Video query: Research directions

by R. M. Bolle
B.-L. Yeo
M. M. Yeung

As digital video databases become more and more pervasive, finding video in large databases becomes a major problem. Because of the nature of video (streamed objects), accessing the content of such databases is inherently a time-consuming operation. Enabling intelligent means of video retrieval and rapid video viewing through the processing, analysis, and interpretation of visual content are, therefore, important topics of research. In this paper, we survey the art of video query and retrieval and propose a framework for video-query formulation and video retrieval based on an iterated sequence of *navigating*, *searching*, *browsing*, and *viewing*. We describe how the rich information media of video in the forms of image, audio, and text can be appropriately used in each stage of the search process to retrieve relevant segments. Also, we address the problem of automatic video annotation—attaching meanings to video segments to aid the query steps. Subsequently, we present a novel framework of structural video analysis that focuses on the processing of high-level features as well as low-level visual cues. This processing augments the semantic interpretation of a wide variety of long video segments and assists in the search, navigation, and retrieval of video. We describe several such techniques.

1. Introduction

More and more video is generated every day. Today, much of this data is produced and stored in some analog form such as VHS video or motion pictures. But the trend is toward total digitization of film and video, and with the arrival of cheaper digital-storage devices, it becomes economically feasible to digitize video data and store and transmit it in some sort of digital form. Eventually, all storage and transport mechanisms to television receivers and computer displays will be dominated by digital technologies [1]. These technologies include CD-ROM, video tape recorders, telecommunication networks, cable, and terrestrial and satellite transmission.

The digital form allows processing of the video data to generate appropriate data abstractions that permit flexible video-database organization and enable content-based retrieval of video. That is, very much as today's large text databases can be searched with text queries, video databases will be able to be searched with combined text and visual queries. Video clips, possibly very short, will be retrieved from longer sequences in large databases on the basis of some sort of organization of the time-oriented structure of the video, and, more interestingly, on the basis of the semantic video content. For the latter case, an example of video content is the presence of an object (i.e., visible item) in the video. In addition to being static, the semantic content of video can be dynamic or motion-based (e.g., the appearance or disappearance of objects, and actions or interactions of objects).

Ideally, the video will be automatically annotated as a result of machine interpretation of the semantic content of

0018-8646/98/$5.00 © 1998 IBM

the video; however, given the state of the art in computer vision, such sophisticated data abstractions may not be feasible in practice. Rather, the computer may offer intelligent assistance in the manual annotation of video, or the computer may perform automatic annotation with limited semantic interpretation.

In order to create the video data abstraction, it is desirable to identify syntactic and semantic components in the video material and to define data models for concisely describing first the structural video properties and second the semantic video content. The data models for representing video have to be general and broad enough to accommodate the range of formats and lengths of different types of programs, whether the video is a two-hour movie, a one-hour talk-show, a five-minute home-video clip, or a thirty-second news segment.

The area of video on demand (VOD) has addressed the issue of efficient video retrieval and rapid video viewing for several years now. The emphasis of VOD is on mechanisms for accessing video data, typically stored on tertiary storage devices, where the amounts of data can be quite enormous. Rather than linear viewing of video, VOD systems should provide the ability to filter out unwanted information (e.g., commercials) while retrieving video of interest.

Rowe et al. [2], by surveying a variety of users of multimedia database systems, characterized the types of video queries they needed, and identified the following "indexes" that should be associated with the video data in order to satisfy the queries:

1. Bibliographic data. This category includes information about the entire video (e.g., title, abstract, subject, and genre) and the individuals involved with the production of the video (e.g., producer, director, and cast), i.e., traditional *metadata* (data about the data).
2. Structure data. Video (and film) can be described by the hierarchy *movie*, *segment*, *scene*, and *shot*, where each entry is composed of one or more entries at a lower level. For example, a segment is composed of a sequence of scenes [3]. At the lowest level, a shot describes continuous action between the start and stop of a camera operation and is the fundamental (or elementary) unit of video.
3. Content data. Users of video-retrieval systems want to find videos on the basis of the semantic content of the video. Video contains visual content and audio content. In addition, because of the nature of video, the visual content is a combination of static content (frames) and dynamic content. Thus, the end user may want to search 1) sets of keyframes[1] (e.g., a keyframe for each actor, or a sequence with one keyframe for each major segment or scene); 2) keyword indexes built from sound track and/or closed-caption; 3) object indexes that indicate entry and exit frames for each appearance of a significant object or individual.

Most of the video in large existing legacy video databases has been annotated solely by hand, if it has been annotated at all, with meticulous previewing of the video. (Some video may have only the type of footage and the date and time it was recorded associated with it. Even worse, the category of footage may not be known, and all that may be known is that the video is of, say, a sporting event.) Research in video-data modeling has been centered on proposing video abstractions that offer enhanced textual descriptions of the video (e.g., "house fire, three fire trucks") beyond visual content [4–7]. These abstractions are determined manually, and annotations are added by an individual, often with the assistance of user interfaces. Such descriptions assist not only the building of query languages that enable efficient storage and retrieval based on visual content of video in database systems, but also the management and manipulation of individual video clips for multimedia applications. In addition, existing clips of video can be readily made available for "re-purposing." The consensus of experts in the field is that when larger databases of video are involved, video processing to automatically extract content information may be crucial [2]. At least, the use of intelligent aids, based on automatic video processing, for manual video annotation is essential.

This paper proposes a model of video retrieval and a novel framework of structural video analysis for video annotation. In particular, it is concerned with the process of video query, the system and structure of building the query, and with the automatic recognition and generation of syntactic and semantic video annotations. Such an integral process of video query and annotation has not been addressed in adequate depth by current literature. In Section 2, we formulate and discuss the different stages of the video query process and address the different information modalities that are available for video, such as closed-caption, sound track, and visual information, and how these modalities might be used in the query process. These different information modalities can be extracted and analyzed to build a list of attributes describing the video, and metadata indexes of video can be constructed from these attributes. We *do not* argue that text and image searches are not important for video retrieval; we *do* argue that the ability to search on other video properties is important.

To allow scalability, many attributes of video can be extracted and annotated explicitly and implicitly by fast computers, but automatic processing algorithms for

[1] A keyframe is a previously selected frame characterizing a shot, scene, segment, or movie.

analyzing video must be developed. In Section 3, we focus on the analysis of visual content in video to extract automatically the attributes for video query. We introduce in Section 3.1 the characteristics of video and its prominent structures for presenting the content, and we note the importance of segmenting a video into shots, the elementary units. Section 3.2 argues that annotation by textually describing shots, and retrieval of video based on the shots, may not be the best approach. We then present a new model of computational analysis based on between-shot relations that provides powerful tools for extracting descriptions of video structure. To illustrate the ideas, selected algorithms and techniques for the automatic processing and analysis of video are summarized in Sections 3.3 to 3.8. We conclude the paper and discuss other aspects of video query in Section 4.

The ability to process, present, and organize digital video is vital for the building of multimedia information systems. In addition, the tools and techniques may become important components of many digital-library applications. While current focus on digital-library efforts has been primarily on textual and on static image data, we hope that the techniques proposed and the video-query model presented in this paper will be able to fill in the voids in video search, manipulation, and presentation.

In addition to having value to IBM, our research is part of the NIST/ATP[2]-funded research consortium[3] formed to develop the high-definition television (HDTV) studio of the future. The consortium is charged with performing the research and development needed to have the necessary components ready for a fully integrated HDTV studio.[4] The video-query portion of the project will offer valuable insights into the creation and management of digital video archives and will provide the foundation for large-scale trial of the system and techniques proposed in this paper.

2. Stages of video query

A video query is more complicated than a traditional query of text databases. In addition to text (closed-captions and manual annotation), a video clip has visual and audio information as well as the dynamics associated with the presentation of such information. Of course, a key question to ask is how a query can be performed on, and across, the multiple information modalities, and how it can be done seamlessly.

The prime concern of any video retrieval system is that a query be natural and easy to formulate, that the user–computer interface assist in a user-friendly way in the query-formulation process, and that the search results be presented in an organized and sensible fashion. The system should also enable the user to quickly realize whether she or he is asking the right question. A second concern, and probably of equal importance, is that the search be performed quickly.

Since video is a rich source of information, it should be searched so that the candidate list (list of video units[5] that satisfy the constraints of the query) is reduced to a manageable size and duration as quickly as possible. Various types of search indexes can be used—text and static and dynamic visual cues—and the use and the presentation of these cues to the system may very well affect the computational performance of the query system. Therefore, the search engine should use the cues in the proper order, and the user interface should prompt the user for the appropriate cues if they are not presented in the appropriate order.

Completely different types of queries can be expected from the end user:

1. The user may have once seen a particular piece of video and wants to retrieve it for viewing or reuse.
2. The user may be looking for a specific video but has never actually seen it before.
3. The user may have only some vague idea of what it is that she or he is actually looking for.

Ideally, the query-formulation process should accommodate these types of queries as well as the queries of a user who just wants to browse video without even having a specific goal in mind. In order that these goals be achieved, the different information modalities should be fully used and exploited in the query-formulation and search processes, in order to allow the end user to find the video of interest, or units of interest, whether he or she finds it by a slogan well remembered ("Hasta la vista, baby" in *Terminator II—the Judgment Day*), by a scene vaguely recalled (the tango scene in *Scent of a Woman*), or simply by some keywords together with extensive browsing of the summaries of video because he or she has no good idea how to formulate a specific query on the segments of interest.

