

# Hierarchical video content description and summarization using unified semantic and visual similarity

Xingquan Zhu<sup>1</sup>, Jianping Fan<sup>2</sup>, Ahmed K. Elmagarmid<sup>3</sup>, Xindong Wu<sup>1</sup>

<sup>1</sup> Department of Computer Science, University of Vermont, Burlington, VT 05401, USA

<sup>2</sup> Department of Computer Science, University of North Carolina, Charlotte, NC 28223, USA

<sup>3</sup> Department of Computer Science, Purdue University, West Lafayette, IN 47907, USA

**Abstract.** Video is increasingly the medium of choice for a variety of communication channels, resulting primarily from increased levels of networked multimedia systems. One way to keep our heads above the video sea is to provide summaries in a more tractable format. Many existing approaches are limited to exploring important low-level feature related units for summarization. Unfortunately, the semantics, content and structure of the video do not correspond to low-level features directly, even with closed-captions, scene detection, and audio signal processing. The drawbacks of existing methods are the following: (1) instead of unfolding semantics and structures within the video, low-level units usually address only the details, and (2) any important unit selection strategy based on low-level features cannot be applied to general videos. Providing users with an overview of the video content at various levels of summarization is essential for more efficient database retrieval and browsing. In this paper, we present a hierarchical video content description and summarization strategy supported by a novel joint semantic and visual similarity strategy. To describe the video content efficiently and accurately, a video content description ontology is adopted. Various video processing techniques are then utilized to construct a semi-automatic video annotation framework. By integrating acquired content description data, a hierarchical video content structure is constructed with group merging and clustering. Finally, a four layer video summary with different granularities is assembled to assist users in unfolding the video content in a progressive way. Experiments on real-world videos have validated the effectiveness of the proposed approach.

**Key words:** Hierarchical video summarization – Content description – Semi-automatic video annotation – Video grouping

## 1. Introduction

Recent years have seen a rapid increase in the use of multimedia information. Of all media types, video is the most challenging, as it combines all other media information into

a single data stream. Owing to the decreased cost of storage devices, higher transmission rates, and improved compression techniques, digital videos are becoming available at an ever-increasing rate. However, the manner in which the video content is presented for access such as browsing and retrieval has become a challenging task, both for application systems and for viewers. Some approaches have been described elsewhere [1–3], which present the visual content of the video in different ways, such as hierarchical browsing, storyboard posting, etc. The viewer can quickly browse through a video sequence, navigate from one segment to another to rapidly get an overview of the video content, and zoom to different levels of detail to locate segments of interest.

Research in the literature [3] has shown that, on average, there are about 200 shots for a 30-minute video clip across different program types, such as news and drama. Assuming that a key-frame is selected to represent each shot, 200 frames will impose a significant burden in terms of bandwidth and time. Using spatially reduced images, commonly known as thumbnail images, can reduce the size further, but may still be expensive if all shots must be shown for a quick browse of the content. Hence, a video summarization strategy is necessary to present viewers with a compact digest that shows only parts of video shots.

Generally, a video summary is defined as a sequence of still or moving pictures (with or without audio) presenting the content of a video in such a way that the respective target group is rapidly provided with concise information about the content, while the essential message of the original is preserved [4]. Three kinds of video summary styles are commonly used:

- A *pictorial summary* [3, 5–11] is a collection of still images (icon images, even varying in size) arranged in time order to convey the highlights of the video content.
- A *video skimming* [4, 12–15] is a collection of moving frames (video shots) arranged in time series to convey the main topics in the video, i.e. it is a trimmed video.
- A *data distribution map* [13, 15] is a picture to illustrate the distribution of some specific data in the database.

Obviously, a video summary is the most appealing in video browsing. By supplying a compact digest, the user can browse the video content quickly and comprehensively. Moreover, the power of visual summary can be helpful in many applications,

such as multimedia archives, video retrieval, home entertainment, digital magazines, etc. More and more video material is being digitized and archived worldwide. Wherever digital video material is stored, a duplicated summary could be stored at any node on the Internet, and the user's query could be processed only at these nodes to work out a rough query result. In this way, the system could release a substantial amount of CPU time and bandwidth to process more queries. Moreover, since the abstract is always far shorter than the original data, each user's query time is reduced significantly. Furthermore, if the library grew to thousands of hours, queries could return hundreds of segments. Generating a summary of the query results would allow users to browse the entire result space without having to resort to the time-consuming and frustrating traversal of a large list of segments.

Nevertheless, without a comprehensive understanding of the video content, the generated video summary would be unsatisfactory for most users. Consequently, various video indexing strategies have been proposed to describe the video content manually, semi-automatically or fully automatically. Based on different types of knowledge utilized, the video indexing can be distinguished into the following three categories:

- *High-level indexing*: this approach uses a set of predefined index terms to annotate videos. The index terms are organized based on high-level ontological categories like action, time, space, etc.
- *Low-level indexing*: these techniques provide access to videos based on low-level features such as color, texture, audio, and closed-captions. The driving force behind these techniques is to extract data features from the video data, organize the features based on some distance measures, and use similarity-based matching to retrieve the video.
- *Domain Specific Indexing*: these techniques use the high-level structure of video to constrain low-level video feature extraction and processing. However, they are effective only in their intended domain of application.

Based on the content description acquired from video indexing, various kinds of applications can be implemented [16–18].

The rest of the paper is organized as follows. In the next section, related work on video annotating and summarization is reviewed. The overall system architecture of our proposed method is described in Sect. 3. In Sect. 4, a video content description ontology is proposed. Based on this ontology, the video content annotation scheme is presented in Sect. 5. Our hierarchical video summarization scheme is introduced in Sect. 6. In Sect. 7, the effectiveness of the proposed approach is validated by experiments over real-world movie video clips, and some potential application domains of the proposed strategies are outlined. Concluding remarks are given in Sect. 8.

## 2. Related work

Video annotation and indexing issues have been addressed with various approaches. Due to the inadequacy of textual terms in describing the video content, a textual based video annotation has led to considerable loss of information. Accordingly, many low-level indexing strategies have emerged [13, 15, 19–22] which use closed-captions, audio information,

speech recognition, etc. to explore video content. Some video classification methods have also been developed to detect the event or topic information within the video [23–26], but they are only effective in their own specific domain; moreover, only a fraction of events can be detected. We are currently able to automatically analyze shot breaks, pauses in audio, and camera pans and zooms, yet this information alone does not enable the creation of a sufficiently detailed representation of the video content to support content-based retrieval and other tasks. As their experiments have shown, there is still a long way to go before we can use these methods to acquire satisfactory results. Hence, manual annotation is still widely used to describe video content.

The simplest way to model video content is using free text to manually annotate each detected shot separately. However, since a segmented part of the video is separated from its context, the video scenario information is lost. To address this problem, Aguiere Smith et al. [27] implement a video annotation system using the concept of stratification to assign description to video footage, where each stratum refers to a sequence of video frames. The strata may overlap or totally encompass each other. Based on this annotation scheme, the *video algebra* [28] was developed to provide operations for the composition, search, navigation and playback of digital video presentation. A similar strategy for evolving documentary presentation could be found in [29]. Instead of using pure textual terms for annotation, Davis et al. [30] present an iconic visual language-based video annotation system, Media Stream, which enables users to create multi-layered, iconic annotations of video content; however, this user-friendly visual approach to annotation is limited by a fixed vocabulary. An overview of research in this area could be found elsewhere [31].

Each of the manual annotation strategies identified above may be efficient in addressing specific issues, but problems still remain:

1. Even though the hierarchical strategy (stratification) has been accepted as an efficient way for video content description, no efficient tool has been developed to integrate video processing techniques (video shot and group detection, joint semantic and visual features in similarity evaluation, etc.) for semi-automatic annotation to free annotators from sequentially browsing and annotating video frame by frame.
2. The keywords at different content levels have different importance and granularity in describing video content; hence, they should be organized and addressed differently.
3. To minimize the subjectivity of the annotator and the influence of wide spread synonymy and polysemy in unstructured keywords, one efficient way is to utilize content description ontologies. The existing methods either fail to define their ontology explicitly (using free text annotation [33]) or do not separate the ontology with annotation data to enhance the reusability of the annotation data.

To address these problems, a semi-automatic video annotation scheme is proposed. We first define the content description ontology, as shown in Fig. 2. Then various video processing techniques are introduced to construct a semi-automatic video annotation strategy.

Even without a comprehensive understanding of video content, many low-level tools have been developed to gen-

erate video summaries by icon frames [3, 6–9, 11] or video objects [5, 10]. A curve simplification strategy is introduced in [6] for video summarization, which maps each frame into a vector of high dimensional features, and segments the feature curve into units. A video summary is extracted according to the relationship among them. In video Managa [7], a pictorial video summary is presented with key frames in various sizes, where the importance of the key frame determines its size. To find the highlight units for video abstracting, Nam et al. [11] present a method that applies different sampling rates on videos to generate the summary. The sample rate is controlled by the motion information in the shot. Instead of summarizing general videos, some abstracting strategies have been developed to deal with videos in a specific domain, such as home videos [12], stereoscopic videos [5], and online presentation videos [8]. The general rules and knowledge followed by the video are used to analyze the semantics of the video. On comparing with abstracting the video with pictorial images, some approaches summarize a video by skimming [13, 15], which trims the original video into a short and highlight stream. Video skimming may be useful for some purposes, since a tailored video stream is appealing for users. However, the amount of time required for viewing a skimming suggests that skimmed video is not appropriate for a quick overview, especially for network-based applications where bandwidth is the most of concern. Neither pictorial abstracts nor skimming has the greater value, and both are supported by the strategy presented in this paper.

Ideally, a video summary should briefly and concisely present the content of the input video source. It should be shorter than the original, focus on the content, and give the viewer an appropriate overview of the whole. However, the problem is that what's *appropriate* varies from viewer to viewer, depending on the viewer's familiarity with the source and genre, and with the viewer's particular goal in watching the summary. A hierarchical video summary strategy [9, 34] is proposed accordingly by supplying various levels of summarization to assist the viewer in determining what is *appropriate*. In [9], a key frame based hierarchical summarization strategy is presented, where key frames are organized in a hierarchical manner from coarse to fine temporal resolution using a pairwise clustering method. Instead of letting a user accept the generated video summary passively, *movieDNA* [34] supplies the user with a hierarchical, visualized, and interactive video feature map (summary) called *DNA*. By rolling the mouse over the *DNA*, users can brush through the video, pulling up detailed meta-information on each segment.

In general, the aforementioned methods all work with nearly the same strategy: grouping videos, selecting low-level features related important units, acquiring users' specification of summary length, assembling. Unfortunately, there are three problems with this strategy: First, it relies on low-level features to evaluate the importance of each unit. But selected highlight may not be able to cover important semantics within the video, since there is no general linkage between low-level feature and semantics. Second, important unit selection is a semantic-related topic. Different users have different value judgments, and it would be relatively difficult to determine how much more important one unit is than the others. Third, the length of the users' specifications for the summary is not always reasonable in unfolding video content, especially if the

user is unfamiliar with videos in the database. As a result, those strategies just present a "sample" of the video data. Hence, we need an efficient technique to describe, manage and present the video content, without merely sampling the video.

Video is a structured media. While browsing video, it is not the sequential frames but the hierarchical structure (video, scenes, group, etc.) behind frames that convey scenario and content information to us. Hence, the video summary should also fit this hierarchical structure by presenting an overview of the video at various levels. Based on this idea and acquired video content description data, this paper presents a hierarchical video summarization scheme that constructs a four layer summary to express video content.

### 3. System architecture

Figure 1 presents the system architecture of the strategy proposed in this paper. It consists of two relatively independent parts: *Hierarchical video content description* and *hierarchical video summarization*.

First, to acquire video content, the content description ontology is proposed (as described in Sect. 4). Then, all shots in the video are parsed into groups automatically. A semi-automatic video annotation is proposed which provides a friendly user interface to assist the system annotator in navigating and acquiring video context information for annotation. A video scene detection strategy using joint visual features and semantics is also proposed to help annotators visualize and refine annotation results.

After the video content description stream has been acquired, it is combined with the visual features to generate a hierarchical video summary. By integrating semantics and low-level features, the video content is organized into a four level hierarchy (video, scene, group, shot) with group merging and clustering algorithms. To present and visualize the video content for summarization, various strategies have also been proposed to select the representative group, representative shots, and representative frames for each unit. Finally, a four layer video summary is constructed to express the video digest in various layers, from top to bottom in increasing granularity.

### 4. Video content description architecture

To enable search and retrieval of video for large archives, we need a good description of video content. Due to the fact that different users may have various perceptions of the same image or video, and moreover, the wide spread synonymy and polysemy in natural language may cause annotators to use different keywords to describe the same object. The ontology based knowledge management is utilized for annotation: We first define the video content description ontology, as shown in Fig. 2. Then a shot based data structure is proposed to describe video content and separate ontology from annotation data. One of the main originalities of our approach is that it allows dynamic and flexible video content description with various granularities where the annotations are independently from the video data.

