 more 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 video now available on the web. While numerous papers have appeared on data mining [1-6], few deal with video data mining directly [7-9][24][27]. We can now use various video processing techniques to segment video sequence into shots, region-of-interest (*ROI*), etc. for database management or retrieval; however, the mining of knowledge from video data is still in its infancy. To facilitate the mining process, these questions must first be answered:

1. What is the objective of video data mining?
2. Can general data mining techniques be used on video data directly?
3. If not, what requirements would allow general mining techniques to be applied to video data?

Simply stated, the objective of video data mining is the "organizing" of video data for knowledge "exploring" or "mining", where the knowledge can be explained as special patterns (e.g., events, clusters, classification, etc.) which may be unknown before the processing. Many successful data mining techniques have been developed through academic research and industry [1-6], hence, an intuitive solution for video data mining is to use these strategies on video data directly; Unfortunately, due to the inherent complexity of the video data, existing data mining tools suffer from the following problems when applied to video databases:

- **Video database modeling**

Most traditional data mining techniques work on the relational database [1-3], where the relationship between data items is explicitly specified. However, video documents are obviously not a relational dataset, and although we may now retrieve video frames (and even physical shots) with satisfactory results, acquiring the relational relationships among video frames (or shots) is still an open problem. Consequently, traditional data mining techniques cannot be utilized in video data mining directly; a distinct database model must first be addressed.

- **Video data organizing**

Existing video retrieval systems first partition videos into a set of access units such as shots, 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, shape, motion etc. Thus, video mining can be achieved by applying the data mining techniques to the metadata directly. Unfortunately, there is a semantic gap

This research is supported by *NSF* under grant 0209120-IIS, 0208539-IIS, 0093116-IIS, 9972883-EIA, 9974255-IIS and by the NAVSEA/Naval Surface Warfare Center, Crane.

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

between low-level visual features and high-level semantic concepts. The capability of *bridging the semantic gap* is the first requirement for existing mining tools to be used in video mining [7]. To achieve this goal, some video mining strategies use closed-caption or speech recognition techniques [24][27]; however, for videos that have no closed-caption or low audio signal quality (such videos represent a majority of the videos from our daily lives) these approaches are invalid.

On the other hand, most schemes use low-level features and various indexing strategies, e.g. Decision tree [7], R-tree [26], etc. for video content management. The results generated with these approaches may consist of thousands of internal nodes, which are consequently very difficult to comprehend and interpret. Moreover, the constructed tree structures do not make sense to database indexing and human perception. Detecting similar or unusual patterns is not the only objective for video mining. The current challenge is how to *organize* video data for effective mining. The capability of *supporting more efficient video database indexing* is the second requirement for existing mining tools to be applicable to video mining.

- **Security and privacy**

As more and more techniques are developed to access video data, there is an urgent need for 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 requirement for the existing mining tools to be applied to video data mining.

In this paper, we introduce a framework, *ClassMiner*, which makes some progress in addressing these problems. In Section 2, we present the 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. 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? To support hierarchical browsing and user adaptive access control, the nodes 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 to various predefined semantic scenes. In this paper, we will focus on mining video content structure and event information to attain this goal. Based on the results of our mining process, we have developed a prototype system, ClassMiner, with the following features:

- ***A semantic-sensitive video classifier*** to narrow the semantic gap between the visual features and the high-level semantic concepts. The hierarchical structure of our semantic-sensitive video classifier is derived from the concept hierarchy of video content and is provided by domain experts or obtained using WordNet [25]. Each node in this classifier defines a semantic concept and thus makes sense to human beings. The contextual and logical relationships between higher-level nodes and their sub-level nodes are derived from the concept hierarchy.
- ***A hierarchical video database management and visual summarization technique*** to support more effective video access. The video database indexing and management structure is inherently provided by the semantic-sensitive video classifier. The organization of visual summaries is also integrated with the inherent hierarchical database indexing structure. For the leaf node of the proposed hierarchical indexing tree, we use a hash table to index video shots. For the non-leaf node (nodes representing high-level visual concepts), we use multiple centers to index video shots because they may consist of multiple low-level components, and it

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

would be very difficult to use any single *Gaussian* function to model their data distribution.