In this section, we model the formulation of a video query as a sequence of stages—each stage being an active filtering of information, to reduce the size of the candidate data pools. Each stage involves interactive query formulation and gives a more refined query to the next stage. The sequence of stages is as follows: query on the category of video (navigating), query on the text and/or audio and visual feature descriptions of video (searching), query on the semantic summary of visual content (browsing), and query on the full-motion audiovisual content (viewing). In other words, we view a video query

[2] National Institute for Standards and Technology/Advanced Technology Program.
[3] Program manager: David K. Hermreck, *david.hermreck@nist.gov*.
[4] See *http://www.atp.nist.gov*.

[5] We use "unit" of video to mean a piece of video, e.g., movie, segment, scene, shot, or part of any of these.

as *navigating*, *searching*, *browsing*, and *viewing* iteratively and flexibly. We present in the following subsections how we envision the different stages of the query-formulation process.

2.1 Navigating

This is the stage at which the user (or ideally the search engine) decides which category of video is to be searched. Navigating is the capability to use metadata, for example, to direct the search to a specific interval of time, direct the search to a topic (Clinton or football), direct the search to a specific category of footage (sitcom, documentary, raw footage, etc.), or even direct the search to a specific video server.

The category of footage is an interesting video feature. A form of classifying video on the basis of "applications" and purpose is discussed in [7], which cites four main classes: entertainment, information, communication, and data analysis. The entertainment class can be further divided into subclasses like fiction (motion pictures, TV programs, etc.), nonfiction (sports, documentaries, talk shows, etc.), and interactive video (games, home shopping, etc.). The information class contains videos that convey information, such as news and training programs. Video conferencing and presentation videos are grouped under the communication subclass. Lastly, scientific video recording, like medical and psychology experiments and surveillance video, may be used mainly for data analysis purposes. Other forms of categorization are possible, of course.

Such a classification of the video could create problems for certain video items. For instance, what class does news about a sporting event belong to? Is it *news* or *sports*; and when should it not be *news* anymore? Thus, certain videos will have to be placed into multiple categories, or the burden could be put on the user to select multiple categories. That is, the user may have to select more than one category to search on.

At present, almost all video has some text associated with it, albeit the amount of text may be quite small—say, a few keywords. Such text can be bibliographic information, subject information, some manual annotation, and more complete closed-caption or speech transcription. So it is not unreasonable, and probably very sensible, that the initial "navigating" be formulated in textual form (from which the category should be automatically inferred). We can then borrow techniques (called *source selection*) which are used in text retrieval, i.e., deducing from the query which body of text should be searched. This will reduce the search times significantly. The drawback is that when the source selection is erroneous, the video item of interest will not be retrieved. Video categorization and source selection are intimately related. When there is sufficient text available with the video and the query is properly posed, source selection is no more of a problem than it is in the text-retrieval area today.

If video categorization and source selection are applied, each video has to be classified according to the source it should belong to, by a combination of available manual textual annotation and visual and audio content. Automatic video categorization based on nontextual content of the video is an open and relevant area of research, because some of the satellite footage received in broadcast studios may be recorded without monitoring or annotating.

2.2 Searching

Searching is the most important part of query in most database systems, and video-database systems are no exception. The result of the search is a list of candidate units that satisfy, in some sense, the constraints of the query. The ultimate goal of the search is to make this list as short as possible without missing the video(s) of interest.

The selection of a particular category of video during the navigating stage limits, to some extent, the scope of data to be searched, but there still is the need for efficient and effective means of further filtering the data of interest. Data-object models are used for organizing data in databases to allow fast retrieval. In video, the data objects developed have to accommodate the various information modalities such as text, audio, and visual content. This poses an interesting challenge, as many aspects of video cannot be described simply in words or by static images (keyframes), and semantic and content-based video modeling is largely unexplored.

Text-based search could be a first step in the searching stage; it serves as a good search filter and is an extremely important search tool that should by no means be neglected in the video-retrieval problem. Text search is also a more mature, straightforward, and faster technology. All of the advances that have been made in recent years in the area of text retrieval should be brought to bear [8]. Many multimedia data objects are modeled with the use of text to describe the attributes of various information modalities. In this context, *metadata* of multimedia data have been proposed by different researchers [9]. The metadata in this current data-modeling research always implies text data. What we argue in this paper is that this metadata can consist of many forms of data, including, of course, keyframes.

It is also important that the text descriptions of the attributes accurately reflect the characteristics of nontext multimedia data types to a certain extent, and with the best capabilities. These text descriptions should also describe the content sufficiently to help the users to locate segments of interest or, at least, not exclude potential segments from the candidate list. Textual attributes of video segments are essential components of the video query, but not the only ones. These attributes can be associated with the video and interpreted. Such attributes as the date of creation, the title, and the source are commonly available at the time of creation. Some keywords or comments may be added later. On the other hand, attributes such as the number of shots, audiovisual events such as dialogue and music, recognized spoken words (by means of speech-recognition technology), and images can also be derived from automatic analysis of the video.

In the following example, we present a search scenario. We assume that logical operations are supported as in any text database systems. The category of video is news. For each news clip in the collection, a metadata object exists containing a list of textual attributes. **Table 1** shows some fields in the attribute list of a video clip.

Let us suppose a user wants to find certain video segments on recent air tragedies. He enters air AND disaster in the proper fields of the user interface and also enters additional constraints, using *start date = 01/01/96* and *end date = 09/30/96*. The resulting candidate list (simplified) might look similar to the list presented in **Table 2.**

The n-star rating in the first column is an indication of how well the particular candidate (or hit) satisfies the query; i.e., how relevant the candidate item is to the query. (It has been found that, rather than a numeric value, users prefer ratings that they are familiar with—like the Michelin restaurant rating.) This particular query has produced 254 video clips, with a total duration of 8.4 hours. Of course, the interface displaying the candidates should be much more graphical, and thumbnail images (or other visual representations in addition to textual summaries) giving some indication of the visual content of the individual hits should also be displayed.

At this point, the user may feel that this is still too much material and may, for example, specify the additional keywords "TWA" and "bomb," by entering air AND disaster AND TWA AND bomb, indicating that the desired video clip should focus on the TWA crash and the possibility of a bomb as the cause of the explosion. An example of the resulting hit list is shown in **Table 3**, which still has 109 hits of total duration 4.7 hours, which is surely too much time for the user to spend watching linearly.

The amount of text associated with the individual video clips in the video database is potentially unbalanced. Some video may have only two or three keywords associated with it, stemming from traditional alphanumeric database and manual annotation technology, while other video may have the script, closed-caption, or speech transcript available. Besides, there is a limit to how much semantic information the textual attributes can provide. For instance, how does one textually describe the pace of a tennis match? Thus, further text querying may not be able

Table 1 Example of attribute list of a video clip and the textual description of each field.

Attribute list (text)
Hand-annotated attributes
Category: news Title: What caused TWA 800 explosion? Date: 7/18/96 Duration: 45 min Source: WNY1 Keywords: TWA, crash, . . . ⋮
Computed attributes
Number of stories: 5 Number of shots: 50 Event types: narration, dialogues Number of dialogues: 10 Number of speakers: 5 ⋮

Table 2 Candidate list (simplified) resulting from query.

Hits: 254	*Total time: 8.4 hours*			
Relevance	*Title*	*Date*	*Duration*	· · ·
***	Insurance companies waiting for the answer	9/3/96	10 min	· · ·
***	Security in New York airports	9/2/96	20 min	· · ·
***	FAA under fire	8/6/96	2 min	· · ·
***	Rough sea deters debris recovery	8/4/96	5 min	· · ·
***	Traces of chemical found	7/29/96	10 min	· · ·
***	What caused TWA 800 explosion?	7/28/96	45 min	· · ·
**	TWA flight crashed	7/18/96	15 min	· · ·
**	FAA under criticism	7/6/96	3 min	· · ·
**	ValuJet plane plunged into the Everglades	6/15/96	15 min	· · ·
⋮	⋮	⋮	⋮	⋮

Table 3 Refined candidate list (simplified).

