Zhang, Yi and Tat-Seng Chua “Video Modeling and Retrieval”
*Multimedia Image and Video Processing*
Ed. Ling Guan et al.
Boca Raton: CRC Press LLC, 2001
Chapter 13

Video Modeling and Retrieval

Yi Zhang and Tat-Seng Chua

13.1 Introduction

Video is the most effective media for capturing the world around us. By combining audio and visual effects, it achieves a very high degree of reality. With the widely accepted MPEG (Moving Pictures Expert Group) [12] digital video standard and low-cost hardware support, digital video has gained popularity in all aspects of life.

Video plays an important role in entertainment, education, and training. The term “video” is used extensively in the industry to represent all audiovisual recording and playback technologies. Video has been the primary concern of the movie and television industry. Over the years, that industry has developed detailed and complete procedures and techniques to index, store, edit, retrieve, sequence, and present video materials. The techniques, however, are mostly manual in nature and are designed mainly to support human experts in creative moviemaking. They are not set up to deal with the large quantity of video materials available. To manage these video materials effectively, it is necessary to develop automated techniques to model and manage large quantities of videos.

Conceptually, the video retrieval system should act like a library system for the users. In the library, books are cataloged and placed on bookshelves according to well-defined classification structures and procedures. All the particular information about a book, such as the subject area, keywords, authors, ISBN number, etc., are stored to facilitate subsequent retrievals. At the higher level, the classification structure acts like a conceptual map in helping users to browse and locate related books. Video materials should be modeled and stored in a similar way for effective retrieval. A combination of techniques developed in the library and movie industries should be used to manage and present large quantities of videos.

This chapter discusses the modeling of video, which is the representation of video contents and the corresponding contextual information in the form of a conceptual map. We also cover the indexing and organization of video databases. In particular, we describe the development of a system that can support the whole process of logging, indexing, retrieval, and virtual editing of video materials.
13.2 Modeling and Representation of Video: Segmentation vs. Stratification

There are many interesting characters, events, objects, and actions contained in a typical video. The main purpose of video modeling is to capture such information for effective representation and retrieval of video clips. There are two major approaches to modeling video, namely, the segmentation [5, 15] and stratification [1] approaches.

In the segmentation approach, a video sequence is segmented into physical chunks called shots. By definition, a video shot is the smallest sequence of frames that possesses a simple and self-contained concept. In practice, video shots are identified and segmented using scene change detection techniques [21]. A change in scene will mark the end of the previous shot and the start of the next one. Figure 13.1 shows the way in which a video sequence is segmented into shots.

![Segmented video shots](image)

**FIGURE 13.1**
Segmentation: modeling and representation of video shots.

In the segmentation approach, a video shot is an atomic unit that can be manipulated for representation and retrieval. Figure 13.1 also shows how a video shot can be represented. Important attributes such as title, description, audio dialogues, color histogram, and visual objects should be extracted and logged. In addition, we can capture cinematic information [8], such as the focal length, scope, and angle of shots, for the purpose of sequencing video shots for presentation. By combining all the attributes mentioned above, a good representation of video shots can be achieved.

Although the segmentation approach is easy to understand and implement, it has several drawbacks. In such a model, the granularity of the information is a shot which typically contains a self-contained but high-level concept. Once the video is segmented, the shot boundaries are fixed and it is not possible for users to access or present video contents within the shot boundaries. This inflexibility limits the users’ ability to model more complicated events in a way not originally intended by the authors. Because of the diverse contents and meaning inherited in the video, certain events are partially lost. For example, a shot may contain multiple characters and actions; it is thus not possible for authors to anticipate all the users’ needs and log all interesting events. Moreover, the segment boundaries may be event dependent. This may result in a common story being fragmented across multiple shots, which will cause discontinuity to users looking for the story. For these reasons, automated description of video contents at the shot level is generally not possible.

To overcome this problem, we investigate an object-based video indexing scheme to model the content of video based on the occurrences of simple objects and events (known as entities).
This is similar to the stratification approach proposed in [1]. In this scheme, an entity can be a concept, object, event, category, or dialogue that is of interest to the users. The entities may occur over many, possibly overlapping, time periods. A strand of an entity and its occurrences over the video stream are known as a stratum. Figure 13.2 shows the modeling of a news video with strata such as the object “Anchor-Person-A”; categories like “home-news,” “international news,” “live reporting,” “finance,” “weather,” and “sports”; and audio dialogues. The meaning of the video at any time instance is simply the union of the entities occurring during that time, together with the available dialogue.