- ***A hierarchical video database access control technique*** to protect the video data and support a secure and organized access. The inherent hierarchical video classification and indexing structure can support a wide range of protection granularity levels, for which it is possible to specify filtering rules that apply to different semantic concepts.

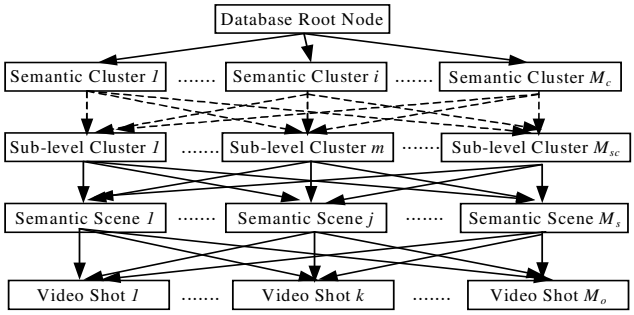

Figure 1. The proposed hierarchical video database model

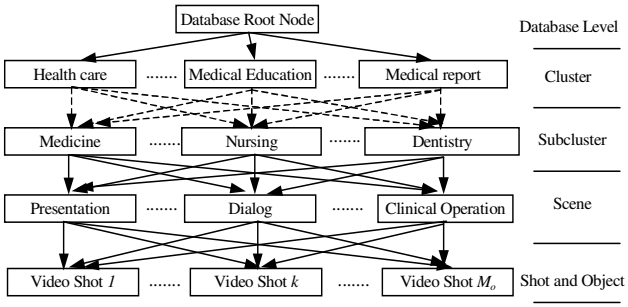

Figure 2. The concept hierarchy of video content in the medical domain, where the subcluster may consist of several levels

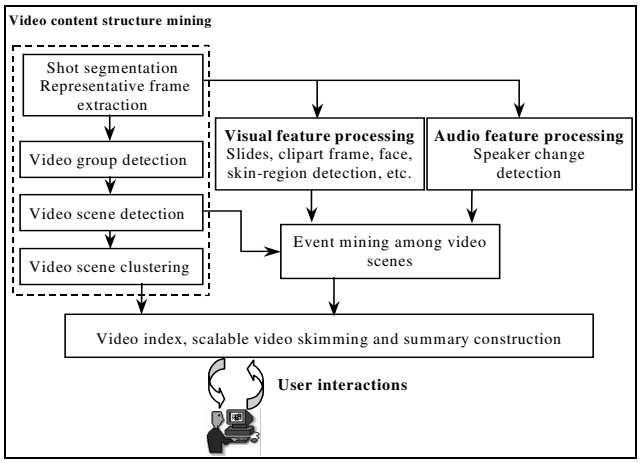

Figure 3. Video mining and scalable video skimming/summarization structure

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

3. Video Content Structure Mining

In general, most videos from daily life can be represented using a hierarchy of five levels (video, scene, group, shot and frame), as shown in Fig. 4. To clarify our objective, we first present the definition of *video content structure*.

Definition 1: The *video content structure* is defined as a hierarchy of clustered scenes, video scenes, video groups and video shots (whose definitions are given below), increasing in granularity from top to bottom. Although there exist videos with very little content structure (such as sports videos, surveillance videos etc.), a content structure can be found in most videos from our daily lives.

Definition 2: In this paper, the video shot, video group, video scene and clustered scene are defined as follows:

- A *video shot* (denoted by S_i) is the simplest element in videos and films; it records the frames resulting from a single continuous running of the camera, from the moment it is turned on to the moment it is turned off.
- A *video group* (denoted by G_i) is an intermediate entity between the physical shots and semantic scenes; examples of groups consist of temporally or spatially related video shots.
- A *video scene* (denoted by SE_i) is a collection of semantically related and temporally adjacent groups depicting and conveying a high-level concept or story.
- A *clustered scene* (CSE_i) is a collection of visually similar video scenes that may be shown in various places in the video.