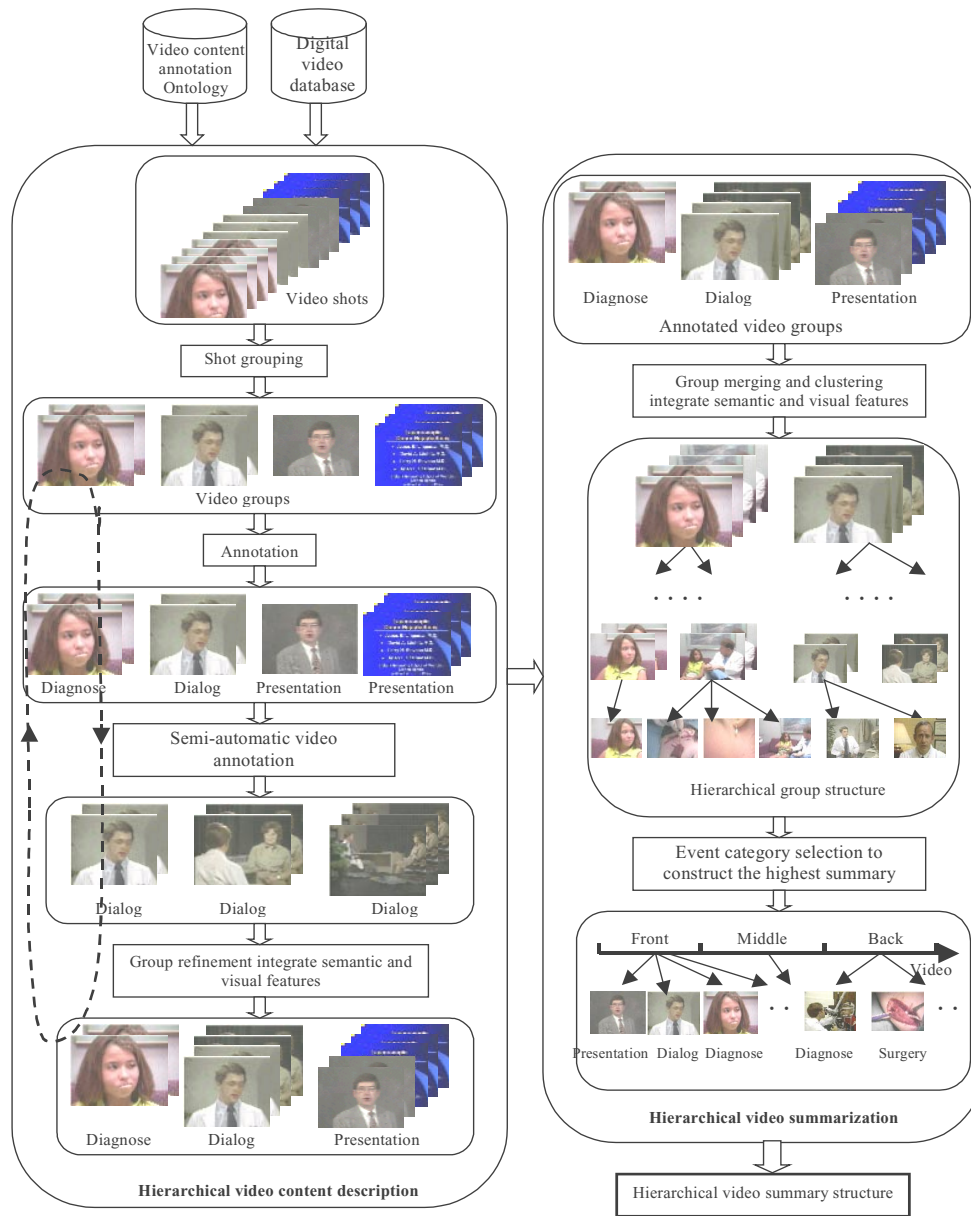


Fig. 1. System architecture

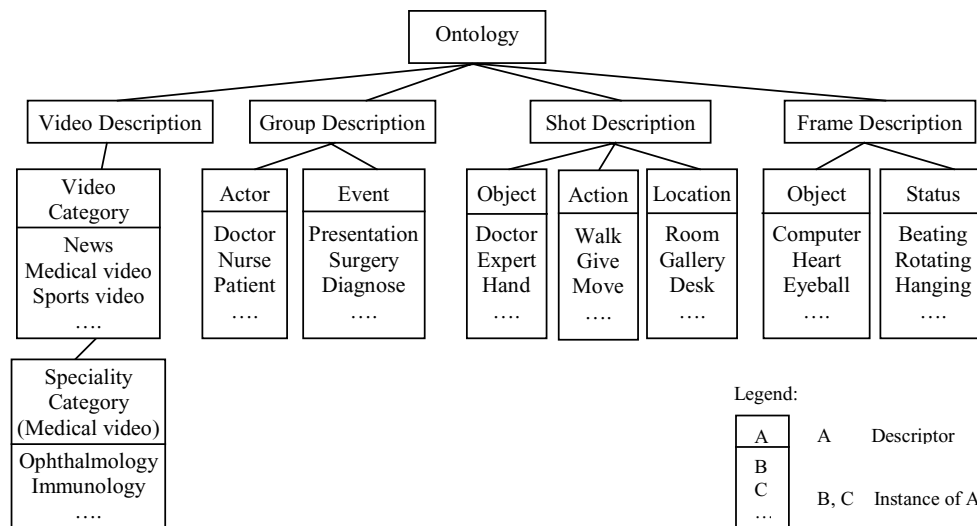


Fig. 2. Video content description ontology

#### 4.1. Ontology-based knowledge management

In recent years the development of ontologies has been moving from the realm of Artificial Intelligence laboratories to the desktops of domain experts. Some successful applications of ontologies have been implemented on the web [47] range from large taxonomies categorizing websites (e.g. Yahoo (www.yahoo.com)) to categorizations of products for sale and their features (e.g. Amazon.com (www.amazon.com)). However, the research of ontology-based video content annotation is rarely addressed.

Generally, the ontology is defined as an explicit and formal specification of a shared conceptualization of knowledge, it provides a suitable format and a common-shared terminology for the description of the content of knowledge sources. Typical ontologies consist of definitions of concepts relevant for the domain, their relations, and axioms about these concepts and relationships. By using a given domain ontology one can annotate content of provided knowledge source in such a way that a knowledge-seeker can find the knowledge source easily, independently of its representation format. Hence, the role of ontologies is twofold: (1) they support human understanding and organizational communication; and (2) they are machine processible, and thus facilitate content-based access, communication and integration across different information system.

An ontology can be constructed in two ways: domain dependent and generic. Generic ontologies (e.g. WordNet [36]) are constructed to make a general framework for all (most) categories encountered by human existence; they are usually very large but not very detailed. For our purposes, to describe video content, we are interested in creating domain dependent ontologies which are generally much smaller. During the construction of ontologies, the features below are taken into consideration to guarantee the quality of the ontology.

- Open and dynamic: ontologies should have fluid boundaries and be readily capable of growth and modification.
- Scalable and inter-operable: an ontology should be easily scaled to a wider domain and adapts itself to new requirements.
- Easily maintained: it should be easy to keep ontologies up-to-date. Ontologies should have a simple, clear structure, as well as be modular. It should also be easy for humans to inspect.

Accordingly, the domain experts or system managers, someone who has mastery over the specific content of a domain [37], should be involved to create and maintain the ontologies by considering the general steps below:

1. Determine the domain and scope of the ontology.
2. Consider reusing existing ontologies.
3. Enumerate important terms in the ontology.
4. Explicitly specify the definition of each class and indicate the class hierarchy.
5. Create the instances of each class.

Some existing schemes and researches have addressed the problems on creating and maintaining the ontologies, the details could be found in [48]. After the ontology has been created, it could then be utilized for knowledge management and description.

#### 4.2. Video content description ontology

For most applications, the entire video document is at too coarse a level of content description [38, 39]. A single frame, on the other hand, is rarely the unit of interest. This is because a single frame spans a very short interval of time and there are too many individual frames, even in a short video document. As we know, most videos from daily life can be represented by using a hierarchy consisting of five layers (video, scene, group, shots, frames), from top to bottom in increasing granularity for content expression [40]. Consequently, a robust and flexible video content description strategy should also be able to describe video content at different layers and with different granularity.

To construct our video content description ontology, we first clarify that our domain are general video data from our daily life. Then, the general structure information among the videos is considered to construct ontology for video data. When looking at a video, what kinds of things do we want to state about it? From most annotators' viewpoint, the annotation should answer five "W" related questions, who? what? when? where? and why? Obviously, at different video content levels, the annotations should have different granularities in addressing these five questions. Hence, a video content description ontology is proposed in Fig. 2, where four content descriptions, video description (*VD*), group description (*GD*), shot description (*SD*) and frame description (*FD*), are adopted with each description defined as below (we eliminate the scene level content description in the ontology, since video scenes depict and convey a high-level concept or story, without semantics, it cannot be detected satisfactorily):

1. The video description addresses the category and specialty taxonomy information of the entire video. The description at this level should answer questions like, "What does the video talk about?"
2. The group description describes the event information in a group of adjacent shots that convey the same semantic information. The description at this level should answer the queries like, "give me all surgery units among the medical videos?"
3. The shot description describes the action in the single shot. This action could be a part of an event. For example, a video shot could show the action of a doctor shaking hands with a patient in a diagnosis event. Hence, the shot description should answer queries like, "give me all units where a doctor touches the head of the patient".
4. At the lowest level, the frame, the description should address the details of objects in a single frame (or series of adjacent frames.) The description should answer queries like "what is in the frame(s)?"

The proposed descriptions are then assembled together to construct the framework of the ontology, as shown in Fig. 2. To present and address more details among descriptions, various *descriptors* are introduced for each description, as shown in Table 1. Obviously, instead of considering the domain information of each video type, the proposed ontology could be utilized to describe most videos in our daily life. However, for certain kinds of videos (e.g. the medical video), the knowledge from domain experts would be necessary for content management and annotation. Hence, we adopt many extendable *descriptors* in the ontology, which are specified by the

**Table 1.** Definition of descriptors in video content description ontology

Description	Descriptor	Definition
VD	Video category	Specify video category information
	Speciality Category	Specify taxonomy information in a specific video domain
GD	Event	Specify the event information in current group
	Actor	Specify the actor of the current event
SD	Object	Specify the object(s) in the shot
	Action	Specify the action of the object(s)
	Location	Specify the location of the action
FD	Object	Specify the object(s) in the current frame (or series of adjacent frames)
	Status	Specify the condition of the object(s)

domain expert or system manager, to address details of each video type.

To determine descriptors and their instances for any given video type, the domain expert is involved to list all possible interesting objects (or classes) among videos. The aggregation of all objects (or classes) will finally determine the descriptors and their instances. In our demonstration system [], the expert from medical school is invited to help us in constructing the ontology descriptors and their instances, as shown in Tables 1–4. In the case that the domain experts are not available, we may utilize the information from Internet, for example, Yahoo has a hierarchy of 1,500,000 categories which might be very helpful in creating an ontology.

The instances of each descriptor are predefined, as shown in Tables 2, 3, and 4. In Table 2, we first classify the video into various domains, such as News program, Medical Video etc. Then, given a specific video domain, the specialty category is used to classify the video into specific directories, as shown in Table 3, we classify medical videos into categories such as ophthalmology, immunology, etc. An example of the predefined event descriptor in the medical video is also given in Table 4. Moreover, the instances of each descriptor are still extensible. While annotating, the annotator may browse the instances of certain descriptor first; if there is no keyword suitable for the current description, one or more instances may be added. The advantage of using content description ontology is outlined below:

1. It supply a dynamic and efficient video content description scheme, the ontology could be extended easily by adding more descriptors.
2. The ontology can be separated from the annotation data. The annotator’s modification with the instances of each descriptor could be shared with other annotators. This will enhance the reusability of the description data.

Note that, since adjacent frames in one shot usually contain the same semantic content, the annotator may mask a group of adjacent frames as one unit, and annotate them in one time.

#### 4.3. Shot-based video temporal description data organization

To enhance the reusability of descriptive data, we should separate the ontology from the description data. That is, the description ontology is constructed, maintained and shared by

**Table 2.** An example of the instances for video category

Home video	Surveillance video	Presentation video
News program	Medical video	Movie
MTV program	Sports video	Animal program
Comedy	Course lesson	...

**Table 3.** An example of the instances for a specialty category of medical video

Ophthalmology	Immunology	Cardiology
Endocrinology	Radiobiology	Oncology
Organ diagram	Microbiology	...

**Table 4.** An example of the instances for event descriptor of medical video

Presentation	Dialog	Diagnose
Surgery	Experiment	Outdoor scene
Organ diagram	Unknown	...

all annotators. With this ontology, various description data could be acquired from different videos. To integrate video semantics with low-level features, a shot based data structure is constructed for each video, as shown in Fig. 3. Given any video shot  $S_i$ , assuming  $KA$  indicates the Keyword Aggregation ( $KA$ ) of all descriptors in the ontology, then  $KA = \{VD_l, l = 1, \dots, NV_i; GD_l, l = 1, \dots, NG_i; SD_l, l = 1, \dots, NS_i; FD_l, l = 1, \dots, NF_i\}$ , where  $VD_l$ ,  $GD_l$ ,  $SD_l$  and  $FD_l$  represent the keywords of the video description ( $VD$ ), group description ( $GD$ ), shot description ( $SD$ ) and frame description ( $FD$ ), respectively, and  $NV_i$ ,  $NG_i$ ,  $NS_i$  and  $NF_i$  indicate the number of keywords for each description. Moreover, to indicate the region where each keyword takes effect, the symbol  $v_{a-b}^{ID}$  is used to denote the region from frame  $a$  to frame  $b$  in the video with a certain identification ( $ID$ ). The temporal description data ( $TDD$ ) for video shot  $S_i$  is then defined as the aggregation of mappings between annotation ontology and temporal frames.

$$TDD = \{S_i^{ID}, S_i^{ST}, S_i^{ED}, Map(KA, V)\} \quad (1)$$

where  $S_i^{ID}$  specifies the identification ( $ID$ ) for current shot  $S_i$ .  $S_i^{ST}$  and  $S_i^{ED}$  denote the start and end frame of  $S_i$  respectively.  $KA$  indicates the keyword aggregation of all descriptors,  $V$  indicates a set of video streams,  $v_{a-b}^{ID} \in V, ID = 1, \dots, n$ ,

and  $Map()$  defines the correspondence between annotations and video temporal information.  $Map(KA_i; v_{a-b}^{ID})$  denotes the mapping between keyword  $KA_i$  to region from frame  $a$  to frame  $b$  in video with certain identification  $ID$ . For instance, the mapping  $Map(FD_l; v_{100-200}^2)$  defines a one-to-one mapping between a  $FD$  descriptor keyword  $FD_l$  and  $v_{100-200}^2$ , where the identification of the video is  $ID = 2$  and the frame region of the mapping is from frame 100 to frame 200. We can also have a many-to-one mapping. For example, the mapping  $Map(FD_k, SD_l; v_{100-300}^2)$  defines a many-to-one relationship to indicate both keywords  $FD_k$  and  $SD_l$  are specified in the region from frame 100 to 300 in video with  $ID = 2$ . Similarly, many-to-many and one-to-many relationship can be defined, as shown in Eq. 2:

$$\begin{aligned} &Map(FD_k, SD_l; v_{100,800}^2, v_{1200-1400}^2); \\ &Map(GD_k; v_{200-700}^2, v_{1300-2300}^2) \end{aligned} \quad (2)$$

The advantage of the above mapping is that we have separated the annotation ontology from the temporal description data. Hence, the same video data can be shared and annotated by different annotators for different purposes, and can be easily reused for different applications.