Hits: 109	*Total time: 4.7 hours*			
Relevance	*Title*	*Date*	*Duration*	· · ·
***	Insurance companies waiting for the answer	9/3/96	10 min	· · ·
***	Security in New York airports	9/2/96	20 min	· · ·
***	Traces of chemical found	7/29/96	10 min	· · ·
***	What caused TWA 800 explosion?	7/28/96	45 min	· · ·
⋮	⋮	⋮	⋮	⋮

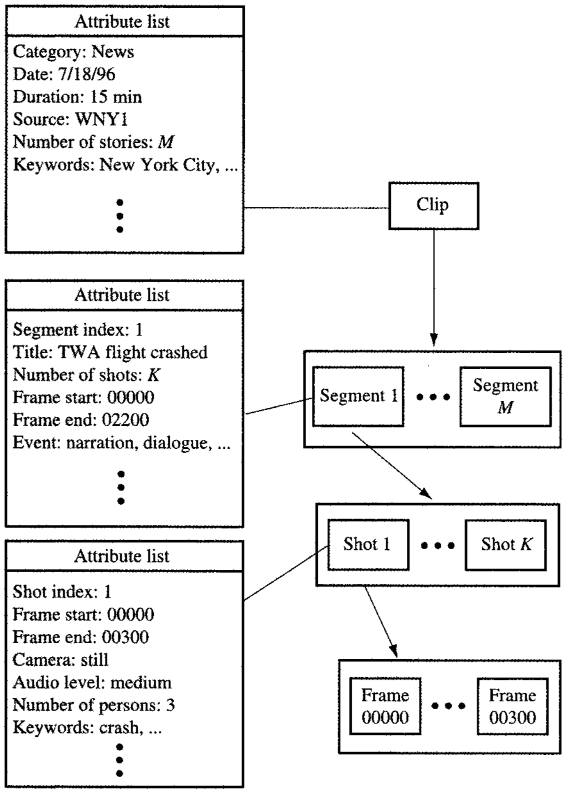

Figure 1

Sample hierarchical representation of video objects and associated attributes.

to narrow down the candidate list much more. At this point in the search, it helps to look at the other attributes, manually annotated or computationally derived, for further information. One attribute of a candidate in the list is the actual clip length (duration), which can give some indication of the content. Other attributes, such as number of stories (visual narratives) and shots in a clip, also indicate the type of programming that is represented by the candidate clip. A clip that contains many shots (most likely the shots have short durations) reflects fast-moving footage like headline news; fewer (and longer) shots, on the other hand, indicate drawn-out programs like senate hearings or news conferences. Hence, the ability to order the candidates by certain numeric attributes will be valuable in the query process. In our example, the candidates in the list are arranged in reverse chronological order, since news pieces are timely material, and such an arrangement may be able to provide hints for the user to locate material of interest from the long list.

Audiovisual attributes beyond text are thus important and augment the text-query process. The difficult question is to define the appropriate audiovisual properties that can be extracted from video and will bring this candidate list down to a manageable size.

One would like to interactively constrain the acceptable visual and/or audio attributes of the candidates and shorten the candidate list. Query-by-image-content techniques have been proposed in recent years—for example, in [10]. The visual attributes may be in the form of color, object shape, and texture. There is also active ongoing research in audio indexing, to extract special audio features (e.g., [11]). Video, through the dynamic nature of its audiovisual content, is equipped with rich reserves of potential features. These features can be syntactic and localized to a particular group of frames, like a set of dominant colors, or semantic in nature, like a dialogue. We address the attributes of video based on visual content in Section 3, including a discussion of the intra-shot (or within-shot) and inter-shot (or between-shot) properties.

Another aspect of video search is that a user may not be interested in entire clips, but rather portions of the clips. This means that the search, in the ideal case, should be able to return as a result data objects (video portions) consisting of relevant footage. This further implies that the data objects should be built in a hierarchical manner. In such a context, the attributes of the top-level objects should be able to pass to the lower-level objects. In addition, each object in the hierarchy will have its own special attributes to describe the characteristics of the video portion. A meaningful hierarchy of video is defined in the film books (such as [12–14], published by cinematographers or other experts in the film world). A shot is the fundamental unit, a scene is a collection of shots unified by the same locale or dramatic incident, a segment is a collection of scenes, and a movie is a collection of segments. In disciplines other than motion pictures, different terminology is used. For example, television news magazines may refer to "news items," which are analogous to segments, and sports events consist of "plays," "quarters," and other similar entities. In the remainder of this paper, we use the following hierarchy: clip (or story), segment (or story unit), and shot. The attributes of a clip should be designed to include the most general features (or attributes) that can be inherited by the stories, segments/scenes, and shots it contains, and similarly those of a segment by its shots. A diagram representing the data hierarchy and a sample attribute list associated with each level of data object is shown in **Figure 1**. At the top level is a clip that comprises M segments. The attributes associated with this clip include the category (news), the title (TWA flight crashed), and some keywords which may be manually annotated or extracted automatically by means of speech recognition of the sound track, for example. At the next level of the hierarchy, each segment is made up of several shots. The attributes include duration, the start and end time-codes, and the types of events that are found in the segment (e.g., dialogue). Finally, at the lowest level of the hierarchy, the detailed attributes of individual shots are listed. Such attributes attached to each shot can include low-level characteristics such as motion, recognized keywords from speech, and frame-level features.

To enable video search, the video clips have to be properly segmented into meaningful semantic units. To make available the derived text attributes, and to provide audiovisual features, syntactic or semantic, the content of the video must be characterized. Manual segmentation (partitioning) and characterization is time-consuming, skill- and knowledge-dependent, and potentially limited to only the attributes that have text equivalents. For example, textually describing a color is hard and varies from one annotator to the next. Automatic analysis of video content can achieve meaningful segmentation, provide valuable characterization of video, and offer features that are *depictable* and *nondepictable* (the words of Eisenstein in [12]) with words. We detail the analysis of visual content in video in Section 3.

2.3 Browsing

The result of the search stage is a collection of video candidates, the total duration of which may be long. Hence, during the next stage of video query, all candidates should be *browsable* in some sense. Browsing the video content is a highly important aspect of video query formulation because textual descriptions can supply only a "view" of the video content. Oftentimes, *what* to query for a video segment of interest is hard to formulate in text equivalents, and what makes text-based query even more difficult is that the choices of wording are subjective and highly dependent on the level of understanding of the material and the education and background of the individual. This means that the textual attributes put forward by an end user may not match in any way the textual descriptions used by the annotator.

In the browsing stage, representations of video that are good high-level overviews of the content of the candidate videos should be displayed. A user, by looking at such representations, can quickly understand the video content and browse through many videos in a short period. In addition, the user rapidly gets an idea whether she or he is asking the "right" questions and determines how to redefine or refine the query. Ideally, the user should also have random access to *any* point of *any* video, no matter how long the actual candidate list may be. Moreover, the user should have the capability to view video in a nonlinear fashion and be able to get an overview of each candidate video by viewing, ideally, only a small area of a single computer-display screen. That is, high-level representations of video, in the form of *visual summaries* (*video visualization*), are necessary or at least highly desirable.

The area of visual summaries is an important facet of the video query process and certainly deserves to be recognized as an important research topic—both for the sake of the representation, interpretation, and refinement of search query results and for low-bandwidth video transmission. The problem is to derive or compute a mapping from the video data, which can be considered to be a three-dimensional object (the three dimensions are the horizontal and vertical dimensions of the individual frames and the time dimension), to the two-dimensional plane for screen presentation. This has to be done in such a fashion that the underlying structure of the video is obvious to the viewer (the end user). The video structure represented in a visual summary should preferably be highly correlated with the semantic content, and allow simple semantic video interpretation for sufficient understanding. An ideal visual summary might be keyframes of the video—as few as possible to adequately depict the narration (e.g., for fiction videos) and the story line. This involves automatically finding the semantically most important dramatic elements within a clip and, subsequently, the most important images within a dramatic element. Thorough high-level semantic interpretation of video by automatic means is a very open and active area of research in the multimedia and computer-vision communities [15, 16].

On the other hand, with today's technology, simple high-level structures can be automatically extracted and interpreted. As discussed in the next section, the structure in terms of segments and shots can be rediscovered by means of agglomerative clustering (see [17] for references) of shots represented by color histograms of keyframes. In a sense, the EDL (Edit Decision List) used by editors of sitcoms is (partially) rediscovered. This permits graphical representations of the scene-to-scene structure of video, which is built on top of the basic units of shots. One such representation is a graph representation, discussed below, in which nodes in the graph represent similar shots [18], and connected subgraphs represent story-units of the video [17]. Other representations of the story structure can also be envisioned—for example, hierarchical representations.

The development of visual summaries representing video depends heavily on the category of the video. Different categories of video will require different forms of visual summaries. While story structures are prominent in sitcoms, news, and films, programs of sporting events do not have the structure of a story. A natural segment (i.e., a meaningful unit) for a team ball sport may be a period during which the ball is in the "possession" of one of the