Object-based modeling of video has several advantages. First, because strata tend to contain only simple entities, the pseudo-object models together with relevance feedback (RF) [7] and other learning methods may be adopted to identify and track entities automatically. Second, the meaning of video at any instance is simply the union of strata occurring at that time. Thus the system can flexibly compose portions of video sequences whose meanings closely match that of the query. Finally, the strata information provides the meta-information for the video. Such information can be used to support innovative functions such as the content-based fast-forwarding and summarization [7] of video.

13.2.1 Practical Considerations

The main advantage of the stratification approach over segmentation is that it is possible to automate the process of indexing. However, when the project was started several years ago, this advantage was not realizable because of the lack of content-based analysis tools for automated indexing. Without such tools, it is extremely tedious to index video content using the stratification approach. Because of this, coupled with the simplicity of the segmentation approach in supporting our initial research goals in video retrieval, virtual editing, and summarization, we adopted the segmentation approach. This project is an extension of the work done in Chua and Ruan [5] using digital video with more advanced functionalities for video manipulation, browsing, and retrieval. We are currently working on the development of a system based on the stratification approach to model video in the news domain.

This chapter describes the design and implementation of our segmentation model for video retrieval. The domain of application is documentary videos.
13.3 Design of a Video Retrieval System

This section considers the modeling of video using the segmentation approach. There are several processes involved in modeling video sequences to support retrieval and browsing. The two main steps are video parsing and indexing. The term parsing refers to the process of segmenting and logging video content. It consists of three tasks: (1) temporal segmentation of video material into elementary units called segments or shots; (2) extraction of content from these segments; and (3) modeling of context in the form of a concept hierarchy. The indexing process supports the storage of extracted segments, together with their contents and context, in the database. Retrieval and browsing rely on effective parsing and indexing of raw video materials. Figure 13.3 summarizes the whole process of video indexing, retrieval, and browsing.

13.3.1 Video Segmentation

The success of the segmentation approach depends largely on how well the video materials are divided into segments or shots. This involves identifying suitable criteria to decide which points in a video sequence constitute segment boundaries. Manual segmentation has been done effectively in the movie industry. However, manual segmentation is time consuming and prone to error. Moreover, its results are biased toward the authors’ intents.

With advances in image processing techniques, computer-aided segmentation is now possible [21]. The main objective is to detect the joining of two shots in the video sequence and
locate the exact position of these joins. These joins are made by the video editing process, and they can be of the following types based on the techniques involved in the editing process:

- Abrupt cut
- Dissolve
- Fade in/fade out
- Curtain and circle

In an abrupt cut, the video editor does nothing but simply concatenate the two shots together. The other three joins are generally known as gradual transitions, in which the video editor uses a special technique to make the join appear visually smooth.

Thus, segmentation becomes a matter of finding those features that constitute transitions. A lot of work has been done on detecting abrupt cuts, and many techniques have been developed to handle gradual transitions in both the compressed and the decompressed domain [11, 13, 19]. We detect the cut transition by sequentially measuring successive inter-frame differences based on features such as the color histogram. When the difference is above the global threshold, a cut transition is declared.

When dealing with gradual transitions, Zhang et al. [21] used two thresholds. They computed accumulated differences of successive frames when the inter-frame difference was above a lower threshold. When this accumulated difference exceeded the high threshold, a gradual transition was declared. Other approaches employ template matching [20], model-based [10], statistical [18], and feature-based [19] methods. Most of the methods employed for detecting gradual transitions require careful selection of the threshold for the method to work effectively [11]. This is a difficult problem. Lin et al. [14] proposed a multi-resolution temporal analysis approach based on wavelet theory to detect all transitions in a consistent manner.

In this chapter, we employ the Zhang et al. [21] approach of computing accumulated differences of color histogram in successive frames to detect both abrupt cut and gradual transitions. We employ this approach because it is simple to implement and has been found to work well. Our tests show that it could achieve a segmentation accuracy of greater than 80%. Based on the set of segments created, a visual interface is designed to permit the authors to review and fine-tune the segment boundaries. The interface of the Shot Editor is given in Figure 13.9. The resulting computer-aided approach has been found to be satisfactory.