Usually, the simplest way to parse video data for efficient browsing, retrieval and navigation is to segment the continuous video sequence into physical shots, and then select representative frame(s) for each shot to depict its content information [12-13]. However, a video shot is a physical unit and is usually incapable of conveying independent semantic information. Accordingly, various approaches have been 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.

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

However, since spatial 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 in [17].

As shown in Fig.1, the most efficient way to address video content for indexing, management, etc. is to acquire the video content structure. Fig. 3 shows that our video content structure mining is executed in four steps: (1) video shot detection, (2) group detection, (3) scene detection, and (4) scene clustering. The continuous video sequence is first segmented into physical shots, and the video shots are then grouped into semantically richer groups. Afterward, similar neighboring groups are merged into scenes. Beyond the scene level, a cluster scheme is adopted to eliminate repeated scenes in the video. Finally, the video content structure is constructed.

3.1 Video shot detection

To support shot based video content access, we have developed a shot cut detection technique [10]. Our shot cut detection technique can adapt the threshold for video shot detection according to the activities of various video sequences, and this technique has been developed to work on MPEG compressed videos. Unfortunately, such techniques are not able to adapt the thresholds for different video shots within the same sequence.

In order to adapt the thresholds to the local activities of different video shots within the same sequence, we use a small window (e.g., 30 frames in our current work) and the threshold for each window is adapted to its local visual activity by using our automatic threshold detection technique and local activity analysis. The video shot detection result shown in Fig.5 is obtained from one video data source used in our system. It can be seen that by integrating local thresholds, a satisfactory detection result is achieved (The threshold has been adapted to the small changes between adjacent shots, such as changes between eyeballs from various shots in Fig. 5, for successful shot segmentation). After shot segmentation, the 10th frame of each shot is taken as the representative frame of the current shot, and a set of visual features (256 dimensional *HSV* color histogram and 10 dimensional tamura coarseness texture) is extracted for processing.

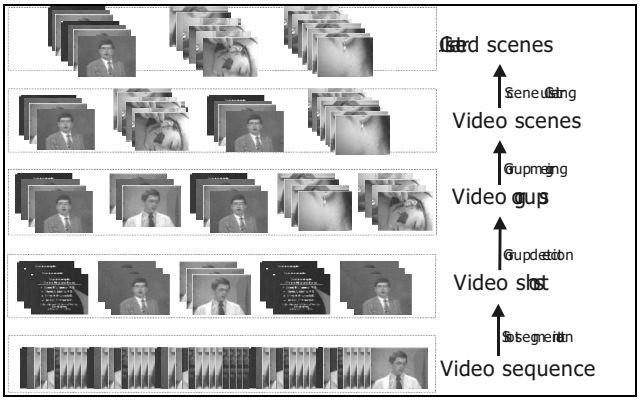

Figure 4. Pictorial video content structure

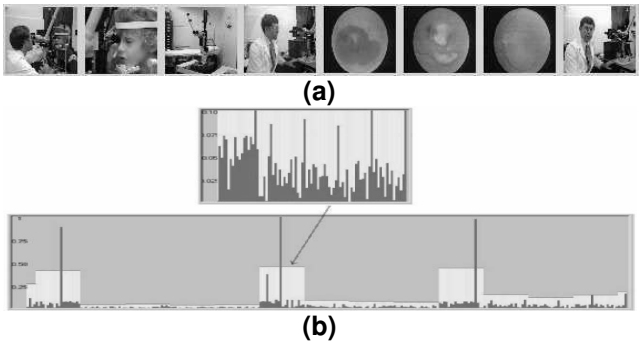

Figure 5. The video shot detection results from a medical education video: (a) part of the detected shot boundaries; (b) the corresponding frame difference and the determined threshold for different video shots, where the small window shows the local frame differences.