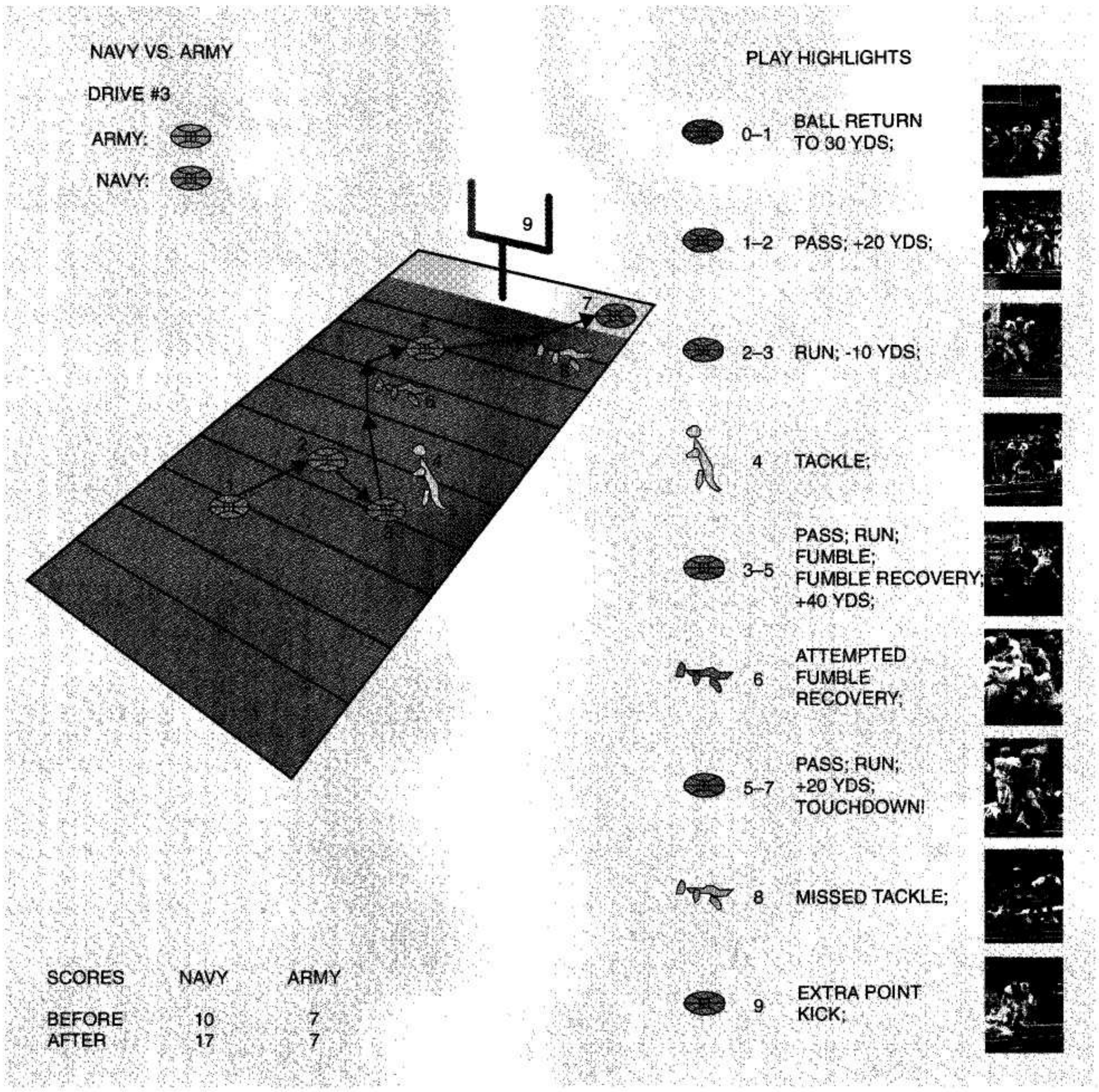

Figure 2

Visual summary of part of a football game (American). Images are © Phil Hoffmann Photography.

teams, and a segment break occurs when the ball changes possession. As an illustration, **Figure 2** presents a possible visual summary of such a segment of a football game (American), summarizing the highlights in a "scoring drive." The ball symbols are at the starting and ending positions of each play in the field; the edges point in the direction of "offense" motion, and the human figures indicate "defense" actions. The user can click on any point along the play to see the video, which is equivalent to providing random access into the video. In addition to a keyword summary (shown to the right of the figure), a visual summary can include some significant keyframes (the boxes on the far right), which feature the memorable images of the play. Other attributes, such as camera zoom-in views and sideline actions (e.g., cheerleader performances) can also be included in the summary presentation. Each highlighted feature, whether it is a keyword-based description, graphic symbol, or image, can be linked to the content it represents, and the video segment can be readily presented to the user should there be interest. Of course, automatic generation of such a summary is very difficult, and it is more feasible to combine manual summary creation with a good degree of automation. More details on the construction of visual summaries is given in Section 3.7.

2.4 Viewing

After one or more candidate videos are selected as the most likely, the user may decide to confirm his or her search. In other words, the user has to view parts of the candidate. The usual functions of today's videocassette players should be available in this stage: *play*, *pause*, *fast-forward*, *reverse*, and *variable play*. Further, capabilities like *semantic fast-forward* (the ability to move forward in the video on the basis of semantic video content rather than time-code or visual inspection) should also be available in the viewing stage. That is, the user should be able to skip forward (or backward) to frames, shots, or scenes that she or he gives some semantic description of. For example, the user should be able to fast-forward to the next news segment.

The huge amount of data inherent in video calls for compression in order to achieve efficient storage and transmission. It is expected that most video clips in digital libraries and multimedia databases will be stored in compressed formats, most likely in standard video compression formats like Motion JPEG[6] [19] and MPEG** (1 and 2) [20, 21]. In MPEG compression standards, predictive and differential coding techniques are used. This implies that random access to any frame of a video cannot be implemented in a straightforward way. For example, efficient algorithms have to be developed for reverse and variable play of MPEG video. The manipulation, access, and management of compressed data streams without full decompression is an area of active research (e.g., [15, 16]).

3. Video analysis and processing for video query

The purpose of this section is to note the type of work that we feel must be done to achieve flexible video-retrieval systems. For that reason, we survey some of the existing work and indicate research directions that we feel are important. No new research results are presented in the survey portion, and the reader is referred to the appropriate literature for details.

As discussed previously, in the existing schemes for flexible video retrieval (e.g., [2]), the data objects in video are often extracted manually, and associated with the video by the operators of video annotation stations, often with the assistance of user interfaces. These annotation types, however, are often proposed with the viewpoint that video has an unstructured presentation format, and they may not have been designed to consider the capabilities of machine automation in the processing of video content and the extraction of semantics and structure. Fundamentally, the description of content that human operators can identify and associate with the video (mostly keywords describing semantic content) is very different from, and difficult for, algorithmic operations—i.e., machine interpretation of video semantics. In addition, no one may ever develop an algorithm to interpret all of the semantics from the video automatically and extract meaningful annotation automatically for every video sequence.

Fortunately, most videos (movies, TV programs, documentaries, sports programs, etc.) are *structured* so as to convey the underlying narrations and messages. That is, the shots are combined in specific ways and according to special orders (in contrast to random organization) to form the *montage* in telling the story. Because of this, certain temporal features can be recognized, and associated information can be extracted, by automatically analyzing the visual contents and temporal associations of the video.

In this section, we first look at the most fundamental unit of video production—the shot. Techniques for detection of the boundaries between shots and current work on processing individual shots are also discussed and surveyed to some extent. We then motivate the need to examine inter-shot relations, in order to extract high-level structure and high-level semantics from video, and we summarize current efforts in this direction.

3.1 Video shots

The fundamental unit of film and video is a shot. The content of a shot depicts continuous action captured by a single camera during some interval. Hence, a shot presents an *image of time*—an interval of space-time. The importance of the shot as the fundamental video unit has been realized by many researchers, and the computational detection of video-shot boundaries has received much attention. The act of automatically segmenting a video into shots is commonly called *scene-change detection*, which should be more properly called *shot-boundary detection*. There are many types of shot boundaries, among which are abrupt and gradual (a *dissolve*, as it would be called by a professional video editor). In **Figure 3**, we illustrate these two types.

Major efforts in shot-boundary detection have been focused on algorithms that operate on the full video frames (see [22–27] and the references therein). These

[6] Motion JPEG codes each frame independently, without interframe coding.

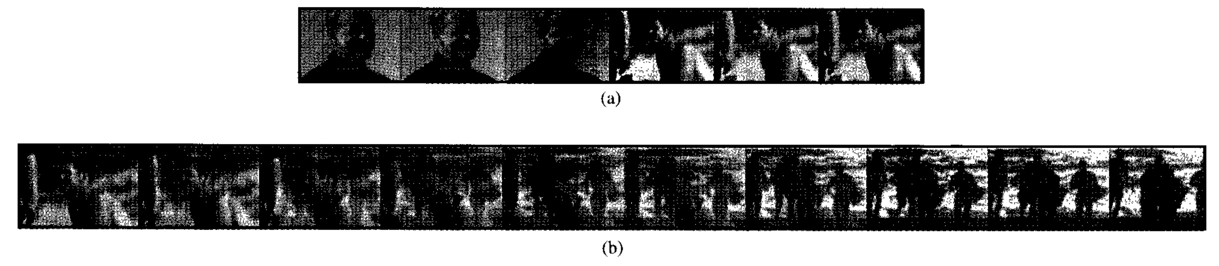

Figure 3

Example shot boundaries from an IBM commercial: (a) Abrupt boundary, with three frames before and after the boundary; (b) gradual transition, with ten frames showing the transition from one shot to the next.

efforts are based primarily on automatic schemes that analyze the variations in visual primitives (e.g., color, edges) from frame to frame. There are also recent efforts in performing segmentation on compressed video. Except for [28], which considers Motion JPEG video, the works study shot-boundary detection on MPEG compressed video [29–33]. Several works rely partly on motion-vector information for the boundary detection [29, 31, 32]. Sometimes decompression has to be performed for further analysis at full resolution [32]. In [30], the dc (direct current) coefficients of I-frames are used to determine luminance histograms, which are employed to perform hypothesis testing—the testing of the hypothesis that there is a shot boundary. It is assumed that the separation between two I-frames is fixed and small (assumed to be six frames in the paper). Yeo and Liu [33] developed their detection algorithms to operate on sequences of reduced images, which are extracted from MPEG-1 video without full decompression. These schemes are sufficiently accurate to segment most shots in a video sequence. Shot-segmentation techniques that analyze video directly in compressed formats save both the auxiliary storage for decompressed data and the computation costs of decompression, since a significant portion, if not all, of the video is typically stored in standard compression formats.