### 13.3.2 Logging of Shots

After the video is segmented, each shot is logged by analyzing its contents. Logging is the process of assigning meanings to the shots. Typically, each shot is manually assigned a title and text description, which are used in most text-based video retrieval systems. Increasingly, speech recognition tools are being used to extract text from audio dialogues [16], and content-based analysis tools are being used to extract pseudo-objects from visual contents [6]. These data are logged as shown in Figure 13.1. The combination of textual and content information enables different facets of content to be modeled. This permits more accurate retrieval of the shots.

Because video is a temporal medium, in addition to retrieving the correct shots based on the query, the shots retrieved must be properly sequenced for presentation. Thus, the result of a query should be a dynamically composed video sequence, rather than a list of shots. To do this, we need to capture cinematic information [8]. Typical information that is useful for sequencing purposes includes the focal length and angle of shots. Such information permits the sequencing of video by gradually showing the details (from far shots to close-up shots), a
typical presentation technique used in documentary video. The complete information logged for the shot thus includes both content and cinematic information, as shown in Figure 13.1.

Video is a temporal medium that contains a large amount of other time-related information. This includes relationships between objects and motion, the order in which main characters appear, and camera motions/operations. However, there is still a lack of understanding of how object motions are to be queried by users. Also, automated detection of motion and object relationships is not feasible with the current technology. These topics are open for research and exploration. Thus, temporal information is not included as part of the shot representation.

13.3.3 Modeling the Context between Video Shots

In the segmentation approach, although the information logged provides a good representation for individual video shots, the context information between video shots is not properly represented. Without contextual information, it is hard to deduce the relationships between shots. For example, in the domain of an animal, the contextual information permits different aspects of the life of an animal, say a lynx, to be linked together (see Figure 13.10). It also provides high-level information such as the knowledge that lion and lynx belong to the same cat family. The context information can be captured in the structured modeling approach using a two-layered model as shown in Figure 13.4.

![Figure 13.4](image)

A two-layered model for representing video context.

The layered model consists of a lower shot layer and the scene layer. The shot layer contains all the shots indexed, whereas the scene layer models the video context information. The context information can be modeled as a scene hierarchy as shown in Figure 13.4. Maintaining the context information only at the scene layer permits more than one scene hierarchy to be modeled above the same set of shots. This facilitates multiple interpretation of the shot layer and leads to a more flexible creation of scenes.

Typical information encoded at the scene includes the title and textual descriptions, together with parent–child relations. Because the scene hierarchy captures the knowledge of the set of video shots, it can be used to support concept-based retrieval of the video shots. A retrieval of a set of video shots can be viewed as the retrieval of an appropriate scene (or concept) in the scene hierarchy. This will be illustrated in a later section.
13.4 Retrieval and Virtual Editing of Video

13.4.1 Video Shot Retrieval

The shots and scenes permit the video to be represented in a form that can be stored and manipulated by the computer. Because the main semantic information captured for the shot is the title, text descriptions, and dialogues, we store these as free text so that a free-text retrieval technique can be employed to retrieve video shots using free-text queries. Conceptually, each shot is stored as a free-text document in an inverted file. We employ the vector space information retrieval (IR) model [17] to retrieve the shots. In the vector space IR model, each shot is represented as a vector of dimension \( t \) of the form:

\[
S_i = (d_{i1} d_{i2} d_{i3} \ldots d_{it})
\]

(13.1)

where \( d_{ij} \) is the weight of term \( j \) in the \( i \)th shot \( S_i \), and \( t \) is the number of text terms used. By representing the query \( Q \) in the same way, that is,

\[
Q = (q_1 q_2 q_3 \ldots q_t)
\]

(13.2)

the similarity between query \( Q \) and shot \( S_i \) can be computed using the cosine similarity formula [17], given by:

\[
Sim(S_i, Q) = \frac{\sum_{k=1}^{t} (d_{ik} \times q_k)}{\sqrt{\sum_{k=1}^{t} (d_{ik})^2} \times \sqrt{\sum_{k=1}^{t} (q_k)^2}}
\]

(13.3)

Equation (13.3) is used as the basis to rank all shots with respect to the query.

13.4.2 Scene Association Retrieval

To permit more effective retrieval of shots, the knowledge built into the scene structure should be used. The scene is used to link shots semantically related to each other under the same scene hierarchy. Many of these shots may not have common text descriptions and thus may not be retrievable together at the same time. To improve higher retrieval accuracy, the idea here is to retrieve higher level scenes as much as possible so that these related shots may be retrieved by the same query. A scene association algorithm is developed for this purpose. The algorithm is best illustrated using Figure 13.5.