Given one video, the assembling of the  $TDD$  of all shots contained will form its temporal description stream ( $TDS$ ). This indicates that all annotation information is associated to each shot, with each shot containing  $VD$ ,  $GD$ ,  $SD$  and  $FD$  information. The reason we utilize such a data structure is clarified below:

1. A data description structure based on the single frame level will inevitably incur large redundancy.
2. We can segment video shot boundaries automatically [42, 43] and with a satisfactory result.
3. Video shots are usually taken as the basic units of the video processing techniques [1, 2], a shot based structure will help us seamlessly integrate low-level features with semantics.
4. If there is large content variance in the shot, more keywords can be used in Frame Description to characterize the changing. Hence, the proposed structure will not lose semantic details of the video.

## 5. Video content annotation

Using the video content description ontology described in Sect. 4, the video content can be stored, browsed, and retrieved efficiently. But no matter how effective an annotation structure is, annotating videos frame by frame is still a time consuming operation. In this section, a semi-automatic annotation strategy which utilizes various video processing techniques (shot segmentation, group detection, group merging, etc.) is proposed to help annotators acquire video context information for annotation. We will first address techniques on parsing video into shots and semantically related groups. Then, a video scene detection strategy is proposed. Finally, by integrating these techniques, a semi-automatic video annotation scheme is presented.

### 5.1. Video group detection

The simplest method to parse video data for efficient browsing, retrieval and navigation, is segmenting the continuous video sequence into physical shots and then selecting one or more key-frame for each shot to depict its content information [1, 2]. We use the same approach in our strategy. Video shots are first detected from a video using our shot detection techniques [43]. For the sake of simplicity, we choose the 10<sup>th</sup> frame of each shot as its key-frame. However, since the video shot is a physical unit, it is incapable in conveying independent semantic information. Various approaches are proposed to determine a cluster of video shots (group or scene \*) that convey relatively higher level video scenario information. Zhong et al. [21] propose a strategy, which clusters visually similar shots and supplies viewers with a hierarchical structure for browsing. However, since spatial shot clustering strategies consider only the visual similarity among shots, the context information is lost. To address this problem, Rui et al. [40] present 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 [32]. In [44], 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 [45]. Nevertheless, the video scene is a semantic unit, it is difficult in some situations to determine boundaries even with the human eye by using only visual features. Hence, the scene segmentation results with current strategies are unsatisfactory. Compared to other strategies that emphasize grouping all semantically related shots into one scene, our method emphasizes merging those temporally or spatially related shots into groups, and then offering those groups for annotations.

The quality of most proposed methods heavily dependent on the selection of thresholds [32, 44, 45]; however, the content and low-level features among different videos are varied. Even in the same video, there may be a large variance. Hence, an adaptive threshold selection for video grouping or scene segmentation is necessary. We use the entropic threshold technique in this paper. It has been shown to be highly efficient for the two-class data classification problem.

#### 5.1.1. Shot grouping for group detection

Video shots in the same scene have a higher probability of sharing the same background, they may have higher visual similarities when compared with other shots which are not in the same scene. Moreover, shots in the same scene may also be organized in a temporal sequence to convey scenario information. For example, in a dialog scene the adjacent shots usually have relatively low similarity, however, similar shots might be shown back and forth to characterize different actors in the dialog. To address the correlation among shots in the same scene, a shot grouping strategy is proposed in this section to merge semantically related adjacent shots into group(s).

To segment spatially or temporally related shots into groups, a given shot is compared with the shots that precede and succeed it (using no more than two shots) to determine

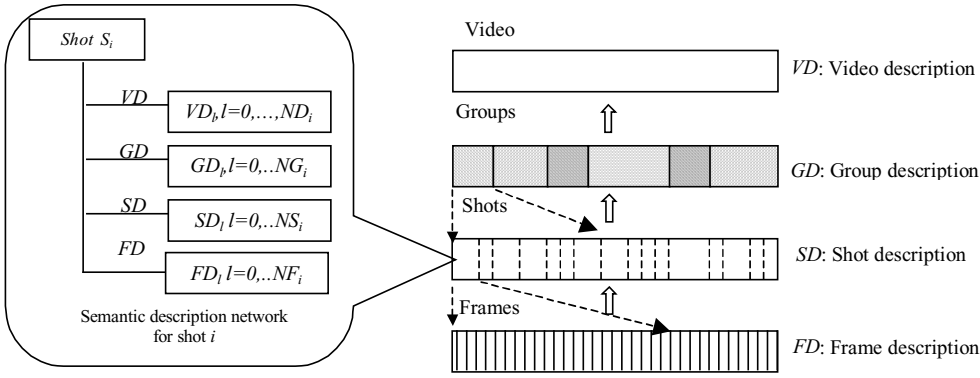


Fig. 3. Data structure for shot based video temporal description

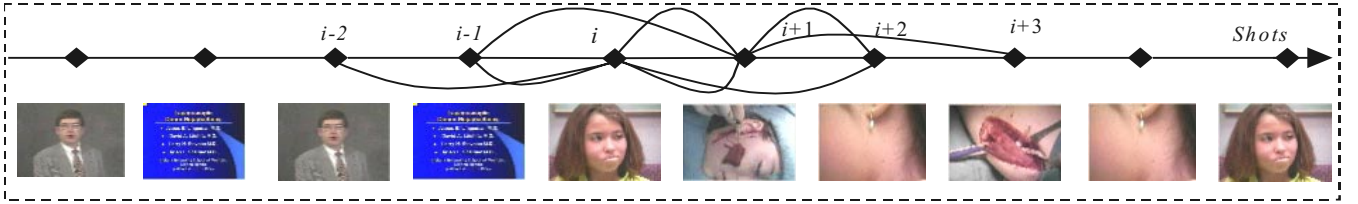


Fig. 4. Exploring correlations among video shots for group detection

correlations between them, as shown in Fig. 4. Since closed-caption and speech information is not available in our strategy, we use visual features to determine the similarity between shots. We adopt a 256 dimensional HSV color histogram and 10 dimensional tamura coarsness texture as visual features. Suppose  $H_{i,l}$ ,  $l \in [0, 255]$  and  $T_{i,l}$ ,  $l \in [0, 9]$  are the normalized color histogram and texture of the key frame  $i$ . The visual similarity between shot  $S_i$  and  $S_j$  is defined by Eq. 3:

$$StSim(S_i, S_j) = W_C \sum_{l=0}^{255} \min(H_{i,l}, H_{j,l}) + W_T \left(1 - \sqrt{\sum_{l=0}^9 (T_{i,l} - T_{j,l})^2}\right) \quad (3)$$

where  $W_C$  and  $W_T$  indicate the weight of color and tamura texture. For our system, we set  $W_C = 0.7$ ,  $W_T = 0.3$ .

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

$$CL_i = \text{Max}\{StSim(S_i, S_{i-1}), StSim(S_i, S_{i-2})\} \quad (4)$$

$$CR_i = \text{Max}\{StSim(S_i, S_{i+1}), StSim(S_i, S_{i+2})\} \quad (5)$$

$$CL_{i+1} = \text{Max}\{StSim(S_{i+1}, S_{i-1}), StSim(S_{i+1}, S_{i-2})\} \quad (6)$$

$$CR_{i+1} = \text{Max}\{StSim(S_{i+1}, S_{i+2}), StSim(S_{i+1}, S_{i+3})\} \quad (7)$$

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. 8 to evaluate a potential group boundary.

$$R(i) = (CR_i + CR_{i+1}) / (CL_i + CL_{i+1}) \quad (8)$$

The shot group detection procedure then takes the following steps:

1. Given any shot  $S_i$ , if  $CR_i$  is larger than  $TH_2 - 0.1$ :
  - (a) If  $R(i)$  is larger than  $TH_1$ , 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  $TH_2$ , 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.

Using this strategy, two kinds of shots are absorbed into a given group:

- Shots related in temporal series, such as a dialog or presentation, where similar shots are shown back and forth. Shots in this group are referred to as *temporally related*. Examples of temporally related shots are shown as row 1 and 2 in Fig. 5, where adjacent shots have relatively low similarity, however the similar shots are interlaced to be shown in one group.
- Shots related in visual similarities, where all shots in the group are visually similar. Shots in this group are referred



**Fig. 5.** Group detection results, with group rows from the top to bottom identifying (in order): presentation, dialog, surgery, diagnosis and diagnosis

to as *spatially related*. Examples of spatially related shots are shown as row 3 in Fig. 5, where adjacent shots are almost similar with each other.

Accordingly, given detected group  $G_i$ , we will assign it to one of two categories: temporally vs spatially related group. Assuming there are  $W$  shots ( $S_i, i = 1, \dots, W$ ) contained in  $G_i$ , the group classification strategy takes following steps.

**Input:** Video group  $G_i$ , and shots  $S_i$  ( $i = 1, \dots, W$ ) in  $G_i$ .

**Output:** Clusters ( $C_{N_c}, N_c = 1, \dots, U$ ) 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 for 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  $C_{N_c}$ . 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.

*Remark:* In this paper, the video group and scene are defined as similar as in [40]: (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. A video scene usually consists of one or more video groups.

### 5.1.2. Automatic threshold detection

As stated in the above section, the thresholds  $TH_1, TH_2$ , are the key values for obtaining good results. An entropic threshold technique is used in this section to select the optimal thresholds for these two factors. A fast entropy calculation method is also presented. To illustrate, assume the maximal difference of  $R(i)$

in Eq. 8 is in the range  $[0, M]$ . In an input *MPEG* video, assume there are  $f_i$  shots whose  $R(i)$  has the value  $i, i \in [0, M]$ . Given a threshold, say  $T$ , the probability distribution for the *group-boundary* and *non-group-boundary* shots can be defined. As they are to be regarded as the independent distributions, the probability for the non-group-boundary  $P_n(i)$  shots can be defined as:

$$P_n(i) = f_i / \sum_{h=0}^T f_h, \quad 0 \leq i \leq T \quad (9)$$

where  $\sum_{h=0}^T f_h$  gives the total number of shots with ratio  $R(i)$  in range  $0 \leq i \leq T$ . The probability for the group-boundary shots  $P_e(i)$  can be defined as:

$$P_e(i) = f_i / \sum_{h=T+1}^M f_h, \quad T+1 \leq i \leq M \quad (10)$$

$\sum_{h=T+1}^M f_h$  is the total number of shots with ratio  $R(i)$  in the range  $T+1 \leq i \leq M$ . The entropies for these two classes, group boundary shot and non-group-boundary shot are then given by:

$$\begin{aligned} H_n(T) &= - \sum_{i=0}^T P_n(i) \log P_n(i); \\ H_e(T) &= - \sum_{i=T+1}^M P_e(i) \log P_e(i) \end{aligned} \quad (11)$$

The optimal threshold vector  $T_C$  has to satisfy the following criterion function [46]:

$$H(T_c) = \max_{T=0 \dots M} \{H_n(T) + H_e(T)\} \quad (12)$$

To find the global maximum of Eq. 12, the computation burden is bounded by  $O(M^2)$ . To reduce the search burden, a fast search algorithm is proposed to exploit the recursive iterations for calculating the probabilities  $P_n(i), P_e(i)$

and the entropies  $H_n(T)$ ,  $H_e(T)$ , where the computational burden is introduced by calculating the re-normalized part repeatedly. We first define the total number of the pairs in the non-group-boundary and group-boundary classes (the re-normalized parts used in Eq. 9 and 16) when the threshold is set to  $T$ :

$$P_0(T) = \sum_{h=0}^T f_h ; \quad P_1(T) = \sum_{h=T+1}^M f_h \quad (13)$$

The corresponding total number of pairs at global threshold  $T + 1$  can be calculated as:

$$\begin{aligned} P_0(T+1) &= \sum_{h=0}^{T+1} f_h = \sum_{h=0}^T f_h + f_{T+1} = P_0(T) + f_{T+1} \\ P_1(T+1) &= \sum_{h=T+2}^M f_h = \sum_{h=T+1}^M f_h - f_{T+1} = P_1(T) - f_{T+1} \end{aligned} \quad (14)$$

The recursive iteration property of the two corresponding entropies can then be exploited by Eq. 15

$$\begin{aligned} H_n(T+1) &= - \sum_{i=0}^{T+1} \frac{f_i}{P_0(T+1)} \log \frac{f_i}{P_0(T+1)} \\ &= - \frac{P_0(T)}{P_0(T+1)} \sum_{i=0}^{T+1} \frac{f_i}{P_0(T)} \log \left\{ \frac{f_i}{P_0(T)} \frac{P_0(T)}{P_0(T+1)} \right\} \\ &= \frac{P_0(T)}{P_0(T+1)} H_n(T) - \frac{f_{T+1}}{P_0(T+1)} \log \frac{f_{T+1}}{P_0(T+1)} \\ &\quad - \frac{P_0(T)}{P_0(T+1)} \log \frac{P_0(T)}{P_0(T+1)} \end{aligned} \quad (15)$$

$$\begin{aligned} H_e(T+1) &= - \sum_{i=T+2}^M \frac{f_i}{P_1(T+1)} \log \frac{f_i}{P_1(T+1)} \\ &= - \frac{P_1(T)}{P_1(T+1)} \sum_{i=T+2}^M \frac{f_i}{P_1(T)} \log \left\{ \frac{f_i}{P_1(T)} \frac{P_1(T)}{P_1(T+1)} \right\} \\ &= \frac{P_1(T)}{P_1(T+1)} H_e(T) + \frac{f_{T+1}}{P_1(T+1)} \log \frac{f_{T+1}}{P_1(T+1)} \\ &\quad - \frac{P_1(T)}{P_1(T+1)} \log \frac{P_1(T)}{P_1(T+1)} \end{aligned}$$

The recursive iteration is reduced by adding only the incremental part, and the search burden is reduced to  $O(M)$ . The same strategy can be applied to find the optimal threshold for  $TH_2$ . Assume the optimal threshold for  $CR_i$ , and  $CL_i$  are detected as  $TLR$ ,  $TLL$ , respectively, then  $TH_2$  is computed as  $TH_2 = \text{Min}(TLR, TLL)$ .