Most people working in this field believe that in a retrieval system, the shot should be the minimum addressable and retrievable unit for the user, rather than a subsequence of a shot, a frame, or individual pixels. The ability to address and retrieve individual frames or pixels is surely required for non-linear editor (NLE) systems (tools used by film and video editors), for example, in which frame-accurate operations such as cutting and pasting, or operations in which pixel values are changed, such as dissolves, are common. For video query, we ask what is the appropriate data abstraction for a video shot. Another question is whether the shots should be treated in isolation or whether the time-oriented structure of the sequences of shots should be taken into account in the annotation data abstraction.

Various researchers [34, 35] observe that the representation of a video shot as a single still image could be a significant step toward a solution of the video indexing and retrieval problem—flexibly retrieving pieces of video from (possibly large) collections of video clips. If there is no independent motion (only camera motion) in a shot, the transformation between subsequent frames can be determined, and the frames can be "pasted" together into a larger image, possibly of higher resolution. This is a technique often referred to as *mosaicking* [36]. If there is independent motion, the moving object(s) must be separated from the background, and the high-resolution image of the background can still be reconstructed—along with images and trajectories of the moving objects. Techniques such as salient stills [34, 37, 38], mosaicking, and video space icons [35] are proposed to obtain such still images. One of the rationales for this processing is that once a complete shot is transformed into a single still image, traditional image-database-browsing techniques such as [10] can be applied. In addition to providing information about static image features such as color, texture, and segment shape (the shape of the object images), the operation of registering, or matching, a sequence of frames to obtain a still image gives some information about camera motion and independent object motions, which permits motion queries [36]. Mosaicking is also used to construct a larger view of selected highlights in, for example, soccer games (e.g., when the ball is near the goalposts) [39]. Such a *background mosaic* is useful for later reconstruction of the *action mosaic*, in which individual frames, when displayed one after the other in real time, permit the spectators or a team's players and/or coaches to watch the action and review performance over a larger interval of time. That is, the actions of players can be observed against the high-resolution mosaicked background. These types of algorithms that register frames, of course, also produce a sort of visual summary—but only of short video sequences. (A problem here is that the content owners do not like to have their content altered.)

Mosaicking provides a single image that represents a shot and can be used as annotation for the shot, including possible annotation on how the mosaic was assembled—for example, motion information.

A simpler way to obtain a single image to represent a shot is to select a keyframe [40] or, if there is much motion and action in the shot, to select multiple keyframes [32, 41]. In this way, a video shot is represented by one or more images, and the problem of video query is reduced to the problem of image query, for which one can rely on traditional image search engines (e.g., [10]) to retrieve video. However, an hour of video is typically composed of a few hundred shots. Searching large video databases then amounts to searching large image databases. For example, a video database of 100 hours (four days of continuous broadcasting by one TV station) contains about 10 million frames. If, on average, the keyframes constitute about 0.5% of the data (i.e., that one in 200 frames is selected as keyframe), the video for 100 hours is still represented by some 50000 images. Apart from the fact that the search results for image databases frequently do not seem to have much to do with the initial query (rather often the search results appear to be a random collection of images constrained in some sense by the search query), even a small number of hours of video can stretch the limits of image-search engines in terms of the size of image candidates and the computational requirements.

Techniques like mosaicking are examples of within-shot processing, in that only the data within the shot is used. Motion analysis of shots can be taken much further. In [42], a hierarchy of representations is proposed that, at least in theory, can be extracted, by means of motion analysis. At the lowest level, one can extract camera motion (the *ego-motion* problem, as it is called in the computer-vision literature). Once that is known, one can determine the independently moving objects, which in turn gives some indication of the visual appearance of the objects. Once camera motion and motion of independently moving objects are known, shape information can be extracted—the next level in the taxonomy. However, with an uncalibrated camera, *only* ordinal shape but not metric shape (the algorithms will only be able to compute, e.g., relative depth—not absolute depth) can be extracted. Finally, at the top level of the hierarchy, is space information—that is, information about the ordinal shape of the locale in which the video shot is filmed. All of these types of information are possible candidates for shot-based indexing.

Clearly, to perform video search, more information has to be used than just static images derived from the video. Video query has to go beyond the static features of images and incorporate the dynamics within a shot and between the shots.

3.2 Inter-shot analysis

In video and film, a story is told by the presentation of the *images of time*; the sequence of such a presentation is called the *montage* or the *edit*. Technically, *montage* refers to the "editing of the film, the cutting and piecing together of exposed film in a manner that best conveys the intent of the work" [43]. Montage was studied by Eisenstein and many others. As put forward by Eisenstein [13], "But this is—montage! It is exactly what we do in the cinema, combining shots that are *depictive*, single in meaning, neutral in content—into *intellectual* contexts and series." Modern cinema uses the montage presentation extensively to convey meanings. Actions have to develop sequentially; simultaneous or parallel processes have to be shown one after the other in a concatenation of shots.

As observed by Miller [44], properly edited video has a continuity of meaning that overwhelms the inherent discontinuity of presentation. Miller notes further that if one watches a film or a TV program from so far away that one can neither hear the sound nor recognize the faces, one will almost certainly be struck by a staccato interruptedness, of which one may not be aware if the same display is seen from close up. Something ensures that the continuity of meaning is preserved when the program is viewed from the proper distance. The continuity that obtains from shot to shot—from wide shot to close-up, from one speaker's face to another's and so on—is achieved by the viewer's ignoring the interruption by using the more or less conscious knowledge or understanding that the situation is identifiable from one shot to the next and what is shown is nothing more than various aspects of the same scene, as noted in [44]. What further glues the pieces of video together into a continuous narration are various rules of thumb that all editors observe in an effort to maintain coherence and cohesion from shot to shot. For example, if two faces are to be seen addressing each other, it is important to guarantee that the eyelines of the faces are aligned [14]. And, more often than not, the editor will occasionally revert to an "establishing shot" [14] to remind the spectator of the spatial setup.

In summary, the medium of film and video has at least two levels of discontinuity. One of these is the frame-to-frame discontinuity, which is psychophysically unperceptible; the other is the shot-to-shot transition.

This latter discontinuity may be unperceived, but if the psychological conditions are not favorable, it is all too perceptible [44]. Because of the mechanical act of film editing, the shot boundaries are in most cases easily detectable computationally. In essence, the shot is the cinematographic building block and often contains only minimal semantic information. In a sense, a shot in a video is very much like a sentence in a piece of text—it has some semantic meaning, but taken out of context it may not have that much.

A story is built of shots. Groups of shots are concatenated to form a depiction of a three-dimensional event (say, a car chase) that is continuous in time—we call such a concatenation of shots a *story unit*. A video usually consists of multiple story units in which, beyond the story-unit discontinuity (where the video changes from one story-unit to the next), there may or may not be continuity of time, the narration can move to a different place, or the narration can move to a different time and place altogether. (Without a change of cast, the former—a change to a different locale without time elapse—is, of course, physically possible only if the narration moves between adjacent locales.) Either way, the continuity of time is not as important as continuity of meaning; i.e., does the same narration continue in the next story unit or segment? This is perhaps the most challenging aspect of automatic video annotation, *finding the underlying discontinuities of meaning*, or, equivalently, establishing from shot to shot whether there is continuity in the meaning or the subject of the video. There will be no continuity of meaning if the TV program goes to a commercial break or if the anchorperson of the news changes to a different news item.

Once video segmentation based on meaning or subject has been performed, one is left with video units, each of which deals with a particular subject and is uninterrupted by video about different subjects such as commercials. The task of automatic video analysis, and the subsequent annotation, then is one of finding high-level interpretations—*not* of individual shots but of collections of shots, and possibly of the video as a whole. It seems that to find these higher-level interpretations, between-shot analyses are at least as important as within-shot analyses via image computations. After all, it is the relation or lack of relation between two shots that we need to compute, which can only be achieved by computations on data that belong to *different* shots. This will allow rediscovering the structures of the video and associating semantic meaning to sequences of shots, thereby interpreting a large amount of data without expending too much computational effort. It will also allow the grouping of shots that exhibit the same meaning and identifying shot breaks where there is a discontinuity of meaning.

The goals of between-shot processing are to derive high-level video structure for automatic annotation of video and for visual presentation of the video. If a sitcom can be segmented into its story units, a user can more conveniently browse through that sitcom on a story-unit basis than on a shot-by-shot basis, as is commonly done in practice. There is typically an order of magnitude reduction from the number of shots to the number of scenes [45]. This represents a significant reduction of information to be conveyed or presented to the user, thus making it easy for the user to identify segments of interest.

Then, the ability to also automatically label a story unit as (say) "dialogue" or "action" means that one can query or further refine the query on the basis of such semantic characteristics. That is, during the search stage, the user will be able to define or refine the query by searching on automatically derived semantic interpretations. Visual summaries can be labeled with such semantic descriptions (in a pictorial or textual fashion), allowing video-content viewing and rapid random access.

Some research efforts in exploiting between-shot relationships are surveyed in the following sections, to further explain the ideas presented in this section.

3.3 Model-based parsing

Video parsing has been successfully applied to recover individual news items from news programs [46, 47]. *A priori* models are constructed through the use of state transitions, where each state corresponds to a phase of a news broadcast such as a news anchor speaking, a reporter speaking, and a weather forecast. In the recognition steps, visual and temporal elements are used together with domain knowledge of spatial layouts of objects. In addition, specific knowledge of station logos (e.g., CNN Headline News) is used to identify the sequence that marks the return from commercial breaks.

After the different items of the news broadcast are recovered in the analysis process, an interface is provided for a user to view any episode of a news program. This offers a high-level random access into different segments of the news. If a user is interested only in the weather forecast, he or she can jump into the segment on the weather forecast without having to view the video linearly.