After the user issues a query, the scene association algorithm proceeds as follows:

1. **Shot retrieval**: First we compute the similarities between the query and all the shots in the shot layer using equation (13.3). The shots are ranked based on the similarity values assigned to them. The list of shots whose similarity values are above a predefined threshold is considered to be relevant to the query. This is known as the initial relevant shot list. Shots that appear on the initial relevant shot list are assigned a value of 1; all other shots are assigned a value of 0. In the example given in Figure 13.5, shots \( S_1, S_3, S_4, S_7, \) and \( S_8 \) are given a value of 1 because they are on the initial relevant shot list. Shots \( S_2, S_5, S_6, \) and \( S_9 \) are assigned a value of 0 because they are not.

2. **Association**: We then compute a similarity value for each leaf node in the scene structure. This is done by computing the percentage of its children that are assigned the value of 1. For example, scene leaf node \( C_4 \) is given a value of 1/3.
3. **Propagation:** The similarity values of the scene leaf nodes are then propagated up the scene hierarchy. Each parent node is given a value equal to the average of the similarity values of its child nodes. For example, $C_1$ is assigned a value that is the average of the similarity values of all its child nodes $C_2, C_3,$ and $C_5$.

4. **Selection:** At the end of propagation, we select the scene node with the highest value as the most relevant scene. If the similarity value of this scene node exceeds a predefined threshold $T_s$, then all shots under this scene together with the initial relevant shot list are returned. Otherwise, only the initial relevant shot list will be returned.

In Figure 13.5, video shots $S_1, S_3, S_4, S_7,$ and $S_8$ are selected in the initial relevant shot list. After the scene association step, scene $C_2$ is selected as the most relevant scene. All the shots under scene $C_2$ and those in the initial relevant shot list are retrieved and returned to the users. Notice that by using scene association, shot $S_2$, which is not in the initial relevant shot list, can also be retrieved. Thus, by giving higher priority to retrieving higher level scenes, we are able to retrieve scenes that represent more general concepts and cover a greater number of scenes and shots. This will result in better recall in video shot retrieval.

### 13.4.3 Virtual Editing

Because video is a temporal medium, it has the additional problem of sequencing the retrieved video shots before presentation. In the computer industry, the automatic creation of a video sequence from video clips is called **virtual editing** [3]. Cinematic rules are used as the basis to sequence the retrieved video shots in order to form a meaningful sequence. This process is more art than science.

#### Parameters for Virtual Editing

Before virtual editing can be carried out, a set of cinematic parameters must be recorded at indexing time. Normally, the parameters logged include:
• **Focal length**

ELS — extreme long shot  
LS — long shot  
DFS — deep focus shot  
MLS — medium long shot  
MS — medium shot  
CU — close-up shot  
ECU — extreme close-up shot  

• **Type of scope**

1-shot — only one main character in the shot  
2-shot  
3-shot  
group-shot — many characters in the shot  

• **Type of angle**

Bird view  
Eye-level angle  
High angle  
Low angle  
Oblique angle  

• **Type of spatial**

Inclusion  
Exclusion  

• **Type of movement**

Dolly  
Hand-held  
Pan  
Tilt  
Zoom  
  
  Zoom in  
  Zoom out  

These parameters are sufficient to generate satisfactory documentary-style movies. In fact, the “focal length,” “type of angle,” and “type of movement” should be enough to generate most typical presentation sequences that start from a general view before zooming into the details or vice versa. Such styles are effective in presenting information.

**Cinematic Rules for Virtual Editing**

Cinematic rules have existed since the birth of film-making [2, 9]. These rules are widely accepted by film editors and have been applied to generate a tremendous number of films. Generally speaking, there are two classes of rules:
1. Basic rules that are common sense based

2. Complex rules that are normally subject matter based and rely on factors such as psychology, aesthetics, and so forth.

The rules provide only general guidelines for how a movie can be made. The making of a memorable film relies largely on the editor’s creativity in applying the rules. Because there is a general lack of understanding of how films are edited and sequenced, it is hard to automate the creative aspects of film-making. Thus, for practical reasons, only the simple and easy-to-understand rules will be automated. The rules that are commonly studied and implemented are:

- **Concentration rule:** An actor, object, or topic is introduced in the sequence from long shots, to medium shots, to close-up shots. The relationships between the objects and the environment are also evolved through this sequence.