Figure 5 presents experimental results of our group detection strategy. As it demonstrates, video shots in one scene are semantically related, and parts of the shots will share the same

background or exhibit the same dominant color, this operation helps in merging shots in each scene into one or several groups. In Sect. 5.3, these groups will help the annotator acquire video context information and annotate video effectively.

## 5.2. Joint semantics and visual similarity for scene detection

Using the procedure described above, video shots can be parsed into semantically related groups. The semi-automatic annotation strategy in Sect. 5.3 then uses these groups to help the annotator determine the video context and semantics for annotation. After video groups have been annotated, semantics and low-level features are both available for each group. Thus, a group similarity assessment using joint semantic and visual features could be used to merge semantically related adjacent groups into scenes. With this detected scene structure, annotators can visually evaluate their annotation results. As we know, the group consists of spatially or temporally related shots, accordingly, the similarity between groups should be based on the similarity between shots.

### 5.2.1. Semantic similarity evaluation between shots

In Sect. 4, we specified that the mapping of each keyword has recorded the frame region where this keyword takes effect. To evaluate the semantic similarity between shots, this region should also be considered since the region information determines the importance of the keyword in describing the shot content. For Video Description, Group Description and Shot Description, the keywords at these levels have longer (or equal) duration than the current shot. Hence, they will be in effect over the entire shot. However, descriptors in the Frame Description may last only one or several frames in the shot, to calculate the semantic similarity between shots, the Effect Factor of each  $FD$  descriptor's keyword should be calculated first.

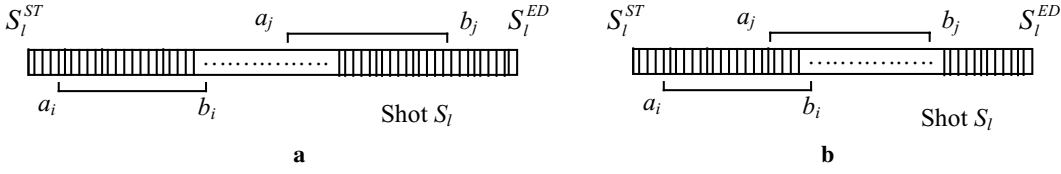
Assuming  $FD_k$  denotes the  $k^{\text{th}}$  keyword of  $FD$ . Given shot  $S_l$ , we suppose there are  $N$  mappings associated with  $FD_k$  in shot  $S_l$ , and their mapping regions are  $V_{a_1-b_1}^2, \dots, V_{a_N-b_N}^2$ . Given any two regions  $V_{a_i-b_i}^2, V_{a_j-b_j}^2$  ( $i \neq j, i, j \in N$ ) in these mappings, Fig. 6 shows two type of relationships between frame region  $(a_i, b_i)$  and  $(a_j, b_j)$  in shot  $S_l$ : with or without temporal overlap.

Assume operator  $\Theta(X, Y)$  denotes the number of overlapped frames between  $X$  and  $Y$ , then,  $\Theta(V_{a_i-b_i}^2, V_{a_j-b_j}^2)$  in Fig. 6 is given by Eq. 16:

$$\begin{aligned} \Theta(V_{a_i-b_i}^2, V_{a_j-b_j}^2) &= \begin{cases} 0 & (a_i, b_i) \text{ and } (a_j, b_j) \text{ have no overlap} \\ b_i - a_j & (a_i, b_i) \text{ and } (a_j, b_j) \text{ have overlap} \end{cases} \end{aligned} \quad (16)$$

Hence, the *Effect Factor* of keyword  $FD_k$  corresponding to shot  $S_l$  is defined by Eq. 17:

$$\begin{aligned} EF(FD_k, S_l) &= \frac{\sum_{m=1}^N (b_m - a_m) - \sum_{m=1}^{N-1} \sum_{n=m+1}^N \Theta(V_{a_m-b_m}^{ID}, V_{a_n-b_n}^{ID})}{S_l^{ED} - S_l^{ST}}, \\ m, n \in N \end{aligned} \quad (17)$$



**Fig. 6a,b.** Frame region relationship between  $(a_i, b_i)$  and  $(a_j, b_j)$  in Shot  $S_l$ . **a** Without temporal overlap, **b** with temporal overlap

where  $V_{a_1-b_1}^{ID}, \dots, V_{a_N-b_N}^{ID}$  is the mapping region associated to  $FD_k$ ,  $a_m$  and  $b_m$  denote the start and end frame of each mapping region. In fact, Eq. 17 indicates that the effect factor of keyword  $FD_k$  is the ratio of the number of all non-repeated frame regions mapping with  $FD_k$  and the number of frames in shot  $S_l$ . It is obvious that  $EF(FD_k, S_l)$  is normalized in  $[0,1]$ . The larger the value, the more important the keyword is in addressing the semantic content of  $S_l$ .

To evaluate the cross-intersection between keywords at various levels, we define  $\overline{VDS}_k, \overline{GDS}_k, \overline{SDS}_k, \overline{FDS}_k$  as the aggregation of keywords which have been used to annotate shot  $S_k$  in  $VD, GD, SD$  and  $FD$ , respectively. That is,  $\overline{GDS}_k$  denotes the keyword aggregation of all descriptors in  $GD$  which has been used in  $S_k$ , and so on. To describe the relationship among series of keywords  $(X_1, X_2, \dots, X_N)$ , three operators  $\{\Omega(X_1, X_2, \dots, X_N), \vartheta(X_1, X_2, \dots, X_N), \Psi(X)\}$  are defined:

1.  $\Omega(X_1, X_2, \dots, X_N) = \{X_1 \cup X_2 \cup \dots \cup X_N\}$  indicates the union of  $X_1, X_2, \dots, X_N$ .
2.  $\vartheta(X_1, X_2, \dots, X_N) = \{X_1 \cap X_2 \cap \dots \cap X_N\}$  is the intersection of  $X_1, X_2, \dots, X_N$ .
3.  $\Psi(X)$  represents the number of keywords in  $X$ .

Given any two shots  $S_i$  and  $S_j$ , assume their temporal description data ( $TDD$ ) are  $TDD_i = \{S_i^{ID}, S_i^{ST}, S_i^{ED}, Map(KA, V)\}$  and  $TDD_j = \{S_j^{ID}, S_j^{ST}, S_j^{ED}, Map(KA, V)\}$ , respectively. Assuming also that  $KAS_i$  denotes the union of all keywords that have been shown in annotating shot  $S_i$ , then  $KAS_i = \Omega(\overline{VDS}_i, \overline{GDS}_i, \overline{SDS}_i, \overline{FDS}_i)$  and  $KAS_j = \Omega(\overline{VDS}_j, \overline{GDS}_j, \overline{SDS}_j, \overline{FDS}_j)$ . The semantic similarity between shot  $S_i$  and  $S_j$ , can be evaluated using Eq. 18:

$$\begin{aligned}
 & SemShotSim(S_i, S_j) \\
 &= W_V \frac{\Psi(\vartheta(\overline{VDS}_i, \overline{VDS}_j))}{\Psi(\Omega(\overline{VDS}_i, \overline{VDS}_j))} + W_G \frac{\Psi(\vartheta(\overline{GDS}_i, \overline{GDS}_j))}{\Psi(\Omega(\overline{GDS}_i, \overline{GDS}_j))} \\
 &+ W_S \frac{\Psi(\vartheta(\overline{SDS}_i, \overline{SDS}_j))}{\Psi(\Omega(\overline{SDS}_i, \overline{SDS}_j))} + \\
 &\quad \sum_k \{EF(FD_k, S_i) \cdot EF(FD_k, S_j)\} \\
 &\times W_F \frac{FD_k \in \vartheta(\overline{FDS}_i, \overline{FDS}_j)}{\Psi(\Omega(\overline{FDS}_i, \overline{FDS}_j))} \quad (18)
 \end{aligned}$$

From Eq. 18, we can see that the semantic similarity between shot  $S_i$  and  $S_j$  is the weighted sum of the cross intersection of keywords at various levels. From  $VD$  to  $FD$ , the keywords will address more and more detailed information in the shot. Therefore, in our system we set the weight of various levels ( $W_V, W_G, W_S, W_F$ ) to 0.4, 0.3, 0.2 and 0.1 respectively. That is, the higher the level, the more important the keyword is in addressing content.

### 5.2.2. Unified similarity evaluation joint semantics and visual similarity

With the semantic similarity between  $S_i$  and  $S_j$ , the unified similarity which joint visual features and semantics is given by Eq. 13:

$$\begin{aligned}
 ShotSim(S_i, S_j) &= (1 - \alpha) \cdot StSim(S_i, S_j) \\
 &+ \alpha \cdot SemShotSim(S_i, S_j) \quad (19)
 \end{aligned}$$

where  $StSim(S_i, S_j)$  indicates the visual similarity which is specified in Eq. 8.  $\alpha \in [0,1]$  is the weight of semantics in similarity measurement, which can be specified by users. The larger the  $\alpha$ , the greater the importance is given to the semantics in the overall similarity assessment. If  $\alpha = 0$ , we use only visual features to evaluate the similarity between  $S_i$  and  $S_j$ .

Based on Eq. 19, given a shot  $S_i$  and a video group  $G_j$ , the similarity between them can be calculated using Eq. 20:

$$StGpSim(S_i, G_j) = Max\{ShotSim(S_i, S_l)\}_{S_l \in G_j} \quad (20)$$

This indicates that the similarity between shot  $S_i$  and group  $G_j$  is the similarity between  $S_i$  and its most similar shot in  $G_j$ .

In general, when comparing the similarity between two groups using the human eye, we usually use the group with less shots as the benchmark, and then determine whether there is any shot in the second group similar to certain shots in benchmark group. If most shots in the two groups were similar enough, we would consider these groups to be similar. Accordingly, given group  $G_i$  and  $G_j$ , assume  $\hat{G}_{i,j}$  is the benchmark group, and  $\tilde{G}_{i,j}$  is the other group. Suppose  $M(X)$  denotes the number of shot in group  $X$ , then, the similarity between  $G_i$  and  $G_j$  is given in Eq. 21:

$$\begin{aligned}
 & GroupSim(G_i, G_j) \\
 &= \frac{1}{M(\hat{G}_{i,j})} \sum_{l=1; S_l \in \hat{G}_{i,j}}^{M(\hat{G}_{i,j})} StGpSim(S_l, \tilde{G}_{i,j}) \quad (21)
 \end{aligned}$$

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

### 5.2.3. Scene detection

As we defined in Sect. 5.1, a *video scene* conveys a high-level concept or story and usually consists of one or more semantically related adjacent groups. For annotating, we ignore the scene level description, since automatic scene detection with satisfactory results is not yet available. After most

video groups have been annotated, we can integrate semantics and visual features among adjacent groups to merge similar groups into semantically related units (scenes). The constructed scenes will help the annotator visualize and refine the annotation results. To attain this goal, all neighboring groups with significant similarity in semantic and visual features will be merged using strategy below:

1. Given any group  $G_i$ , assume  $GDE_i$  denotes the keyword aggregation of  $GD$ 's event descriptors which has been used in all shots of  $G_i$ .
2. For any neighboring groups  $G_i$  and  $G_j$ , if  $\vartheta(GDE_i, GDE_j) = \emptyset$ , these two groups are not merged. Otherwise, go to step 3. That is, if the keyword two groups' event descriptor is totally different, they cannot be merged into one group.
3. Using Eq. 15 to calculate the overall similarity between these two groups; go to step 2 to find all other neighboring groups' similarities. Then go to step 4.
4. All neighboring groups with their similarities larger than threshold  $TH_3$  ( $TH_3 = TH_2 - 0.2$ ) are merged into one group. As a relatively special situation, if there are more than 2 sequentially neighboring groups, e.g.  $A, B, C$ , with similarities  $GroupSim(A, B)$  and  $GroupSim(B, C)$  both larger than threshold  $TH_3$ , all groups are merged into a new group (scene).

Clearly, groups in one scene should have higher correlations in semantic and visual features. By integrating semantic and visual similarity, they will have a higher probability of being merged into one scene.

### 5.3. Semi-automatic video annotation

As we stated above, a sequential annotation strategy is a time consuming and burdensome operation. Instead of annotating the video sequentially, we can utilize the results of the video processing techniques above to help annotators determine the video context and semantics. Some semi-automatic annotation schemes have been implemented in image database [47, 48] using semantics, visual features and relevance feedback [49] to assist the annotator in finding certain type of images for annotation. Based on these schemata, a semi-automatic annotation scheme for video database is presented in this section and the main flow is shown in Fig. 1.

#### 5.3.1. Semi-automatic annotation with relevance feedback

As the first step, the shot grouping method is adopted to segment spatially or temporally related shots into groups (Since the video semantics are not available in current stage, our group detection strategy uses only low-level features.). Then, the groups are shown sequentially in the interface for annotation, as shown in Fig. 7. Given any group in the interface, the annotator has three options:

1. Annotate a certain shot by double clicking the key-frame of the shot (the result is illustrated in Fig. 8.) The annotator can assign both shot description and frame description keywords related to the shot (and frames in the shot). A series of function buttons such as play, pause, etc. are available

to help the annotator browse video shots and determine semantics of the shot and frames.