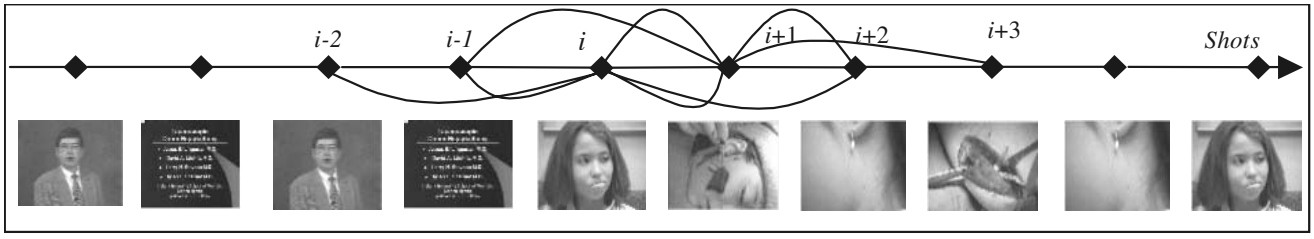

Figure 6. Exploring correlations among shots for video group detection

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

3.2 Video group detection

The shots in the same video group generally share 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.6. We adopt 256-color histogram and 10-tamura coarseness texture for visual features. Suppose $H_{i,k}$, $k \in [0,255]$ and $T_{i,k}$, $k \in [0,9]$ are the normalized color histogram and texture of the representative frame of shot i. The similarity between shot i, j is then defined by Eq. (1).

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

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

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

Accordingly, the video group detection procedure 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 a shot that is dissimilar with groups on its both sides, where the shot itself acts as a group separator (like the anchor person in a News program.) Step 2 is used to detect such boundaries. The thresholds T_1 and T_2 can be automatically determined via the fast entropy technique described in [10].

Using this strategy, 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 relatively similar in visual features. Shots in this group are referred to as *spatially related*.

3.2.1 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. A successful classification will help us in determining the feature variance in G_i and selecting its representative shot(s). 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_{N_c}, $N_c=1,..K$) of shots in G_i.

Procedure:

1. Initially, set variant $N_c=1$; cluster C_{N_c} has no members.
2. Select the shot (S_k) in G_i with the smallest shot number as the seed of cluster C_{N_c}, 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 other shots S_j in G_i, If $StSim(S_k,S_j)$ is larger than threshold T_h, absorb shot S_j into cluster C_{N_c}, and subtract S_j from G_i.
4. Iteratively execute step 3, until there are no more shots that can be absorbed into current cluster C_{N_c}. 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 its content information. We denote this procedure as *SelectRepShot()*.

[SelectRepShot]

The representative shot of group G_i is defined as the shot that best represents the content in G_i. Since semantics is not available, we use only visual features in our strategy. With the technique above, all shots in G_i have been 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

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

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)

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

That is, among all shots in C_i, the shot which has the largest average similarity with others shots is selected as the representative shot of C_i.

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.

3.3 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 a shot and group. Based on Eq. (1), given shot S_i and group G_j, the similarity between them is defined by Eq. (8).

$$StGpSim\ (S_i, G_j) = Max_{S_k \in G_j}\{StSim\ (S_i, S_k)\} \quad (8)$$

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 video groups using the human eye, we take the group with fewer shots as the benchmark, and then determine whether there is any shot in the second group similar to shots in the benchmark group, as shown in Fig.7. If most shots in the benchmark group are similar enough to the shots in the other group, they are treated as similar. Given group G_i and G_j, assume $\hat{G}_{i,j}$ 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).

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

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.

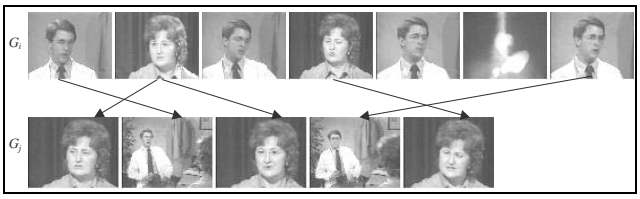

Figure 7. Group similarity evaluation (arrows indicate the most similar shots in G_i and G_j)

3.4 Group merging for scene detection

Since our group detection scheme places much emphasis on 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 than with other groups from different scenes. Hence, we introduce a group merging method for scene detection as follows:

1. Given M 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.

$$SG_i = GpSim(G_i, G_{i+1});\qquad i=1,..,M-1 \quad (10)$$

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. Merge adjacent groups with similarity larger than T_G into a new group. If there are more than 2 sequentially adjacent groups with their 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 others. The *SelectRepGroup()* strategy is then used to select the representative group of each scene for content representation and visualization.