Model-based video parsing takes advantages of specific domain knowledge of the news program. Some changes to the model might be necessary to accommodate news from other news stations with varying styles of news presentations.

Another approach to news-story segmentation can be found in [48], where a unique combination of sources of information is used. Typically, for closed-captioning (subtitling for the hearing-impaired), the beginning of a new news item is indicated by the symbols > > > at the start of the closed-caption line. Also, because of the (often) real-time nature of closed-captioning, the actual closed-caption (encoded in line 21 of the NTSC signal) may lag behind the actual spoken words. In [48], a shot-boundary-detection algorithm in combination with the detection of audio silences and the new-item indicator of the closed-captioning is used to segment the news into individual news items. In addition, the closed-caption text is synchronized with the visual news item and used for text-based retrieval of news items.

A different approach is taken by Shahraray and Gibbon [49] on the use of closed-captioning information for news archive retrieval. The pictorial transcript systems they developed transcribe news programs into HTML-based (Hypertext Markup Language) summaries. Each HTML page contains several news items, each being represented by a few keyframes with detailed text derived from closed-captions. Furthermore, linguistic processing and closed-caption control information are used to align complete sentences with individual keyframes and for conversions of closed-caption texts to lowercase with correct capitalization.

3.4 Classifying and labeling video shots

Beyond the parsing based on specific domain knowledge of the video, a generic step toward video processing is the semantic labeling of video shots. We want to associate with each shot s_i a label L_i that provides a description of the content of the shot. For example, a shot of a news anchorperson could be labeled as "news anchor," "newsroom," or "man behind table." Such semantic description, however, is difficult to derive automatically. Instead, one can take advantage of a consequence of the edited representation of parallel or simultaneous events—repetitions of shots with similar contents. To capture such repetitions, we cluster video shots.

Time-constrained clustering [17] has been developed to compute the hierarchical structure of certain types of video content. Typically, a video that tells a story, like a sitcom, is composed of a sequence of story units denoted by $U_1, U_2, \cdots, U_n$. The story takes place in a small number of locales, $\mathscr{L}_1, \mathscr{L}_2, \cdots, \mathscr{L}_L$. Then, if we denote the fact that a story unit U_i takes place in locale $\mathscr{L}_j$ by $U_i \in \mathscr{L}_j$, the structure of the video may look like the following:

$$U_1 \in \mathscr{L}_3, U_2 \in \mathscr{L}_1, U_3 \in \mathscr{L}_4, \cdots, U_n \in \mathscr{L}_3.$$

Time-constrained clustering computes this structure of the video by fitting (parsing) this model of the video editing to data derived from the shots.

This clustering process groups the shots according to the data content of the shots. It takes into account both visual characteristics and temporal location of shots within the video. It means that any two shots that are far apart in viewing time, even if they share similar visual (data) content, may represent different contexts or occur in different scenes, for example, a shot in U_1 and a shot in U_n above. Time-constrained clustering prevents two such shots that are far apart in time but similar in data content from being clustered together. In the clustering process, a measure of the distance between the data content of two shots has to be devised, while the temporal distance between the shots has to be taken into account [41].

In addition, the matching is performed only on reduced-resolution-frame sequences called dc (direct current) images, which can be efficiently extracted from Motion-JPEG or MPEG video by means of the algorithm of Yeo [50]. The extraction algorithms perform minimal decoding of the compressed bit-streams; only dc, a few lower-order ac (alternating current) coefficients, and motion-vector information are used. Because no full decompression of the bit-streams is performed, the extraction process is fast. For spatial resolution of 352 pixels × 240 pixels, the dc images, each of which is reduced by a factor of 8 in each dimension to size 44 × 30, are useful for many processing steps described in this section.

A color histogram of a representative frame in a shot is the data content that is used for clustering. The metric used is a distance between color histograms; hence, shots that are filmed in the same locale cluster together. The result is that one can group the shots into several clusters $C_1, C_2, \cdots, C_N$, each corresponding to a different story unit. Features other than color histograms extracted from the shots can also be used, of course. Such features include shot duration, spatial color distribution of the shots, dominant-motion characteristics, dominant texture patterns, spatial moments, and audio features. The same approach can be taken by clustering these features derived from the shots, and different groupings of the shots will be found.

3.5 Finding meaningful story units

In a given story unit, we often find that multiple objects (like members of the cast) co-exist, and shots of these objects are concatenated, with multiple shots of the same objects occurring. Shots from different scenes are not concatenated in time, except at the transition from one story unit to the next. Because of the "intense interactions" between shots within a story unit, one can label shots on the basis of the (data) content derived from the shots (as in Section 3.4). When labels for two different shots are the same, it is very likely that these shots are images of the same objects, because the labels are derived from images containing similar visual data content. These label sequences can be used to segment a video into story units, which closely approximate the "scenes" as defined by cinematographers [12].

The algorithm to find meaningful story units examines the sequence of labels and identifies the subsequences of labels that are of minimal length and which contain all of the identical (or recurring) labels. Consider, for example, a video sequence of 10 shots labeled as follows: A, B, A, C, D, F, C, G, D, F. The first story unit has to contain the first shot as well as the last shot having the same label as the first shot. Furthermore, it contains the intermediate shots. For each intermediate shot, the story unit must contain the first and last shot with the same label. This process of inclusion can be recursively applied to successive shots in the story unit until the last shot in the first story unit has been reached. The second and subsequent story units are then recovered in a similar fashion. In this example, the first story unit consists of shots 1 to 3, and the second story unit consists of shots 4 to 10. The algorithm operates in O(S) time, where S is the number of shots. It can be shown that this method of segmentation is the same as graph-based segmentation through identification of cut edges in a scene transition graph representation [17] (these graphs are discussed below).

We tested the segmentation algorithm on a variety of video materials, ranging from sitcoms, movies, and documentaries to cartoons. It is reported in [17] that there is an order of magnitude reduction from the number of shots to the number of story units found. For example, for a typical sitcom, there are about 300 shots in a half hour of program and about 20 story units identified. This implies that for each half-hour program, a user must examine 20 units instead of 300 shots to get a vague visual summary of the program. Furthermore, it means that a user can retrieve one of 20 story units instead of one of 300 shots.

3.6 *Modeling of temporal events*

In addition to segmenting a video into larger units, such as story units, it is also advantageous to recognize common temporal events within a story unit. Label sequences can be used to recognize dialogue and action events [45]. Using the degree of repetition or the lack of repetition in a subsequence of labels, one can classify the subsequence into one of three categories: dialogues, actions, and other.

A dialogue refers to actual conversation or a conversation-like montage presentation of two or more concurrent processes. In motion pictures, presenting two or more processes as simultaneous and parallel requires that they be shown one after the other in a sequential temporal order. Dialogue events are cinematographic characterizations of intense interactions between two (or more) dominant parties, possibly interspersed with so-called establishing shots [14] or shots of other parties. Models can be constructed to capture the repetitive nature of two dominant shots while incorporating the possibility of "noise" labels. A noise label could represent a shot of the locale where the dialogue takes place, an establishing shot, but it could also represent some other shot. Consider an example of a video sequence of 22 shots with the following label sequence:

$$\underbrace{A,B,A},X,Y,Z,\overbrace{A,B,A,B,A,B},C,\underbrace{D,E,F,E,D,E},G,H,I.$$

Here the labels are derived from visual data content of the shots; hence, shots with the same label are likely to contain the same object and background—for a dialogue, the object is a person. The label subsequence A, B, A at the start of the sequence is not considered a valid label sequence of a dialogue event because of a lower limit on the number of shots in a dialogue. The label subsequence A, B, A, B, A, B characterizes a dialogue in which there is no "noise" label. The subsequence D, E, F, E, D, E also characterizes a dialogue in which label F is a "noise" label.

An action event represents exciting action sequences in action movies and even reflects the progression of the story in more static program types like sitcoms. An action sequence in motion pictures or video is characterized by a progressive presentation of shots with contrasting visual data content to express the sense of fast movement and achieve strong emotional impact. This type of sequence of shots would most likely be found in a scene where there is a rapid unfolding of the story, where the camera is not fixed at a location or following an event, and where there is a significant amount of object movement. In such a sequence, there is typically little or no recurrence of shots taken from the same camera or of the same person or background locale. A model can be constructed to capture the lack of repetition of labels. In addition, further classifications can consider shot durations. For example, one could characterize an action sequence with many shots of short duration as a "fast action" sequence.

The algorithms reported in [43] were tested on more than four hours of video materials of different types. It is reported that dialogue and action events constitute an average of about 50–70% of each of the video programs. In programs such as sitcoms and talk shows, there is a high percentage of dialogue. On the other hand, the percentage of action events is significantly higher in action movies.

The problem of recognizing certain temporal events can also be seen as a shot-grouping and classification problem: i.e., given a sequence of shots $s_1, s_2, s_3, \cdots, s_m$ and class labels $K_1, K_2, \cdots, K_M$, assign class K_i (or possibly more than one class label) to a subsequence of shots $s_j, s_{j+1}, \cdots, s_k$. To facilitate such a task, one can use the label L_i assigned to shot s_i, as discussed in Section 3.4. Identifying dialogue and action events can thus be treated as a grouping and classification problem with three classes: K_1 = "action," K_2 = "dialogue," and K_3 = "other."