2. If the annotator thinks that the current group belongs to the same event category, he (she) can specify group description and video description keyword(s) to the group by clicking the hand-like icon at the left of the group, and select corresponding keywords to annotate it.
3. If the annotator thinks the current group contains multiple events, he (she) can manually separate it into different groups (with each groups belonging to only one event category) by dragging the mouse to mask shots in the same event category and then clicking the hand-like icon to assign keywords. By doing this, the current group is separated into several groups and shown separately in the interface.

As described in Sect. 4, since we use a hierarchical content description ontology and shot based annotation data structure, all units at the lower level inherit keywords from the level(s) above. For example, if we assign a group description keyword to a group, all shots in this group are annotated with this keyword.

At any annotation state, the annotator can select one or a group of shots as the query to find similar groups for annotation. Then, the relevance feedback ( $RF$ ) strategy is activated to facilitate this operation:

1. All selected shot(s) are treated as a video group. The annotator should assign keywords to annotate them before the retrieval.
2. After the annotator clicks the "find" button, the similarity evaluation strategy in Eq. 21 is used to find similar groups.
3. After the result has been retrieved, the annotator can either annotate those similar groups separately or mark the result (or part of them) as the feedback examples, and click " $RF$ " button to trigger a  $RF$  processing. By selecting  $RF$ , all selected shots are annotated with keywords the annotator specified in step 1. Then Eq. 22 is used to retrieve other similar groups.

Recursively execute relevance feedback iterations above, more and more video groups could be annotated. In this situation, the system works like a video retrieval system. However, there is some difference, since the retrieved groups are not shown sequentially from top to bottom using their similarity scores. Instead, they remain located at their original position, because the annotator needs the preceding or succeeding groups to provide the context information for annotation. The similarity score is displayed at the left of each group. For the sake of saving interactive space, those groups with a large distance (determined by the annotator's specification of the number of groups to be returned) to current feedback iteration are displayed as a small symbol in the interface. A double click on the symbol displays all shots in the group on the screen.

Equation 22 presents the simplified  $RF$  model in our system (based on Bayesian formula [50]), there are no negative feedback examples in our system; all selected shots are treated as positive feedback examples.) Assuming  $G_i$  denotes the selected feedback examples in current iteration, for any group  $G_j$  in the database, its global similarity  $Sim(j)^k$  in the current iteration ( $k$ ) is determined by its global similarity in the previous iteration  $Sim(j)^{k-1}$  and its similarity to current selected feedback examples  $GroupSim(G_i, G_j)$ .  $\eta$  is an operator that

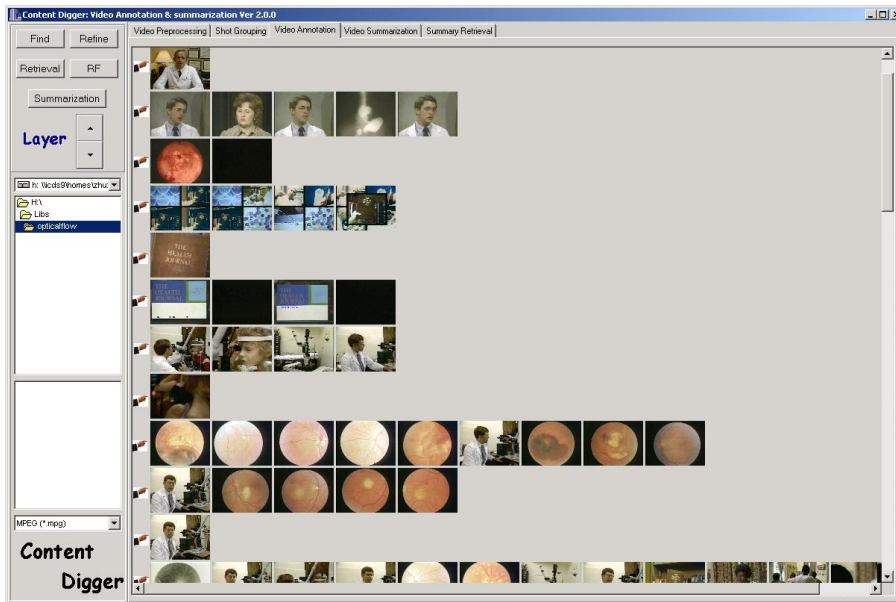


Fig. 7. Hierarchical video content annotation interface

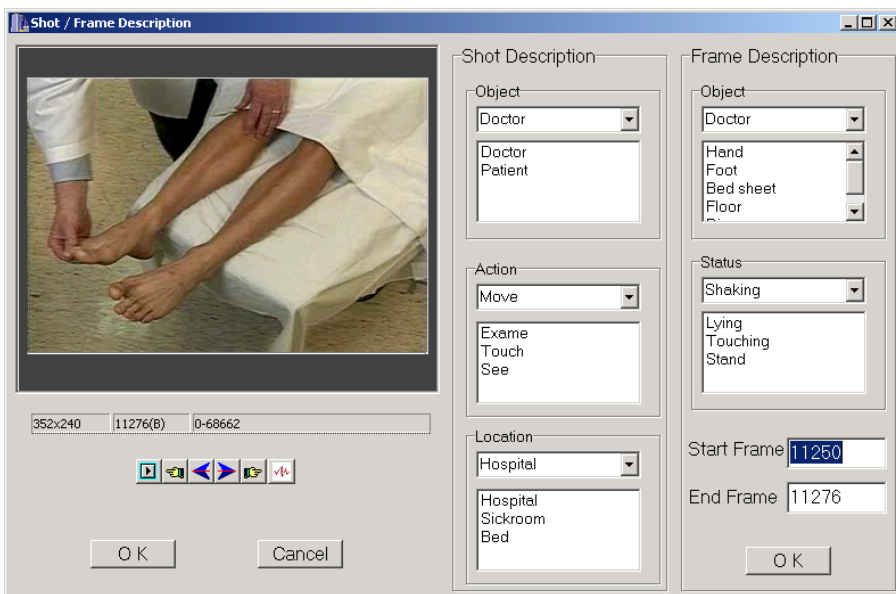


Fig. 8. Shot and frame annotation interface

indicates the influence of the history to the current evaluation. In our system we set  $\eta = 0.3$ .

$$Sim(j)^k = \eta Sim(j)^{k-1} + (1 - \eta) GroupSim(G_i, G_j) \quad (22)$$

### 5.3.2. Annotation results visualization and refinement

In content description ontology, we ignore the scene level description since automatic scene detection is not yet available. However, by integrating the annotated semantics related to groups, we can merge semantically related adjacent groups into scenes, and present the annotation results to the viewers. The annotator can accordingly evaluate and refine the annotations with more efficiency.

1. At any annotation stage, the annotator can click the “refine” button (shown in Fig. 7). Then, the scene detection strategy

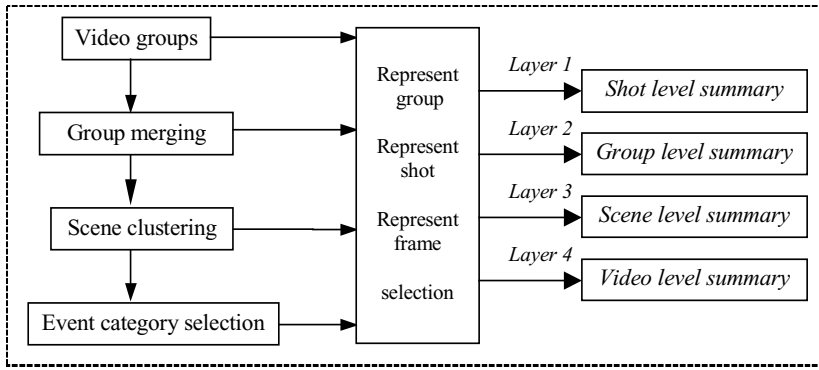
in Sect. 5.2) is invoked to merge adjacent similar groups into scenes.

2. The annotator can specify different values for  $\alpha$  to modify the contribution of the semantics to similarity assessment and to evaluate the annotation quality in different situations.

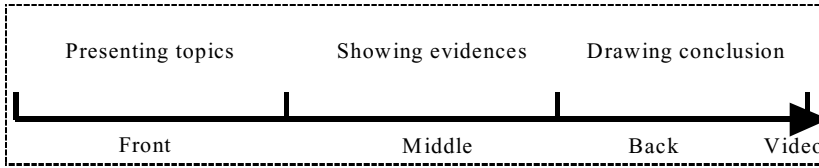
By doing this, the annotator can visually browse the video content structure and evaluate the quality of the annotation. After that, the annotator can terminate or resume the operation at any time, i.e. a series of annotation, refinement, annotation, can be recursively executed until a satisfactory result is achieved.

Using these strategies, a more reliable and efficient video content annotation is achieved. It is better than manual manner in terms of efficiency, and better than automatic scheme in terms of accuracy:

1. A hierarchical video content description ontology is utilized in the system, which address the video content in



**Fig. 9.** Diagram of hierarchical video summarization



**Fig. 10.** Simplified video scenario and content presentation model

different granularity. The categories, events and detailed information about the video are presented in different levels. It can minimize the influence of the annotator’s subjectivity, and enhance the reusability of various annotators’ descriptive keywords.

2. A semi-automatic annotation strategy is integrated which utilizes various video processing techniques (video shot and group detection, joint semantic and visual similarity for scene detection, relevance feedback) to help annotators acquire video context information and annotate video more efficiently. Moreover, the annotated video content structure could be visualized directly.

## 6. Hierarchical video summarization and skimming

Among all five layers of the video content hierarchy, it is the scene that conveys the semantic meaning of the video to the viewers by using groups, shots and frames to address the detailed information within each story unit. Hence, the video summary should also fit this hierarchy by presenting the digest at different levels and with different granularity. Generally, video summaries built only on low-level features are either too rough to retain video content or they contain too much redundancy since it is not possible for them to get semantics. Using the content description data acquired by methods described in previous sections, we can address video content at various level and different granularity, and present a meaningful video summary. In this way, the visual features and semantic information are integrated to construct a four layer summary: *shot level summary*, *group level summary*, *scene level summary* and *video level summary*. These levels correspond to the summary at each layer of the content hierarchy, and present the video digest from top to bottom in increasing granularity. The flow chart of the proposed approach is shown in Fig. 9. Note that:

1. Since groups consist of spatially or temporally related shots, they are the ideal units to summarize shots. That

is, video summary at the shot level (*layer 1*) consists of all groups to uncover details among the video.

2. As we defined in Sect. 5.1, a *video scene* conveys a high-level concept or story, and it usually consists of one or more semantically related groups. By using the scene detection strategy in Sect. 5.2, we can combine similar groups into semantically related units (scenes) and use them as group level summaries.
3. Since similar scenes may be shown repeatedly in the video, the summary at the scene level is determined by clustering algorithms to eliminate redundant scenes in the video and present a visually and semantically compact summary.
4. In general, video scenarios can be separated into three parts: (1) presenting subject or topic information; (2) showing evidence and details; and (3) drawing conclusions. These three parts are usually shown separately at the *front*, *middle*, and *back* of the video. A simplified video scenario and content presentation model is shown in Fig. 10. Hence, the summary at the video level is constructed by selecting meaningful event categories from the third layer summary to fit this model and supply the viewer with the most general overview.

Figure 11 illustrates the corresponding steps to construct the summary, which could be described below:

**Input:** video groups, temporal description stream (*TDS*) of the video.

**Output:** A four layer video summary in pictorial format or in trimmed video stream format (skimming).

**Procedure:**

1. Construct the summary at layer 1 (*shot level summary*): Using all groups as summary candidate groups, use *SelectRepShot()* and *SelectRepFrame()* (introduced below) to select representative shots and representative frames for each candidate group. The combination of these representative frames and shots will form the video skimming and pictorial summary at shot level.

2. Construct the summary at layer 2 (*group level summary*): Set  $\alpha = 0.3$  and use the scene detection strategy introduced in Sect. 5.2 to merge those neighboring groups with higher semantic and visual similarity into scenes. Then, use *SelectRepGroup()* (introduced below) to select the representative group for each scene. Take all representative groups as summary candidates, where *SelectRepShot()* and *SelectRepFrame()* are used to select representative shots and frames, and assemble them to form the second layer video skimming and summary. The experimental results in Sect. 7 illustrates why we set  $\alpha$  to 0.3.

In general, neighboring groups in the same scene are related with each other semantically, even they have relatively low visual similarity. On the other hand, if they belong to different scenes, there will be no correlation with their semantics. Hence, by considering visual features and semantics, a scene structure is determined which supplies a well-organized summary for groups.

3. Construct the summary at layer 3 (*scene level summary*): Based on scene detection results, the *SceneClustering()* (introduced below) is applied to cluster all detected scenes into a hierarchical structure. Then use *SelectRepGroup()* to select representative groups. Their representative frames and shots are assembled as the video summary and skimming at the third layer.

Since similar scenes are usually shown in the video several times, a clustering operation will eliminate redundancy among them and present a compact summary of scenes.