[SelectRepGroup]

For any scene, SE_i, the representative group is defined as the video group in SE_i that represents 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 ($j=1,..,N_i$), its representative group, $R_p(SE_i)$, is given by Eq. (11)

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

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

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

2. If there are only two groups in SE_i, we use the number of shots and time duration in the group as the measure. 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 by this rule, the group with longer time duration is selected as the representative group.
3. If there is only one group in SE_i, it 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.

3.5 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 produce 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, $i=1,..,NG$) contained in them.

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 ($i, j \in [1, NG-1]; i \neq j$). The similarity matrix SM_{ij} for all groups is then constructed using Eq. (12).

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

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 video groups, the similarity matrix of all scenes (SM'_{ij}) can be derived from the similarity matrix (SM_{ij}) of their representative groups by using Eq. (13)

$$SM'_{ij}(SE_i, SE_j) = GpSim(R_p(SE_i), R_p(SE_j)); \quad i, j \in [1, M-1], i \neq j \tag{13}$$

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* 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 reduces 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 inter-cluster distance. Assume *N* denotes the number of clusters. Then the optimal cluster would result in the $\rho(N)$ with smallest value, where $\rho(N)$ is defined in Eq. (14)

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

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

where 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 the optimal value. 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:

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

Fig. 8 presents several experimental results from medical videos. By applying shot grouping and group merging, some typical video scenes can be correctly detected.

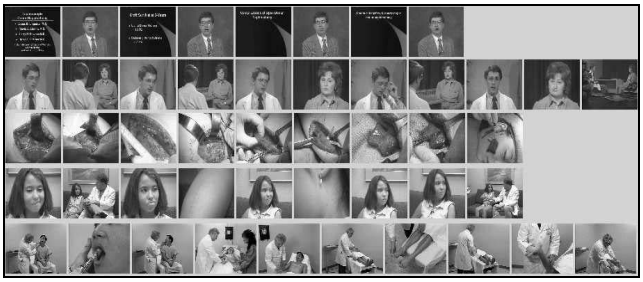

Figure 8. Video scene detection results, with the rows from top to bottom indicating "Presentation", "Dialog", "surgery", "Diagnosis" and "Diagnosis", respectively.

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

4. Event Mining Among Video Scene

After the video content structure has been mined, the event mining strategy is applied to detect the event information within each detected scene. A successful result will not only satisfy a query such as "Show me all patient-doctor dialogs within the video", it will also bridge the inherent gap between video shots and their semantic categories for efficient video indexing, access and management. Since medical videos are mainly used for educational purposes, video content is usually recorded or edited using the style formats described below (as shown in the pictorial results in Fig.8):

- Using presentations of doctors or experts to express the general topics of 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 (or between the doctors) to acquire other knowledge about medical conditions.

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 frames 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, as shown in Fig.9 and Fig.10. Due to the lack of space, we will describe only the main idea; 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 distinct features are utilized to detect slides, clip art and black frames. Following this step, the video text and gray information are used to distinguish the slides, clip art and black frames from each other. To detect faces, skin and blood-red regions, *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, texture filter and morphological operations are implemented to process the detected regions. A facial feature extraction algorithm is also applied. Finally, a template curve-based face verification strategy is utilized to verify whether a face is in the candidate skin region.

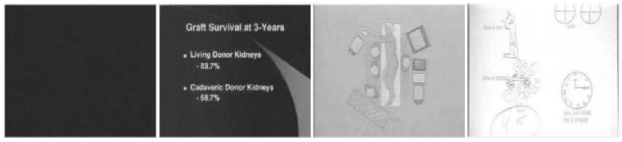

Figure 9. Visual feature cues (special frames), with frames from let to right depicting the black frame, slide, clipart, and sketch frame.

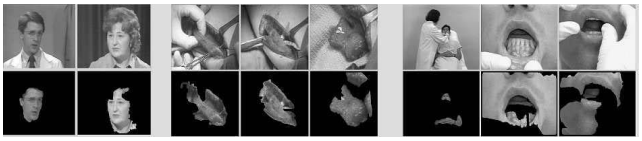

Figure 10. Visual feature cues (special regions). With (*a*), (*b*) and (*c*) indicating the face, blood-red and skin regions.