- **Enlarge rule:** This is the inverse of the concentration rule.

- **General rule:** This is a combination of the concentration and enlarge rules. It intends to present an intact action in a sequence that supports better understanding.

- **Parallel rule:** This aims to present two different themes alternately. Normally, the two themes are similar to or contrast with each other. Examples of such themes include the alternating shots of two actors walking toward each other, or the typical chase scenes of police and villains.

- **Rhythm rule:** The shots are chosen based on the rhythm requested. Longer duration shots are used for slower rhythms, whereas shorter duration shots are used for faster rhythms. The rhythm of the intra-shot action should be matched with that of the presentation.

- **Sequential rule:** Shots are ordered chronologically.

- **Content rule:** Shots in a sequence share common content attributes.

### From Concept to Video Sequence

To generate a video sequence from a concept, the user needs to specify not just the query but also presentation information such as the time constraint, the cinematic rule to apply, and the necessary parameters for the cinematic rule. This can be achieved by using a script, as shown in Figure 13.6. The script, as a template, is used to record necessary information for sequencing purposes.

<table>
<thead>
<tr>
<th>Script title: lynx family</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cinematic rule: sequential</td>
</tr>
<tr>
<td>Time duration: 20</td>
</tr>
<tr>
<td>Query: lynx family together</td>
</tr>
<tr>
<td>Scene (optional): lynx</td>
</tr>
</tbody>
</table>

**FIGURE 13.6**
An example of a script for virtual editing.

A general script class is defined. A specific script for each cinematic rule is implemented. Each script has its own method to sequence the retrieved video shots. The most important cinematic parameter used is the focal length, which is recorded for the main character. It
is used by the script to generate effects of the concentration rule, enlarge rule, and general rule. Video shots are sorted based on this parameter to form the appropriate presentation sequence. Other parameters such as the number of frames help in restricting the duration of the presentation.

Given a script, the following steps are followed to generate a meaningful video sequence:

1. User defines the script, which contains all necessary information for sequencing, including script title, cinematic rule, query terms, time duration, etc.
2. The query is first used to retrieve the relevant video shots using, if appropriate, the scene association method.
3. The shots retrieved are ordered using the cinematic parameter(s) determined by the cinematic rule chosen. For example, if the enlarge rule is chosen, the shots are ordered from long shots to close-up shots.
4. The sequence is formed by selecting a suitable number of shots in each cinematic category, subject to the time limit.
5. Before presentation, user may edit the sequence by adding new shots and/or cutting shots from the sequence.
6. When the result is satisfactory, the final sequence is presented to the user.

13.5 Implementation

This section describes the implementation of the digital video retrieval system. The overall design of the system is given in Figure 13.7. The three major components of the system are:

- **Shot Editor**: Caters to the indexing and logging of video shots.
- **Scene Composer**: Supports the creation and editing of scenes. It uses a retrieval tool to help users semiautomate the creation of scenes by issuing queries. A subset of this tool supports the browsing of scenes by end users.
- **Video Presentor**: Supports the retrieval and virtual editing of video materials.

Figure 13.7 also shows the interactions between the users and the system, and among different components of the system. The system is designed to support two types of users: (1) high-level users or indexers who will use the sophisticated tools to index and edit video shots and scenes, and (2) normal end users who will interact only with the Scene Browser to browse through the scenes, and with the Video Presentor to query and sequence video shots. The functionalities provided to these two types of users are quite different. Figure 13.8 gives the main user interface and the subset of functionality provided to the indexers and end users. The specific interfaces and functions of the components are described separately below.

**Shot Editor**

The Shot Editor is a tool that helps users index and log the raw video shots. Figure 13.9 shows the interface of the Shot Editor. The video is viewed on the top-left window, which provides a VCR-like interface for users to browse the contents of the shots and/or to fine-tune
FIGURE 13.7
System components of the digital video retrieval system.

the shot boundaries. Users may log or edit the semantic information for the shots, such as the title, text descriptions, and dialogues, and cinematic parameters. The list of shots created is shown in the top-right list browser.