4. Construct the summary at layer 4 (*video level summary*): The event category selection for video level summary construction is executed by selecting one group that belongs to different event categories from each part (*front*, *middle*, *back*) of the video, and then assembling those groups to form the summary at the highest layer. Usually, different events vary in their ability to unfold the semantic content of different types of videos. For example, the presentation and dialog events in medical videos are more important than events such as surgery, experiment, etc., since the former uncovers general content information, and the latter address the details. A scheme for selecting event categories to abstract medical video is proposed in [51]. As a general video summarization strategy, we merely suppose all event categories have the same importance. The following sequential selection strategy is adopted:

- a. Separate the video almost equally into three parts {*front*, *middle*, and *back*}.
- b. For each part of the video, assuming there are  $N$  representative groups,  $RG_1, RG_2, \dots, RG_N$ , which have been selected as the *scene level summary*. Given any representative group ( $RG_i$ ), since  $GDE_i$  denotes the keyword aggregation of the event descriptors of all shots in  $RG_i$ ,  $\Omega(GDE_1, \dots, GDE_N)$  denotes the union of  $GDE_1, \dots, GDE_N$ . Sequentially check each representative group ( $RG_1, \dots, RG_N$ ), and the group  $RG_i$  with  $GDE_i \subset \Omega(GDE_1, \dots, GDE_N)$  is selected as the summary candidate. Then, all keywords in  $GDE_i$  are deleted from  $\Omega(GDE_1, \dots, GDE_N)$ . That is, only one group is selected to represent each event type.

- c. Recursively execute step “b” until there is no keyword contained in  $\Omega(GDE_1, \dots, GDE_N)$ , and go to step “d”.
  - d. Use the same strategy to select summary candidate groups for other two parts of video, then use *SelectRepShot()* and *SelectRepFrame()* to select the representative shots and frames for all selected summary candidate groups. Their combination will form the highest layer video skimming and summary.
5. Organize the hierarchical video summary structure and return.

Figure 12 shows the system interface of the hierarchical video summarization. The first row shows the pictorial summary of current layer and all other rows indicate current group information. The skimming of each layer is stored as *MPEG* file on disk.

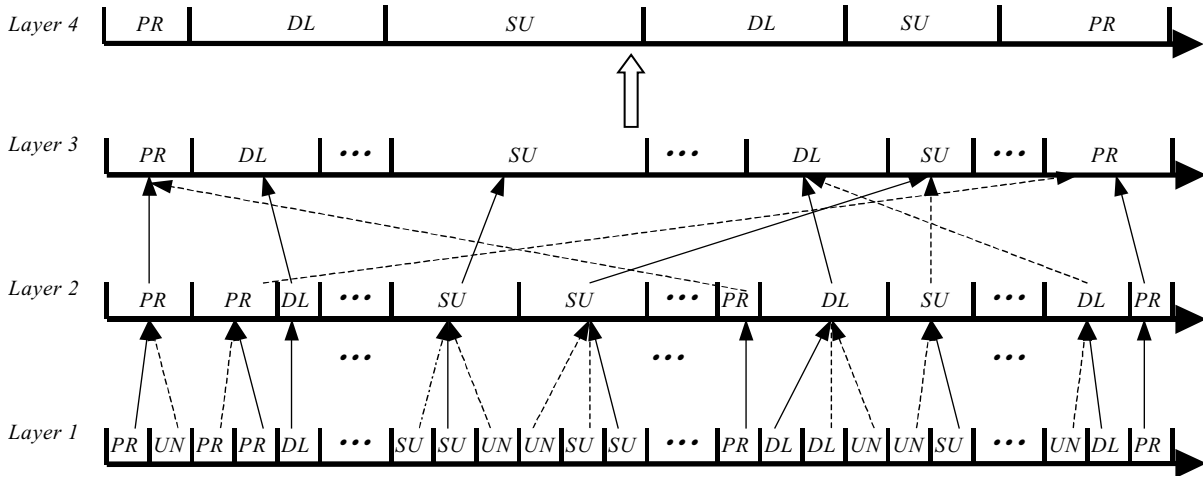
In order to utilize the video content description data more efficiently, a keyword list is also constructed by gathering all keywords of selected representative shots in current layer. Hence, for video summary at layer  $k$ , the keyword lists,  $\Omega(KAS_1, KAS_2, \dots, KAS_N)$  ( $N$  is the number of shot in current summary layer  $k$ ), is also displayed to supply the viewer with a compact textual description of video content.

To present and visualize video content for summarization, the representative unit for scene, group and shot are selected to construct the pictorial summary or skimming. Various strategies below are utilized to select representative shots (*SelectRepShot*) and frames (*SelectRepFrame*) from groups, and select representative groups from scenes (*SelectRepGroup*). Moreover, the scene clustering algorithm (*SceneClustering*) and similarity evaluation between scenes (*SceneSim*) are also proposed.

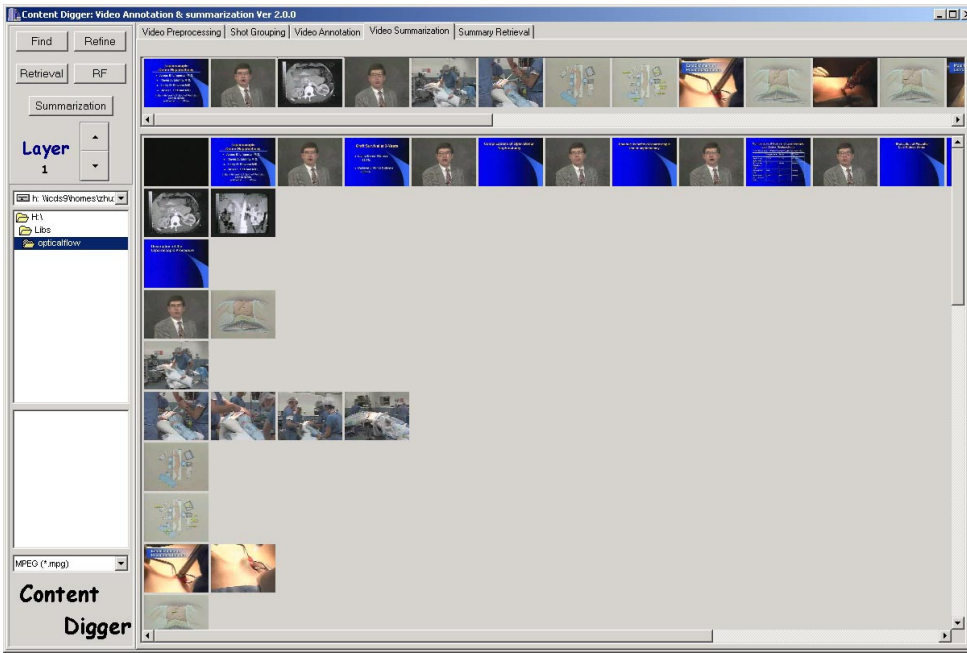
#### [SelectRepShot]

The representative shot of group  $G_i$  is defined as the shot that represents the most content in  $G_i$ . Given any group  $G_i$ , this procedure will select its representative shot  $RT_i$ . In Sect. 5.1, we have merged all shots in  $G_i$  into  $N_c$  clusters, these clusters will help us to select the representative shots. 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 selection of the representative shot for  $G_i$  is based on the cluster information and the description keywords among shots:

1. Given  $N_c$  clusters  $C_i (i = 1, \dots, N_c)$  in  $G_i$ , use steps 2, 3 and 4 to extract one representative shot for each cluster  $C_i$ . In all, there are  $N_c$  representative shots selected for each  $G_i$ .
2. Given any cluster  $C_i$ , the shot in  $C_i$  which has more keywords and larger time duration usually contains more semantics. Hence, it is selected as the representative shot. Notice that  $KAS_i$  denotes the union of all keywords which have been shown in describing shot  $S_i$ , then  $\Psi(KAS_i)$  indicates the number of keywords in shot  $S_i$ .
  - $RT = \arg \max_{S_i} \{\Psi(KAS_i), S_i \in G_i\}$ , if there is only one shot contained in  $RT$ , it is selected as the representative shot of  $G_i$ .
  - Otherwise, the shot in  $RT$  that has the largest time duration is selected as the representative shot.



**Fig. 11.** Hierarchical video summarization strategy. (*PR*, *DL*, *SU* and *UN* represent presentation, dialog, surgery, and unknown events respectively; the solid line indicates that this group is selected as the representative group of the newly generated unit)



**Fig. 12.** Hierarchical video summary at the first layer (with the first row presents the video summary at current layer, other rows in the interface indicate current group information)

3. The collection of representative shot for each cluster  $C_i$  forms the representative shots of  $G_i$ .

#### [SelectRepFrame]

After we find the representative shot(s) for each group, the key frame of the representative shot(s) is taken as the representative frame(s) of the group.

#### [SelectRepGroup]

The representative group of scene  $SE_i$  is defined as the group in  $G_i$  which addresses the most content information of  $SE_i$ . After group merging or scene clustering, the similar groups are merged to form a semantically richer unit (scene), the representative group of the constructed scene should be selected

to represent and visualize its content. Assume  $SE_i$  represents the newly generated scene which is merged by  $N_i$  groups ( $G_l$ ,  $l = 1, \dots, N_j$ ), i.e.  $SE_i = G_1 \cup G_2 \cup \dots \cup G_{N_i}$ . Assuming  $KAG_l$  denotes the union of keywords which have been shown in describing all shots in  $G_l$ . Then  $\Psi(KAG_l)$  indicates the number of keywords in  $G_l$ , and  $\Psi(\Omega(KAG_1, \dots, KAG_{N_i}))$  denotes the number of keywords which have been shown in describing all shots in  $SE_i$ . The representative group ( $RG_i$ ) of  $SE_i$  is selected using the procedure below:

- $RG = \arg \max_{G_l} \left\{ \frac{\Psi(KAG_l)}{\Psi(\Omega(KAG_1, \dots, KAG_{N_i}))}, l = 1, \dots, N_i \right\}$ . That is, the group in  $SE_i$  which contains most of keywords in  $SE_i$  is considered as the representative group. If there is only one group contained in  $RG$ , it is taken as the representative group ( $RG_i$ ) of  $SE_i$ .
- Otherwise, the group in  $RG$  which has the longest time duration is selected as the representative group of  $SE_i$ .

### [SceneClustering]

As shown in Fig. 11, this procedure will construct a hierarchy beyond the *scene level summary*. After scene detection, the newly generated scenes may consist of several other groups. In this case,  $SceneSim()$  (introduced below) is used to calculate the similarity between scenes. The procedure of scene clustering is given below:

1. Since a scene with too many shots may result in an ambiguous event category, and in addition, may result in a higher probability to absorb other groups, any scene containing more than 20% of shots in the video is no longer used for clustering.
2. Use the typical clustering algorithm – *ISODATA* to merge similar scenes into classes. While clustering, the scene similarity is calculated using step 3 and 4 below.
3. Set  $\alpha = 0.3$ . Given any two scene  $SE_i$  and  $SE_j$ , assume there are  $N_i$  and  $N_j$  groups  $(G_{i1}, G_{i2}, \dots, G_{iN_i}; G_{j1}, G_{j2}, \dots, G_{jN_j})$  contained in  $SE_i$  and  $SE_j$  respectively, then  $\Omega(GDE_{i1}, \dots, GDE_{iN_i})$  denotes keywords of event descriptors which have been shown in all groups of  $SE_i$ . If  $\vartheta(\Omega(GDE_{i1}, \dots, GDE_{iN_i}), \Omega(GDE_{j1}, \dots, GDE_{jN_j})) = \emptyset$ , the similarity between them is set to 0. Otherwise, go to step 4. That is, the scenes with mutually exclusive keywords of event descriptors cannot be merged into one class.
4. Use  $SceneSim()$  to calculate the overall similarity between scene  $SE_i$  and  $SE_j$ ;
5. Return the clustered scene structure.

### [SceneSim]

Video scenes consist of groups. Hence, given scene  $SE_i$  and  $SE_j$ , assume there are  $N_i$  and  $N_j$  groups  $(G_{i1}, G_{i2}, \dots, G_{iN_i}; G_{j1}, G_{j2}, \dots, G_{jN_j})$  contained in  $SE_i$  and  $SE_j$ , respectively. We define  $K(SE_i)$  to be all groups in  $SE_i$ ,  $N(SE_i)$  denotes the number of groups in  $SE_i$ , that is  $K(SE_i) = \{G_{i1}, G_{i2}, \dots, G_{iN_i}\}$ , and  $N(SE_i) = N_i$ . Compare  $SE_i$  and  $SE_j$ , the scene containing fewer groups is selected as the benchmark scene. We denote the benchmark scene as  $\hat{B}_{i,j}$ , and the other scene is denoted as  $\tilde{B}_{i,j}$ . Then, the similarity between  $SE_i$  and  $SE_j$  is calculated using Eq. 17. That is, the similarity between any two scenes is the average maximal similarity between the groups in the benchmark scene and their most similar groups in the other scene.

$$\begin{aligned}
 & SceneSim(SE_i, SE_j) \\
 &= \frac{1}{N(\hat{B}_{i,j})} \sum_{l=1, G_l \in K(\hat{B}_{i,j})}^{N(\hat{B}_{i,j})} \\
 & \quad \text{Max}\{GroupSim(G_l, G_k); \\
 & \quad G_k \in K(\tilde{B}_{i,j}); k = 1, 2, \dots, N(\tilde{B}_{i,j})\} \quad (23)
 \end{aligned}$$

## 7. Experimental results, analysis and applications

Two types of experimental results, group detection and hierarchical summarization, and some potential applications of the

proposed strategies are presented in this section. About eight hours of medical videos and four hours of news programs are used as our test bed (all the video data are *MPEG-I* encoded, with the digitization rate equal to 30 frames/s). Videos are first parsed with the shot segmentation algorithm to detect the gradual and break changes. The gradual change frames between shots have been removed successfully during shot segmentation.

### 7.1. Group detection results

The group detection experiment is executed among four medical videos and four news programs. Experimental comparisons are made with [40, 45] (in [40], we only use their group detection strategy). Moreover, to judge the quality of the detected results, the following rule is applied: the group is judged to be correctly detected if and only if all shots in the current group belong to the same scene (semantic unit), otherwise the current group is judged to be falsely detected. Thus, the group detection precision ( $P$ ) in Eq. 24 is used for performance evaluation.

$$P = \frac{\text{How many groups are rightly detected}}{\text{The number of detected groups}} \quad (24)$$

Clearly, by treating each shot as one group, the group detection precision would be 100%. Hence, another *compression rate factor (CRF)* is also defined in Eq. 25:

$$CRF = \frac{\text{Detected group number}}{\text{total shot number in the video}} \quad (25)$$

The experimental results and comparisons of group detection strategies are given in Table 5.