4.2 Audio feature processing

Audio signals are a rich source of information in videos. They 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) determine whether representative clips of different shots belong to the same speaker.

For each video shot, we separate the audio stream into adjacent clips, such that each is about 2 seconds long (a video shot with its 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.

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

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:

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

The maximum likelihood ratio between hypothesis H_0 (no speaker change) and H_1 (speaker change) is then defined by Eq. (18):

$$R(\Lambda) = \frac{N_{\Re}}{2}\log|\Sigma_{\chi}| - \frac{N_i}{2}\log|\Sigma_{\chi_i}| - \frac{N_j}{2}\log|\Sigma_{\chi_j}| \tag{18}$$

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 variations of the *BIC* value between the two models (one Gaussian *vs* two different Gaussians) is then given by Eq. (19):

$$\Delta BIC\ (\Lambda) = -R(\Lambda) + \lambda P \tag{19}$$

where the penalty is given by $P = \frac{1}{2}(p + \frac{1}{2}p(p+1))\log N_{\Re}$, p is the dimension of the acoustic space, and λ is the penalty factor. If $\Delta BIC(\Lambda)$ is less than zero, a change of speaker is claimed between shot 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 face close-up (human face with its size larger than 10% of the total frame size), and there should be 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 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 least one shot in SE_i contains blood-red or a close-up of a skin region (skin region with size larger than 20% of the total frame size) or where more than half of the shots in SE_i contain skin regions.

Based on the above definitions, 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:
   - If there is no slide or clip art frame contained in SE_i, go to step 3. If there is no face close-up contained in SE_i, go to step 3.
   - If all groups in SE_i consist of spatially related shots, go to step 3.
   - If there is any speaker change between adjacent shots of SE_i, go to step 3,
   - Assign the current group to the "Presentation" category; go to end or process other scenes.
3. Test whether SE_i belongs to "Dialog":
   - If there is either no face or no adjacent shots which both contain faces in SE_i, go to step 4.
   - If all groups in SE_i consist of spatially related shots, go to step 4.
   - If there is no speaker change between all adjacent shots which both contain faces, go to step 4.
   - 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":
   - If there is a speaker change between any adjacent shots, go to step 5.
   - If there is any skin close-up region or blood-red region detected, SE_i is assigned to "Clinical Operation".
   - 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 other scenes.

5. Scalable Video Skimming System

To utilize mined video content structure and events directly in video content access and to create useful applications, a scalable video skimming tool has been developed for visualizing the overview of the video and helping users access video content more effectively, as shown in Fig. 11. Currently, a four layer video skimming has been constructed, with level 4 through level 1 consisting of representative shots of clustered scenes, all scenes, all groups, and all shots, respectively. Note that 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 an interesting video unit.

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

To help the user visualize the mined events information within the video, a color bar is used to represent the content structure of the video so that scenes can be accessed efficiently by using event categorization. As shown in Fig. 11, the color of the bar for a given region indicates the event category to which the scene belongs. With this strategy, a user can access the video content directly.

In addition to the scalable video skimming, the mined video content structure and event categories can also facilitate more applications like hierarchical video browsing, pictorial summarization, etc.

Figure 11. Scalable video skimming tool

6. Algorithm Evaluation

In this section, we present the results of an extensive performance analysis we have conducted to: (1) evaluate the effectiveness of video scene detection and event mining, (2) analyze the performance of our cluster-based indexing framework, and (3) assess the acquired video content structure in addressing video content.

6.1 Video scene detection and event mining results

To illustrate the performance of the proposed strategies, two types of experimental results, video scene detection and event mining, are presented. Our dataset consists of approximately 6 hours of *MPEG-1* encoded medial videos which describe face repair, nuclear medicine, laparoscopy, skin examination, and laser eye surgery. Fig.12 and Fig.13 present 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. (20) is utilized for performance evaluation.

P= Rightly detected scenes / All detected scenes (20)

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. (21).