Scene Composer and Browser

The Scene Composer is used to compose scenes that capture the context information and domain knowledge of the video shots. Through the interface given in Figure 13.10, users may create a new scene by clicking on the “New” button at the top menu bar. The edit panel at the bottom permits users to view the list of existing shots and scenes that can be used as members of the new scene. Because the possible set of shots and scenes may be large, we may limit the set by using the shot retrieval and scene association modules (see Section 13.3) to filter out only those that match the content of the new scene being created. This facilitates the task of composing complex scenes and opens up the possibility of creating some of the scenes automatically. Users may browse and edit existing scenes at any time. Users may also view the video sequence generated from the scene by clicking on the “Play” button.

A simpler interface, without all the editing functions, is provided to end users to browse the content of the scenes. Because the scene structures provide rich context information and domain knowledge of the underlying interface, it is extremely beneficial for new users to have a good understanding of the overall structure before accessing the video database.

Video Presentor

The Video Presentor is the main interface users interact with to perform video retrieval and virtual editing. Figure 13.11 shows the interface together with its child windows. To retrieve
the video sequence, users need to pose a free-text query. Users may optionally enter a time limit as well as cinematic rules and parameters. Based on the query script, the system first retrieves a relevant set of shots (given in the Retrieved List Browser) by using only the scene association retrieval function. It then performs the virtual editing function to arrive at a final sequence (listed in the Sequenced List Browser) that meets the users’ presentation specifications. Users may view the generated sequence automatically by manipulating the VCR-like interface at the top-left window. Users may also choose to view each sequenced shot individually or edit the sequence manually. The final sequence may be saved for future viewing.

Instead of issuing a query, users may choose to view an existing sequence generated previously.

13.6 Testing and Results

The system was developed at the Multimedia Information Laboratory at the National University of Singapore. It was implemented on the Sun Solaris using C++ and employed the MPEG-TV tool to manipulate and display digital video in MPEG-1. We chose a documentary
video in the domain of animal for testing. We selected more than 24 minutes of video materials in 7 different animal categories. The video was analyzed and logged using the procedures as outlined in Section 13.3. Altogether, 164 video shots and 45 scenes were created. A summary of the video shots created in different categories is given in Table 13.1.

![User interface of the Shot Editor.](image)

**FIGURE 13.9**
User interface of the Shot Editor.

**Table 13.1** Characteristics of Video Materials Used

<table>
<thead>
<tr>
<th>Category</th>
<th>Total Duration (s)</th>
<th>Number of Shots Created</th>
<th>Number of Scenes Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Koala</td>
<td>219.9</td>
<td>33</td>
<td>8</td>
</tr>
<tr>
<td>Lion</td>
<td>222.9</td>
<td>27</td>
<td>7</td>
</tr>
<tr>
<td>Lynx</td>
<td>213.2</td>
<td>22</td>
<td>7</td>
</tr>
<tr>
<td>Manatee</td>
<td>184.4</td>
<td>16</td>
<td>4</td>
</tr>
<tr>
<td>Manta ray</td>
<td>242.2</td>
<td>16</td>
<td>5</td>
</tr>
<tr>
<td>Marine turtle</td>
<td>225.7</td>
<td>23</td>
<td>6</td>
</tr>
<tr>
<td>Ostrich</td>
<td>205.7</td>
<td>27</td>
<td>8</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td><strong>1447.2</strong></td>
<td><strong>164</strong></td>
<td><strong>45</strong></td>
</tr>
</tbody>
</table>
A set of seven queries was chosen to test the retrieval effectiveness. The queries, together with the list of relevant shots, are summarized in Table 13.2. The results of retrieval, using the shot retrieval technique, are presented in Table 13.3 in terms of normalized recall and precision. From Table 13.3, it can be observed that the shot retrieval technique is quite effective. Further tests using the scene association technique demonstrated that we could achieve a further 10 to 12% improvement in retrieval performance.

To evaluate the effectiveness of virtual editing, we conducted a series of retrievals using similar queries but with different cinematic rules. The results are shown in Figure 13.12. Figures 13.12a–c show the results of issuing the query “koala sleeps” to retrieve the set of relevant video shots using the scene association technique but applying different cinematic rules to sequence the shots. The time limit chosen is 25 seconds in all three cases. The sequences generated using concentration, enlarge, and general rules are shown in Figures 13.12a–c, respectively. The final sequence shown in Figure 13.12d is generated by using the query “koala scratches” with the parallel cinematic rule and longer duration.