To identify the methods used in the experiment, we denote our method as  $A$ , and the two methods in [40, 45] are denoted as  $B$  and  $C$  respectively. From the results in Table 5, some observations can be made:

- Our video grouping methods achieves the best precision among all methods; about 87% shots are assigned in the right groups. Hence, the annotator will not be required to separate many groups which contain multiple scenes.
- Comparing all three methods, since method  $C$  is proposed for scene detection, it achieves the highest compression rate. However the precision of this method is also the lowest. On the other hand, this strategy is a threshold based method, there is no doubt that some of the groups are over segmented or missed.
- As a trade-off with precision, the compression ratio of our method is the worst (28.9%), that is, in an average situation, each group consists of approximately 3.5 shots. However, to supply the video group for content annotation, it is often worse to fail to segment distinct boundaries than to over-segment a scene. In addition, other strategies in the paper such as group merging, hierarchical summarization will enforce the compression ratio. From this point of view, our system achieve relatively better performance.

**Table 5.** Video group detection results

Movie name	Shots	A			B			C		
		Detected Groups	$P$	CRF	Detected Groups	$P$	CRF	Detected Groups	$P$	CRF
Medical 1	265	62	0.89	0.23	34	0.78	0.13	26	0.66	0.098
Medical 2	221	70	0.84	0.32	38	0.67	0.17	18	0.57	0.081
Medical 3	388	121	0.82	0.31	61	0.64	0.16	39	0.72	0.101
Medical 4	244	72	0.87	0.30	38	0.76	0.15	27	0.48	0.111
News 1	189	58	0.91	0.31	22	0.81	0.12	14	0.64	0.074
News 2	178	46	0.85	0.26	26	0.72	0.15	18	0.71	0.101
News 3	214	57	0.91	0.27	24	0.76	0.11	23	0.62	0.107
News 4	190	59	0.90	0.31	27	0.80	0.14	19	0.67	0.100
Average	1889	545	0.87	0.289	270	0.74	0.143	184	0.64	0.097

### 7.2. Hierarchical video summarization results

As stated before, a four layer video summary is produced for each video. Three questions are introduced to evaluate the quality of the summary 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 scenario of the video? (3) Is the summary concise? For each of the questions, a score from 0 to 5 (where 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. 13). A second evaluation process uses the rate between the numbers of representative frames at each layer and the number of all key frames to indicate the compression rate ( $RC$ ) of the video summary. To normalize this value with the scores of the questions, we multiply  $RC$  by 5 and use this value in Fig. 13. The influence of the factor  $\alpha$  with the quality of the summary is also addressed by changing the value of  $\alpha$  and evaluating the generated summaries. The results are shown in Figs. 13–15.

From Figs. 13–15, we can see that as we move to lower levels, the ability of the summary to cover the main topic and the scenario of the video is greater. The conciseness of the summary is the worst at the lowest level, since as the level decreases, more redundant groups are shown in the summary. 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 users an overview of the video selected from the database for the first time.

Comparing Figs. 13–15, the influence of the factor  $\alpha$  on the video summary can be clearly evaluated. When  $\alpha$  changes from 0 to 0.5, the summary at each layer has larger and larger compression ratio ( $RC$ ). A more concise video summary is acquired for each layer, since with the increase in significance of semantics in the similarity evaluation, all groups with the same event categories would be grouped into one group, and the visual similarity among those groups tends to be neglected. At higher levels, the summary is more concise, however, the

scenario of the video summary is worse. Based on these results, we set  $\alpha = 0.3$  in most other experiments.

To evaluate the efficiency of our hierarchical video summarization more objectively, a retrieval based evaluation method is presented. In this experiment, all 16 videos in the database are first annotated with our semi-automatic annotation strategy, and then each video is manually separated into three clips (the whole video database contains  $16 \times 3 = 48$  clips). There is no shot or group overlapping with the manually segmented boundaries; that is, the boundary of the segmentation is also the boundary of the shot and group. We then use hierarchical summarization to construct the summary for each clip. We randomly select one clip from database, and use its summary from different layers as the query to retrieve from the database. The ranks of the other two clips which are in the same video as the query are counted to evaluate the system efficiency. The similarity assessment between the query summary and clips in video database is evaluated using the strategy below:

- For any given query summary at a specified level  $i$ , collect all its representative groups at this level, and denote it as  $QG_i = \{RG_{i1}, \dots, RG_{iN}\}$ , where  $N$  indicates the number of representative groups.
- For clip  $CP_j$  in the database, gather all its representative groups at each level  $k$ , and denote the result as  $DG_j^k = \{RG_{j1}, \dots, RG_{jM}\}$ , where  $M$  indicates the number of representative groups.
- Use Eq. 26 to calculate the similarity between the query summary and the summary at layer  $k$  in  $CP_j$ .
- The similarity between the query summary and  $CP_j$  is evaluated with Eq. 27, and its rank is used for system performance evaluation.

$$SumtoSumSim(QG_i, DG_j^k)$$

$$= \begin{cases} \frac{1}{N} \sum_{l=1}^N \text{Min}\{\text{GroupSim}(RG_{il}, RG_{ja}); \\ RG_{il} \in QG_i, RG_{ja} \in DG_j^k, a = 1, \dots, M\} \\ \quad \text{if } N \leq M \\ \frac{1}{M} \sum_{l=1}^M \text{Min}\{\text{GroupSim}(RG_{jl}, RG_{ia}); \\ RG_{jl} \in DG_j^k, \\ RG_{ia} \in QG_i, a = 1, \dots, N\} \quad \text{Otherwise} \end{cases} \quad (26)$$

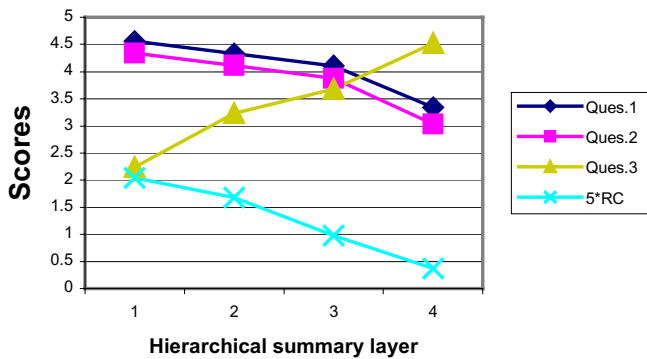


Fig. 13. Hierarchical video summary evaluation ( $\alpha = 0.3$ )

$$\begin{aligned} & SumtoViSim(QG_i, CP_j) \\ & = \min_{k=1,2,3,4} \{ SumtoSumSim(QG_i, DG_j^k) \} \end{aligned} \quad (27)$$

We randomly select 16 retrieval results from the medical video and news programs, and show them in Table 6. To reduce retrieval time, we select only the summary at level 3 as the query to compare the similarity, and only summaries of the highest three levels ( $k = 4, 3, 2$ ) in the database are used. From Table 6, we see that our retrieval strategy achieves reasonably good results: the average location of retrieved clips that are in the same video as the query is 3.187 (out of 47 clips, since the query clip is excluded from database). Nevertheless, we notice that the retrieval results for news are worse than for medical videos, because the news programs are usually different from general video data. In common videos, a similar scene may be shown repetitively in the video, but in news programs, most story units are only reported once. Hence, the summary at the front part of the video may be quite different from the summary at the middle or back part.

In addition, to address the influence of the layer of query summary with the retrieval efficiency, the summaries at different layers ( $k = 4, 3, 2$ ) are used as queries to retrieve from the database (to evaluate the influence of the semantic in video retrieval, we set  $\alpha$  equal to 0.3 and 0.0 respectively.) The results are shown in Fig. 16. It can be seen that with the layer goes higher, the retrieval accuracy become worse. With  $\alpha = 0.3$ , even at the highest level, the average location of a correctly retrieved clips is ranked 5.26, which is still much better than the query results (6.788) of retrieval at level 2 with  $\alpha = 0.0$ . Thus, by considering structured video semantic information, the video retrieval results are improved substantially.

We do the retrieval in only three layers ( $k = 4, 3, 2$ ), since more time is needed to calculate the similarity at lower layers.

The system is implemented in C++ with an *MPEG-I* decoder that has been developed by our group. Since we need to generate the video skimming at each level, *MPEG-I* editing tools have also been developed to assemble several video clips into one *MPEG-I* stream with integrated audio signal.

### 7.3. Potential application domains

With proposed strategies, video content description and summarization could be acquired in an improved way, some potential applications may also be implemented within domains below:

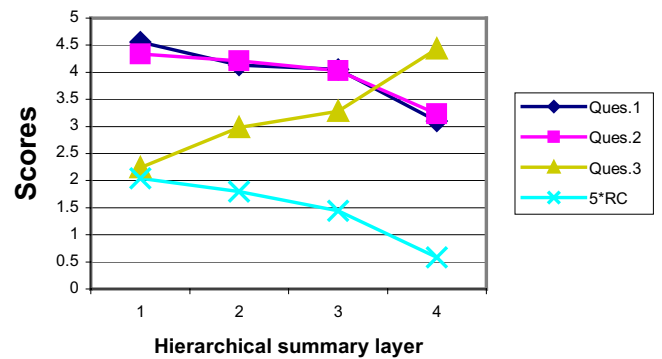


Fig. 14. Hierarchical video summary evaluation ( $\alpha = 0.0$ )

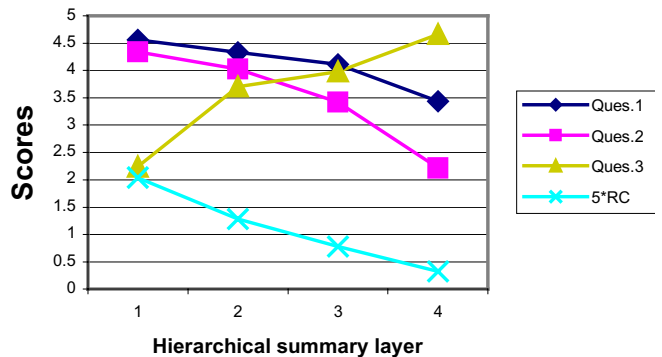


Fig. 15. Hierarchical video summary evaluation ( $\alpha = 0.5$ )

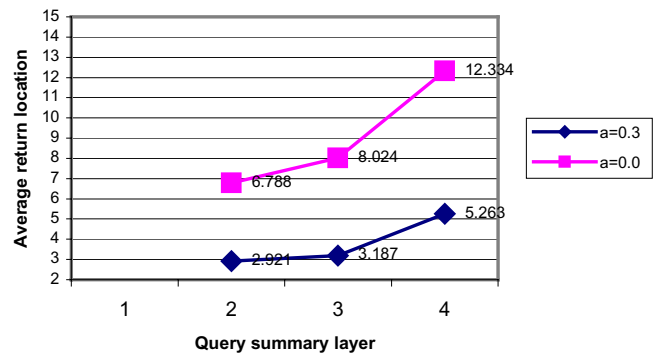


Fig. 16. Video retrieval results with summaries at different levels

1. Comprehensive content annotation for special videos or home entertainment videos  
As we mentioned in Sect. 2, it should be very hard, if not impossible, to annotate video content accurately and automatically, especially for those special videos, e.g. the medical videos, where users usually have interests with some regions (or semantic objects) which are impossible for automatic annotation. On the other hand, with the widely spread of home entertainment video equipments, an efficient annotation and summarization strategy is also urgently needed for video management. Hence, the proposed strategies provide a practical solution to acquire content descriptions for those video data.
2. Improved content-based video retrieval efficiency  
Historical content-based video retrieval systems employ either textual word based or visual feature based retrieval. Obviously, by integrating video content annotation and

**Table 6.** Video retrieval results with summaries (Query with summaries at layer 3,  $\alpha = 0.3$ )

Videos	Query 1		Query 2		Query 3		Query 4		Query 5		Query 6		Query 7		Query 8	
Medical videos	2	1	3	2	3	1	4	2	1	2	1	3	3	6	1	3
News program	2	4	5	3	2	6	4	3	7	2	4	2	7	5	4	3
Average	3.18745															

visual features, the retrieval performance could be improved remarkably [49]. Moreover, the query expansion [51] technology could also be integrated to enhance the performance of the video database system.

### 3. Improved remote video database access

The network condition for video transmission is always changing and the transmitted video bit rate is also variable, thus it is very important to support adaptive content delivery and quality of service (QoS) control for online video database system. By utilizing video content description and hierarchical summarization, the video streaming, adaptive transmission and QoS could be directly implemented by considering video content scale and network bandwidth for effective remote video database access.

### 4. Comprehensive video browsing

Browsing has the advantage of keeping the user in the loop during the search process. However, current video retrieval systems do not support efficient browsing because of the lack of an efficient summary organization structure. With the acquired video annotation and hierarchy summaries, both hierarchical browsing and category browsing are easily supported by presentation of multilevel summaries and annotated video database structure.

Obviously, the proposed strategies provide the solutions from video content annotation to summarization. We believe that by integrating those schemes, some more potential applications might be implemented in other multimedia systems.

## 8. Conclusions

In this paper, we have addressed the problem of video content description and summarization for general videos. Due to the unsatisfactory results of video processing techniques in automatically acquiring video content, annotations are still widely used in many applications. Hence, strategies to describe and acquire video content accurately and efficiently must be addressed. We have proposed a content description ontology and a data structure to describe the video content at different levels and with different granularities. A semi-automatic annotation scheme with relevance feedback is implemented by utilizing video group detection, joint semantics and visual features for scene detection, etc. to substantially improve annotation efficiency.

Based on acquired content description data, a hierarchical video summarization scheme has been presented. Unlike other summarization strategies which select important low-level feature related units to build video summaries (since the semantics are not available for them), our method has used the acquired semantic information and visual features among video data to construct a hierarchical structure that describes the video content at various levels. Our proposed strategy con-

siders the content hierarchy, video content description data, and redundancy among videos. With this scheme, the video content can be expressed progressively, from top to bottom in increasing levels of granularity.