CRF=Detected scene number / Total shot number (21)

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. 12 and Fig. 13, 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 (best) 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), since our strategy consider more visual variance details among video scenes. It may cause one semantic unit be falsely segmented into several scenes. However, 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.

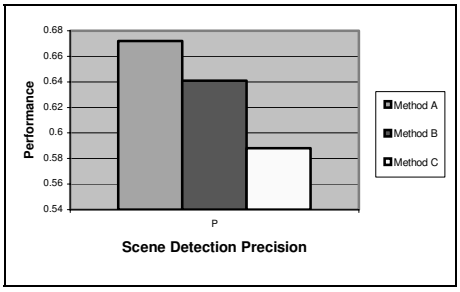

Figure 12. Scene detection performance

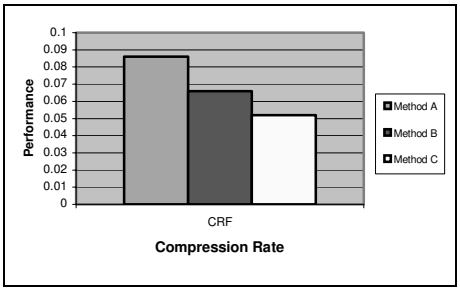

Figure 13. Compression Rate

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 benchmarks. 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. (22) and

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

Eq.(23) respectively. On average, our system achieves relatively good performance (72% in precision and 71% in recall) when mining these three types of events.

$$PR = True\ Number / Detected\ Number \quad (22)$$

$$RE = True\ Number / Selected\ Number \quad (23)$$

Table 1. Video event mining results where *SN*, *DN* and *TN* denote selected number, detected number and true number respectively

Events	*SN*	*DN*	*TN*	*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*

6.2 Cluster-based indexing analysis

The search time T_e for retrieving video from a large-scale database is the sum of two times: (*a*) time T_s for comparing the relevant videos in the database; (*b*) time T_r for ranking the relevant results. If no database indexing structure is used for organizing this search procedure, the total retrieval time is:

$$T_e = T_s + T_r = N_T \cdot T_m + O(N_T \log N_T) \quad (24)$$

where N_T is the number of videos in the database, T_m is the basic time to calculate the *m*-dimensional feature-based similarity distance between two video shots, and $O(N_T \log N_T)$ is the time to rank N_T elements.

Our cluster-based multi-level video indexing structure can provide fast retrieval because only the relevant database management units are compared with the query example. Moreover, the dimension reduction techniques can be utilized to guarantee that only the discriminating features are selected for video representation and indexing, and thus the basic time for calculating the feature-based similarity distance is also reduced ($T_c, T_{sc}, T_s, T_o \le T_m$ because only the discriminating feature are used). The total retrieval time for our cluster-based indexing system is

$$T_c = M_c \cdot T_c + M_{sc} \cdot T_c + M_s \cdot T_s + M_o \cdot T_o + O(M_o \log M_o) \quad (25)$$

where M_c, M_{sc}, M_s are the numbers of the nodes at the cluster level, the most relevant subcluster and the scene levels, respectively, M_o is the number of video shots that reside in the most relevant scene node, T_c, T_{sc}, T_s, T_o are the basic times for calculating the similarity distances in the corresponding feature subspace, and $O(M_o \log M_o)$ is the total time for ranking the relevant shots residing in the corresponding scene node. Since $(M_c + M_{sc} + M_s + M_o) << N_T$, $(T_c, T_{sc}, T_s, T_o) \le T_m$, thus $T_c << T_e$.

6.3 Scalable skimming and summarization results

Based on the mined video content structure and events information, a scalable video skimming and summarization tool was developed to present at most 4 levels of video skimming and summaries. To evaluate the efficacy of such a tool in addressing video content, three questions are introduced to evaluate the quality of the video skimming at each layer: (1) How well do you think the summary addresses the main topic of the video? (2) How well do you think the summary covers the scenarios of the video? (3) Is the summary concise? For each of the questions, a score from 0 to 5 (5 indicates best) is specified by five student viewers after viewing the video summary at each level. Before the evaluation, viewers are asked to browse the entire video to get an overview of the video content. An average score for each level is computed from the students' scores (shown in Fig.14). From Fig.14, we see that as we move to the lower levels, the ability of the skimming to cover the main topic and the scenario of the video is greater. The conciseness of the summary is worst at the lowest level, since as the level decreases, more redundant shots are shown in the skimming. At the highest level, the video summary cannot describe the video scenarios, but can supply the user with a concise summary and relatively clear topic information. Hence, this level can be used to show differences between videos in the database. It was also found that the third level acquires relatively optimal scores for all three questions. Thus, this layer is the most suitable for giving the user an overview of the video selected from the database for the first time.