In speech processing, hidden Markov models (HMMs) [51] have been successfully used for word recognition and speaker identification, both of which are classification problems. Such techniques could also be applied to video classification. Here, the challenges are in the use of appropriate features (for video shots, labels L_i), the judicious choice of model, and the assignment of probability distribution.

Another example of such work can be found in [52]. Here hidden Markov models of the one-dimensional sequence of shot durations are used to detect inserted commercial material and actionlike sequences within long video sequences.

3.7 *Visual summary for browsing of video*

Visual summaries give the end user the ability to quickly get an idea of what a particular candidate is about. They allow the user to flexibly refine the query to get a quick idea whether the query has been posed correctly, and they allow rapid nonlinear viewing of the video. That is, the visual summary permits random video access and gives the user an idea of the visual content of a possibly lengthy video clip.

A compact representation of video content and structure called the *scene-transition graph* (STG) [18] can be built on clusters of similar video shots. This representation is a form of visual summary that maps a sequence of video shots onto a two-dimensional display on the screen. The graph representation has nodes, which capture the temporal video content information, and directed edges, which depict the temporal flow of the story.

Formally, an STG is denoted $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$, where $\mathcal{V} = \{V_i\}$ is the node set, $\mathcal{E}$ is the edge set, and $\mathcal{F}$ is a mapping that partitions the set of shots $\{s_i\}$ into $V_1, V_2, \cdots$, the members of $\mathcal{V}$. For given nodes U and W in $\mathcal{V}$, (U, W) is a member of $\mathcal{E}$ if there exists some s_l in U and an s_m in W such that $m = l + 1$. That is, a directed edge is drawn from node U to node W if there is a shot in node U that immediately precedes a shot in node W. The set of shots $\{s_i\}$ is partitioned by $\mathcal{F}$ into the nodes $V_1, V_2, \cdots$ of $\mathcal{G}$, so that shots in each V_i are sufficiently close according to some similarity measures (e.g., distance measures between color histograms) such as those studied in [41].

A simple example would be a video clip of a conversation between two persons in which the camera alternates between shots of each person. Here, the graph $\mathcal{G}$ consists of two nodes, V_1 and V_2, and edges (V_1, V_2) and (V_2, V_1) are both members of $\mathcal{E}$. Establishing shots and other types of interspersed shots would result in additional nodes, possibly with only one member shot.

Figure 4 shows an STG of the IBM commercial "France" in which there are four nodes. The window at the bottom shows a linear view of the commercial. Each image icon represents a video shot. The content of a node consisting of four different shots of the two characters in the cast is shown in another window. Such a representation allows a user to first get a quick overview of the underlying content and then zoom in on items of interest. It is a time-oriented multi-resolution visual presentation (with views of the entire clip, story units, and shots) that lends itself very well to programming material that tells a story.

The STG together with time-constrained clustering of the video shots allows the segmentation of a video into story units based on "cut-edges" of the graph [17]. The story units are shown to be identical to those found using the method described in Section 3.5 [53]. The STG can thus represent the progression of the story, with each story unit being represented by a connected subgraph and connected to the next story unit via a cut-edge. Within each subgraph, the image content and temporal structure of the story unit are compactly represented through the nodes and edges. An STG depicting the story units and flow of the story for a half-hour sitcom or movie segment can, in general, be laid out and easily displayed on a single computer screen (for examples, see [17, 53]). A user can thus look at a single screen to get a quick overview of a half hour of video material.

Another form of visual summary, called the *pictorial summary*, is introduced in [54]. A pictorial summary is a sequence of representative images arranged in temporal order. Each representative image, called a *video poster*, consists of one or more subimages, each of which represents a unique "dramatic element" of the underlying story, such that the layout of the subimages emphasizes the dramatic "importance" of individual story units. The pictorial summary is a general scheme of visual presentation, and each video poster visually summarizes the dramatic events taking place in a meaningful story unit of the story.

Visual summaries, like the STG and pictorial summary, are means for condensing the visual content and representing the temporal flow of video that has underlying story structures (e.g., sitcoms, movies, news, and documentaries). In video of sports, a different type of visual summary would be needed, such as the one shown in Figure 2, which depicts a series of plays for a single possession in a football game. As we use more domain knowledge of the video content, we can create visual summaries that better capture the essence of the video for presentation. Visual summaries are crucial for fast and effective filtering of results returned from a video search at the browsing stage and are essential for efficient bandwidth utilization during the transmission of the

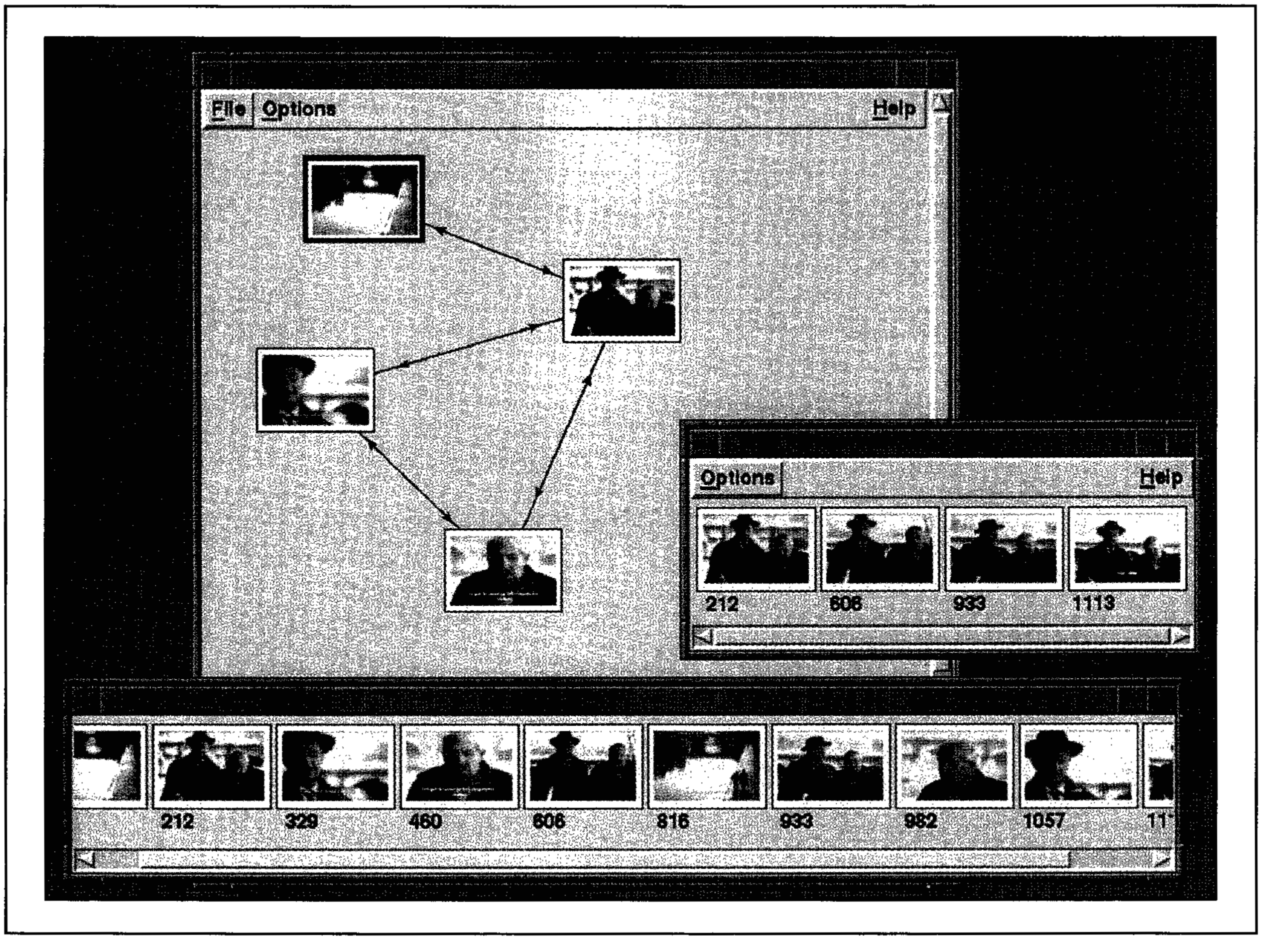

Figure 4

Scene transition graph for the television commercial "France."

results. That is, during browsing, pictorial descriptions with low bit content are transported over the network, rather than low- or high-resolution data being streamed.

3.8 Automatic generation of attributes

The features that are discovered by the algorithms described in the preceding sections can provide high-level annotation of video. The annotations form the attributes that are attached to the different levels of the hierarchy of video representation and can be used for querying purposes (e.g., see Figure 1). For example, for a video sequence of a half-hour sitcom, the story units partition the video into a small number of meaningful units, each of which represents a dramatic event or a locale. Furthermore, the semantic classification of events occurring within each story unit forms the attributes of the story unit (e.g., a certain story unit might have three dialogue events, one lasting for more than three minutes and two short ones of less than thirty seconds). At the lowest level of the hierarchy are the attributes of each shot. Two types of information can be extracted from each shot: temporal and static. Temporal properties are the consequences of time-varying characteristics of a shot and can be derived from motion analysis, object tracking, changes of color characteristics, etc. Static properties include the properties of the shot itself (e.g., duration) and those of its images (e.g., keyframes or mosaics), such as color histograms, texture patterns, and number of human faces.