Video summaries that only take into account the low-level features of the audio, video or closed-captioned tracks put a great deal of emphasis on the details but not on the content. Video content or structure analysis is necessary prior to video summarization, because the most useful summary may not be just a collection of the most interesting visual information. Hence, our hierarchical summarization strategy achieves a more reasonable result. With this scheme, the video data can be parsed into a hierarchical structure, with each node containing the overview of the video at the current level. As a result, the structure can be widely used for content management, indexing, hierarchical browsing, or other applications.

We can currently explore the possibility of using sequential pattern mining techniques in data mining to automate and enhance video grouping for hierarchical video content description. Since video summarization techniques can be used to construct a summary at any layer, we believe that the hierarchical architecture proposed in this paper can be generalized as a toolkit for video content management and presentation.

*Acknowledgements.* The authors would like to thank the anonymous reviewers for their valuable comments. We would also like to thank Ann C. Catlin for her help in preparing the manuscript. This research has been supported by the *NSF* under grants 0209120-IIS, 0208539-IIS, 0093116-IIS, 9972883-EIA, and by the U.S. Army Research Laboratory and the U.S. Army Research Office under grant number DAAD19-02-1-0178.

## Appendix

We provide the following table of notations used for easy reference by the reader.

$V$	:	a set of video stream.
$S_i$	:	the $i^{\text{th}}$ shot in the video.
$G_i$	:	the $i^{\text{th}}$ group in the video.
$SE_i$	:	the $i^{\text{th}}$ scene in the video.
$CP_i$	:	the $i^{\text{th}}$ clip in the video
$VD$	:	Video Description.
$GD$	:	Group Description.
$SD$	:	Shot Description.
$FD$	:	Frame Description.
$KA$	:	Keyword aggregation of video ontology.
$KAS_i$	:	the union of all keywords which have been shown in describing shot $S_i$ .
$KAG_i$	:	the union of all keywords which have been shown in describing group $G_i$ .
$TDD$	:	Temporal description data, each shot has one $TDD$ .
$TDS$	:	Temporal description stream, each video has one $TDS$ .
$v_{a-b}^{ID}$	:	the region from frame $a$ to frame $b$ in video with certain $ID$ .
$Map(KA, V)$	:	correspondence between annotation ( $KA$ ) and the video temporal information ( $V$ ).
$\overline{VDS}_k$	:	aggregation of keywords which have been used to annotate shot $S_k$ in $VD$ .
$\overline{GDS}_k$	:	aggregation of keywords which have been used to annotate shot $S_k$ in $GD$ .
$\overline{SDS}_k$	:	aggregation of keywords which have been used to annotate shot $S_k$ in $SD$ .
$\overline{FDS}_k$	:	aggregation of keywords which have been used to annotate shot $S_k$ in $FD$ .
$\Omega(X_1, X_2, \dots, X_N)$	:	indicates the union of $X_1, X_2, \dots, X_N$ . $\{X_1 \cup X_2 \cup \dots \cup X_N\}$ .
$\vartheta(X_1, X_2, \dots, X_N)$	:	means the intersection of $X_1, X_2, \dots, X_N$ . $\{X_1 \cap X_2 \cap \dots \cap X_N\}$ .
$\Psi(X)$	:	the number of keyword in $X$ .
$\Theta(X, Y)$	:	the number of overlapped frames between $X$ and $Y$ .
$EF(KA_k, S_i)$	:	normalized factor which keyword $KA_k$ taking effect in shot $S_i$ .
$GDE_i$	:	the aggregation of event descriptor's keyword which has been used in $GD$ of $G_i$ .
$RT_i$	:	representative shot of group $G_i$ .
$RG_i$	:	representative group of scene $SE_i$ .
$StSim(S_i, S_j)$	:	visual features similarity between shot $S_i$ and $S_j$ .
$SemStSim(S_i, S_j)$	:	semantic similarity between shot $S_i$ and $S_j$ .
$ShotSim(S_i, S_j)$	:	Unified similarity between $S_i$ and $S_j$ which integrate visual features and semantics.
$StGpSim(S_i, G_j)$	:	similarity between shot $S_i$ and group $G_j$ .
$GroupSim(G_i, G_j)$	:	similarity between group $G_i$ and $G_j$ .
$SceneSim(SE_i, SE_j)$	:	similarity between scene $SE_i$ and $SE_j$ .

## References

1. Zhang H, Kantankanhalli A, Smoliar S (1993) Automatic partitioning of full-motion video. *ACM Multimedia Syst* 1(1) CE 1 ■
2. Zhang H, Low CY, Smoliar SW, Zhong D (1995) Video parsing, retrieval and browsing: an integrated and content-based solution. *Proceedings ACM Conference on Multimedia*, CE 2 ■
3. Yeung M, Yeo B (197) Video visualization for compact presentation and fast browsing of pictorial content. *IEEE Trans CSVT* 7:771–785
4. Pfeiffer S, Lienhart R, Fischer S, Effelsberg W (1996) Abstracting digital movies automatically. *VCIP* 7(4):345–353
5. Doulamis N, Doulamis A, Avrithis Y, Ntalianis K, Kollias S (2000) Efficient summarization of stereoscopic video sequences. *IEEE Trans CSVT* 10(4) CE 3 ■
6. DeMenthon D, Kobla V, Doermann D (1998) Video summarization by curve simplification. *Proceedings ACM Conference on Multimedia*. Bristol, UK, pp 13–16
7. Uchihashi S, Foote J, Girgensohn A, Boreczky J (1999) Video Managa: generating semantically meaningful video summaries. *Proceedings ACM Conference on Multimedia*, Orlando, FL pp 383–392
8. He L, Sanocki W, Gupta A, Grudin J (1999) Auto-summarization of audio-video presentations. *Proceedings of ACM Conference on Multimedia*. Orlando, FL pp 489–498
9. Ratakonda K, Sezan M, Crinon R (1999) Hierarchical video summarization. *IS&T/SPIE Conference on Visual Communications and Image Processing'99*, San Jose, CA 3653:1531-1541
10. Kim C, Hwang J (2000) An integrated scheme for object-based video abstraction. *Proceedings of ACM Conference on Multimedia Los Angeles, CA* pp 303–311
11. Nam J, Tewfik A (1999) Dynamic video summarization and visualization. *Proceedings of ACM International Conference on Multimedia*, Orlando, FL
12. Lienhart R (1999) Abstracting home video automatically. *Proceedings ACM Multimedia Conference*, CE 4 ■ pp 37–40
13. Christel M, Hauptmann A, Warmack A, Crosby S (1999) Adjustable filmstrips and skims as abstractions for a digital video library. *IEEE Advances in Digital Libraries Conference*, MD, USA
14. Lienhart R, Pfeiffer S, Wffelsberg W (1997) Video abstracting. *Commun ACM* 40(12)
15. Christel M (1999) Visual digest for news video libraries. *Proceedings ACM Multimedia Conference*, CE 5 ■
16. Nack F, Windhouwer M, Hardman L, Pauwels E, Huijberts M (2001) The role of highlevel and lowlevel features in style-based retrieval and generation of multimedia presentation. *New Review of Mypermedia and Multimedia (NRHM) 2001*
17. Venkatesh S, Dorai C (2001) Bridging the semantic gap in content management systems: Computational medial aesthetics. *Proceedings of COSIGN*, Amsterdam, pp 94–99
18. Windhouwer M, Schmidt R, Kersten M (1999) Acoi: A system for indexing multimedia objects. *International Workshop on Information Integration and Web-based Applications & Services*, Indonesia
19. Smoliar S, Zhang H (1994) Content based video indexing and retrieval. *IEEE Multimedia* 1(2):62–72
20. Brunelli R, Mich O, Modena C (1996) A survey on video indexing. *IRST-technical report*
21. Zhong D, Zhang H, Chang S (1997) Clustering methods for video browsing and annotation. *Technical report*, Columbia University, NJ
22. Satoh S, Sato T, Smith M, Nakamura Y, Kanade T CE 6 ■: Naming and detecting faces in News video. *Network-Centric Computing special issue* CE 7 ■
23. Girgensohn A, Foote J (1999) Video classification using transform coefficients. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*, Phoenix, AZ 6:3045–3048
24. Dimitrova N, Elenbaas H, McGee T, Leyvi E, Agnihotri L (2000) An architecture for video content filtering in consumer domain. *International Conference on Information Technology: Coding and computing (ITCC'00)* CE 8 ■
25. Zhou W, Vellaikal A, Kuo CCJ (2001) Rule-based video classification system for basketball video indexing. *Proceedings of ACM International Conference on Multimedia*, Los Angeles, CA
26. Haering N, Qian R, Sezan M CE 9 ■: Detecting hunts in wildlife videos. *Proceedings of the IEEE International Conference on Multimedia Computing and Systems Volume I* CE 10 ■
27. Aguierre Smith T, Davenport G (1992) The Stratification System: A design environment for random access video. *Third International Workshop on Network and Operating System Support for Digital Audio and Video*, CE 11 ■ pp 250–261
28. Weiss R, Duda A, Gifford D (1994) Content-based access to algebraic video. *IEEE International Conference on Multimedia Computing and Systems*, Boston, MA, pp 140–151
29. Davenport G, Murtaugh M (1995) Context: towards the evolving documentary. *Proceedings of ACM Multimedia Conference*, San Francisco, CA
30. Davis M (1993) Media streams: An iconic visual language for video annotation. *IEEE Symposium on Visual Language*, pp 196–202
31. Petkovic M, Jonker W (2000) An overview of data models and query languages for content-based video retrieval. *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, Italy
32. Kender J, Yeo B (1998) Video scene segmentation via continuous video coherence. *Proceedings of CVPR* CE 12 ■
33. Jiang H, Montesi D, Elmagarmid AK (1997) Video text database systems. *Proceedings of IEEE Multimedia Systems*, Ottawa, Canada
34. Ponceleon D, Dieberger A (2001) Hierarchical brushing in a collection of video data. *Proceedings of the 34th Hawaii International Conference on System Sciences*, CE 13 ■
35. Luke S, Spector L, Rager D (1996) Ontology-based knowledge discovery on the world-wide web. *Proceedings of the Workshop on Internet-based Information Systems, AAAI-96*, Portland, OR
36. Miller G (1995) Wordnet: A lexical database for English. *Commun ACM* 38(11)
37. Mena E, Keshyap V, Illarramendi A, Sheth A (1998) Domain specific ontologies for semantic information brokering on the global information infrastructure. *Proceedings of FOIS'98* CE 14 ■
38. Bloch G (1988) From concepts to film sequences. *Proceedings of RIAO*, Cambridge, MA, pp 760–767
39. Parkes A (1992) Computer-controlled video for intelligent interactive use: a description methodology. In: ADN Edwards, S Holland (eds) *Multimedia Interface Design in Education*. New York
40. Rui Y, Huang T, Mehrotra S (1999) Constructing table-of-content for video. *ACM Multimedia Syst J* 7(5) 359–368

41. Aref W, Elmagarmid A, Fan J, Guo J, Hammad M, Ilyas I, Marzouk M, Prabhakar S, Rezgui A, Teoh A, Terzi E, Tu Y, Vakali A, Zhu X (2002) A distributed database server for continuous media. Proceedings of IEEE 18th ICDE demonstration, San Jose, CA
42. Yeo B, Liu B (1995) Rapid scene analysis on compressed video. IEEE Trans CSVT 5(6)
43. Fan J, Aref W, Elmagarmid A, Hacid M, Marzouk M, Zhu X (2001) MultiView: Multilevel video content representation and retrieval. J Electr Imag 10(4):895–908
44. Yeung M, Yeo B (1996) Time-constrained clustering for segmentation of video into story units. Proceedings of ICPR'96
45. Lin T, Zhang H (2000) Automatic video scene extraction by shot grouping. Proceedings of ICPR 2000,  CE 15
46. Fan J, Yu J, Fujita G, Onoye T, Wu L, Shirakawa I (2001) Spatiotemporal segmentation for compact video representation. Signal Process: Image Commun 16:553–566
47. Zhu X, Liu W, Zhang H, Wu L (2001) An image retrieval and semi-automatic annotation scheme for large image databases on the Web. IS&T/ 4311:168–177
48. Lu Y, Hu C, Zhu X, Zhang H, Yang Q (2000) A unified semantics and feature based image retrieval technique using relevance feedback. Proceedings of ACM Multimedia Conference,  CA,  CE 17
49. Zhu X, Zhang H, Liu W, Hu C, Wu L (2001) A new query refinement and semantics integrated image retrieval system with semi-automatic annotation scheme. J Electr Imag – Special Issue on Storage, Processing and Retrieval of Digital Media 10 (4):850–860
50. Vasconcelos N, Lippman A (1998) A Bayesian framework for content-based indexing and retrieval. Proceedings of DCC'98, Snowbird, UT  CE 18
51. Fan J, Zhu X, Wu L (2001) Automatic model-based semantic object extraction algorithm. IEEE Trans Circuits and Systems for Video Technology 11(10)10: 1073–1084
52. Zhu X, Fan J, Elmagarmid A, Aref W (2002) Hierarchical video summarization for medical data. SPIE: Storage and Retrieval for Media Databases 4676:395–406
53. Hjelsvold R, Midtstraum R (1994) Modelling and querying video data. Proceedings of VLDB Conference  CE 19
54. Schreiber ATH, Dubbeldam B, Wielemaker J, Wielinga B (2001) Ontology-based photo annotation. IEEE Intell Syst 16(3):66–74
55. Gruber T (1992) Ontolingua: A mechanism to support portable ontologies. Technical Report KSL-91-66, Knowledge Systems Laboratory, Stanford University, Palo Alto, CA
56. Voorhees E (1994) Query expansion using lexical-semantic relations. Proceedings 17th International Conference on Research and Development in Information Retrieval (ACM SIGIR),  CE 20