A second evaluation process used the ratio between the numbers of frames at each skimming layer and the number of all frames (Frame Compression Ratio *FCR*) to indicate the compression rate of the video skimming. Fig.15 shows the results of *FCR* at various skimming layers. It can be seen that at the highest layer (layer 4) of the video skimming, a 10% compression rate has been acquired. This shows that by using the results of video content structure mining, an efficient compression rate can be obtained in addressing the video content for summarization, indexing, management etc.

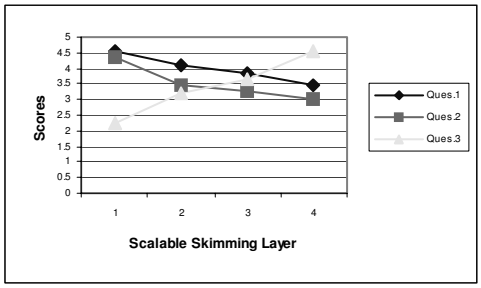

Figure 14. Scalable video skimming and summarization results

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE

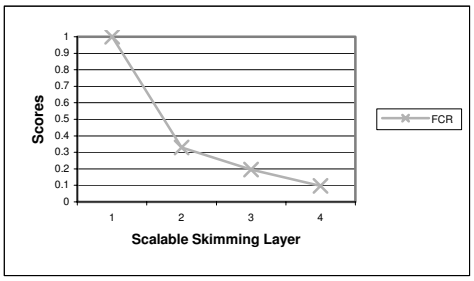

Figure 15. Frame compression ratio

7. Conclusion

In this paper, we have addressed video mining techniques for efficient video database indexing, management and access. To achieve this goal, a video database management framework is proposed by considering the problems of applying existing data mining tools among video data. A video content structure mining strategy is introduced for parsing the video shots into a hierarchical structure using shots, groups, scenes, and clustered scenes by applying shot grouping and clustering strategies. Both visual and audio processing techniques are utilized to extract the semantic cues within each scene. A video event mining algorithm is then adopted, which integrates 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 c ontent access prototype system is constructed to help the user visualize the overview and access video content more efficiently. Experimental results demonstrate the efficiency of our framework and strategies for video database management and access.

8. References

[1] R. Agrawal, T. Imeielinski, and A. Swami, "Data mining: A performance perspective", *IEEE TKDE, 5(6), 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, "VideoCube: A new tool for video mining and classification", *ICADL Dec., 2002, Singapore*.

[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, X. Zhu, "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 Multimedia system journal*, *7(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. Fan, X. Zhu, L. 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,vol.2, pp.113-116, 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*.

[24] X. Zhu, J. Fan, W.G. Aref, A.K. Elmagarmid, "ClassMiner: mining medical video content structure and events towards efficient access and scalable skimming", *Proc. of ACM SIGMOD workshop on DMKD, pp.9-16, WI, 2002.*

[25] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller, "Introduction to WordNet: An on-line lexical database", *Journal of Lexicography, V.3, pp.235-244, 1990.*

[26] C. Faloutsos, W. Equitz, M. Flickner, W. Niblack. S. Petkovic, and R. Barber, "Wfficient and effective querying by image content", *Journal of Intelligent Information System, vol. 3, pp.231-262, 1994.*

[27] J.-Y. Pan and C. Faloutsos, "Geoplot: Spatial Data Mining on Video Libraries", *CIKM, Nov., VA, 2002.*

Proceedings of the 19th International Conference on Data Engineering (ICDE'03)
1063-6382/03 $ 17.00 © 2003 IEEE