From the results (Figures 13.12a–c), it can be seen that our system is able to generate sequences that conform to the desired cinematic rules within the time limit. Figure 13.12(d) also shows that the system is able to generate complicated sequences based on parallel themes. These results clearly meet the expectations of users and demonstrate that both our retrieval and virtual editing functions are effective.
Query terms: the user can state what he wants.
Time duration: limits the presentation time.

FIGURE 13.11
User interface of the Video Presentor.
FIGURE 13.12
Results of virtual editing using the same query but different cinematic rules. (a) query: koala sleeps; rule: Concentration; duration: 25 sec (b) query: koala sleeps; rule: Enlarge; duration: 25 sec (c) query: koala sleeps; rule: General; duration: 25 sec (d) query: koala scratches itches; rule: Parallel; Theme1: body; Theme2: head; duration: 30 sec.

13.7 Conclusion

This chapter discussed the modeling, representation, indexing, retrieval, and presentation of digital video. It also reviewed the design and implementation of a digital video retrieval system to accomplish the above task. The effectiveness of the system was demonstrated using a documentary video in the domain of animals. The results showed that virtual editing can be carried out effectively using simple cinematic rules.

With advances in automated (pseudo) object identification and transcription of audio, we are beginning to research ways to automate the process of logging video shots. We are also developing a new system based on the stratification approach, centered on advanced functionalities for retrieval, interaction, semantic-based fast-forwarding, and video summarization.
Table 13.2 Queries and Set of Relevant Video Shots

<table>
<thead>
<tr>
<th>Query</th>
<th>Relevant Video Shots</th>
</tr>
</thead>
<tbody>
<tr>
<td>Koala on a tree</td>
<td>koala001.mpg.txt, koala003.mpg.txt, koala007.mpg.txt, koala008.mpg.txt, koala009.mpg.txt, koala010.mpg.txt, koala015.mpg.txt, koala021.mpg.txt, koala022.mpg.txt, koala024.mpg.txt</td>
</tr>
<tr>
<td>Lion walking</td>
<td>lion007.mpg.txt, lion009.mpg.txt, lion010.mpg.txt</td>
</tr>
<tr>
<td>Lynx hunting</td>
<td>lynx013.mpg.txt, lynx014.mpg.txt, lynx015.mpg.txt, lynx016.mpg.txt</td>
</tr>
<tr>
<td>Manatee family</td>
<td>manatee008.mpg.txt, manatee009.mpg.txt, manatee010.mpg.txt, manatee011.mpg.txt, manatee013.mpg.txt</td>
</tr>
<tr>
<td>Manta ray flying</td>
<td>manta_ray001.mpg.txt, manta_ray010.mpg.txt, manta_ray012.mpg.txt</td>
</tr>
<tr>
<td>Marine turtle giving birth</td>
<td>marine_t012.mpg.txt, marine_t013.mpg.txt, marine_t014.mpg.txt, marine_t015.mpg.txt, marine_t016.mpg.txt, marine_t018.mpg.txt, marine_t019.mpg.txt</td>
</tr>
<tr>
<td>Ostrich running</td>
<td>ostrich003.mpg.txt, ostrich014.mpg.txt, ostrich015.mpg.txt</td>
</tr>
</tbody>
</table>

Table 13.3 Normalized Recall and Precision of Retrieval

<table>
<thead>
<tr>
<th>Query</th>
<th>Recall (normalized)</th>
<th>Precision (normalized)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Koala on a tree</td>
<td>0.816388</td>
<td>0.662276</td>
</tr>
<tr>
<td>Lion walking</td>
<td>0.798734</td>
<td>0.616213</td>
</tr>
<tr>
<td>Lynx hunting</td>
<td>0.882279</td>
<td>0.655542</td>
</tr>
<tr>
<td>Manatee family</td>
<td>0.847985</td>
<td>0.668782</td>
</tr>
<tr>
<td>Manta ray flying</td>
<td>0.871069</td>
<td>0.743911</td>
</tr>
<tr>
<td>Marine turtle giving birth</td>
<td>0.692742</td>
<td>0.514673</td>
</tr>
<tr>
<td>Ostrich running</td>
<td>0.753145</td>
<td>0.592238</td>
</tr>
<tr>
<td>Average</td>
<td>0.808906</td>
<td>0.636234</td>
</tr>
</tbody>
</table>

References