Beyond visual features, information from the bibliographic data, audio track, and closed-caption can also provide textual annotation. Audio properties of a shot can be static or dynamic. For example, word-spotting techniques (defining a thesaurus of words and processing the speech to recognize only these words) can be used to identify certain keywords of interest; speech analysis can be performed to identify speakers; audio analysis can be used to classify the audio characteristic of a video segment (e.g., music, cheers, speech, silence). All of these derived attributes are then combined with the visual attributes.

The techniques based on between-shot analysis provide high-level attributes useful for early stages or iterations of the query-formulation process. Image search based on shot-level details is more useful in the later stages. If high-level attributes are concentrated on initially, the amount of video data becomes much more manageable. After the candidate list has been narrowed down to a reasonable number, the type of images that represent the keyframes (or mosaics) will be sufficiently reduced that it is unlikely that unreasonable (or random) results will occur. Then, image search is feasible for further reducing the result list. Excellent work on image search and related topics is published in [15, 16].

4. Conclusions and discussions

We have proposed a framework for video retrieval and query formulation that is based on an iterated sequence of *navigating*, *searching*, *browsing*, and *viewing*. This framework calls for certain capabilities of video-query systems—in particular, search on dynamic visual properties of video and the ability for rapid nonlinear viewing of video. To achieve these capabilities, algorithms have to be developed for automatic extraction of dynamic visual video properties and for processing of video to extract visual features for compact presentation of the video content. Such representations facilitate nonlinear access into video and give quick views of the visual content of video. The algorithms must involve the type of processing we have called between-shot video analysis.

To give instances of the type of algorithms that are needed, we have "presented" some examples—not as a complete survey, but as a demonstration of the type of algorithms that must be developed. The algorithms described have been tested by processing more than 20 hours of video having a variety of video categories, with encouraging results. Nonetheless, further testing is needed for performance evaluations of the automatic video-processing systems. (The testing of video-processing algorithms presents many problems: storage of the video is expensive, the computational requirements may be high, and evaluation of the results is difficult since the "true" annotation of the video is not available and is expensive and time-consuming to generate. Nonetheless, extensive testing should be encouraged and should be emphasized.)

On the other hand, we have to evaluate the search results of video query. Accurate content-processing and annotation, whether it is automatic or manual, may not imply the retrieval of desired segments—in other words, may not relate directly to good query results. We have mentioned the possibility of different types of queries expected from different types of end users: one who may have seen a particular piece of video and wants to find it or may not have seen it but knows that the video exists; and one who may not have seen the piece but only has some vague idea what he or she is looking for. In either case, one could define what it would mean for a retrieved video to be appropriate to the query. For the first case, a candidate can be appropriate only if it is the particular video clip or if the candidate has the clip embedded in it in some form. For the second case, the concept of appropriateness is less clear, and may be something best left up to the user to decide.

However, even without a proper definition, we can discuss some of the issues involved in search-performance evaluation. Here we define some terms widely used in data retrieval and system performance literature [8, 55]. An ideal retrieval system is one that retrieves all of the items or information appropriate to the user's need and which retrieves nothing else; furthermore, it must do so on every occasion. Retrieval effectiveness may be quantified. *Recall* is the ratio of a) appropriate items retrieved when a query is put to the system to b) the number of appropriate items in the system (not all of which will necessarily be retrieved). *Precision* is a measure which represents the ratio of a) appropriate items retrieved to c) the total number of items retrieved when the query is put to the system. What recall or precision do not consider are the number of inappropriate items that the system successfully manages not to retrieve, sometimes called *fallout*. Definitions of appropriateness of a returned item are given in terms of relevance and pertinence. Relevance, in practice, relates only to the particular query and not to the underlying need that caused the query. It is frequently assumed that candidate items are either relevant or not; more sophisticated approaches allow relevance to vary on a scale, say, between 1 and 10. *Pertinence* is also used to indicate the appropriateness of a search result—the difference between relevance and pertinence is that an item that is retrieved for the second time because the user issues the same or a similar query is not pertinent, because it is already known to the user.

The issues involved in evaluating the performance of a video-retrieval system are more complicated. The interactiveness of the query process makes it difficult to evaluate the performance of a system. The performance has to be measured not only in terms of recall and precision, but also in terms of the average length of time to select the segments of interest. The discipline of text retrieval has already given these issues some valuable thought, which will be useful for video-query performance evaluation. Nonetheless, the measures for performance of video query are still open research issues.

We believe that audiovisual features, together with textual descriptions, are integral components of video query. Analysis of visual content is one step beyond the traditional database query approach, analysis of audio features is another step forward, and the integration of the available multiple-media features is the ultimate step toward the successful deployment of video query in multimedia database systems.

Acknowledgments

The work reported in this paper was funded in part by the National Institute for Standards and Technology, the Advanced Technology Program (NIST/ATP) under Contract No. 70NANB5H1174, the Intel Foundation Graduate Fellowship, and the Wallace Memorial Fellowship of Princeton University. The support of the NIST program and the NIST Technical Program Manager, David Hermreck, Barbara Cuthil, and Omid Omidvar is greatly acknowledged. M. M. Yeung gladly acknowledges Frederick Mintzer of Image Library Applications at the IBM Thomas J. Watson Research Center for his support of the work. We also thank Chitra Dorai for reading the earlier drafts of this paper, the anonymous reviewers for their thoughtful comments and suggestions, and Sharath Pankanti for his help in preparing the artwork.

**Trademark or registered trademark of Moving Picture Expert Group.

References

1. D. Anastassiou, "Digital Television," *Proc. IEEE* **82,** No. 4, 510–519 (April 1994).
2. L. A. Rowe, J. S. Boreczky, and C. A. Eads, "Indices for User Access to Large Video Database," *Storage and Retrieval for Image and Video Database II, Proc. SPIE* **2185,** 150–161 (February 1994).
3. G. Davenport, T. A. Smith, and N. Pincever, "Cinematic Primitives for Multimedia," *IEEE Computer Graph. & Appl.* **5,** No. 4, 67–74 (July 1991).
4. R. Weiss, A. Duda, and D. Gifford, "Composition and Search with a Video Algebra," *IEEE Multimedia,* pp. 12–25 (Spring 1995).
5. E. Oomoto and K. Tanaka, "OVID: Design and Implementation of a Video-Object Database System," *IEEE Trans. Knowledge & Data Eng.* **5,** No. 4, 629–643 (August 1993).
6. S. Hibino and E. A. Rundensteiner, "A Visual Query Language for Identifying Temporal Trends in Video Data," *Proceedings of the International Workshop on Multi-Media Database Management Systems,* Blue Mountain Lake, NY, 1995, pp. 74–81.
7. R. Jain and A. Hampapur, "Metadata in Video Databases," *ACM SIGMOD* **23,** No. 4, 27–33 (December 1994).
8. I. Witten, A. Moffat, and T. Bell, *Managing Gigabytes: Compressing and Indexing Documents and Images,* Van Nostrand Reinhold, New York, 1994.
9. Special issue on metadata for digital media, *ACM SIGMOD* **23,** No. 4 (December 1994).
10. W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloustos, and G. Taubin, "The QBIC Project: Querying Images by Content Using Color, Texture and Shape," *Storage and Retrieval for Image and Video Databases, Proc. SPIE* **1908,** 13–25 (1993).
11. E. Chan, S. Garcia, and S. Roukos, "TREC-5 Ad Hoc Retrieval Using K Nearest Neighbors Re-scoring," *Proceedings of NIST TREC-5, 1996 (Text REtrieval Conference),* Gaithersburg, MD, November 1996; available on-line: *http://trec.nist.gov/pubs/trec5/t5_proceedings.html.*
12. S. Eisenstein, *The Film Sense,* Harcourt, Brace & Company, New York, 1970.
13. S. Eisenstein, *The Film Form—Essays in Film Theory,* Harcourt, Brace & Company, New York, 1977.
14. E. Pincus and S. Ascher, *The Filmmaker's Handbook,* New American Library (Division of Penguin Books), New York, 1984.
15. IS&T/SPIE, *Storage and Retrieval for Still Image and Video Databases I–V,* 1993–1997.
16. IEEE, *Proceedings of the 13th International Conferences on Multimedia Computing and Systems,* 1994–1996.
17. M. M. Yeung and B. L. Yeo, "Time-Constrained Clustering for Segmentation of Video into Story Units," *Proceedings of the 13th International Conference on Pattern Recognition,* August 1996, Vol. C, pp. 375–380.
18. M. M. Yeung, B. L. Yeo, W. Wolf, and B. Liu, "Video Browsing Using Clustering and Scene Transitions on Compressed Sequences," *Multimedia Computing and Networking 1995, Proc. SPIE* **2417,** 399–413 (February 1995).
19. G. K. Wallace, "The JPEG Still Picture Compression Standard," *Commun. ACM* **34,** No. 4, 30–44 (April 1991).
20. ISO, MPEG-1 Standard, "Coding of Moving Pictures and Associated Audio for Digital Storage Media up to 1.5 Mbits/s," *ISO/IEC JTC1 CD 11172,* ISO Central Secretariat, 1, rue de Varembé, Case postale 56, CH-1211 Genève 20, Switzerland, 1992.
21. ISO, MPEG-2 Standard, "Generic Coding of Moving Pictures and Associated Audio," *ISO/IEC JTC1 CD 13818,* ISO Central Secretariat, 1, rue de Varembé, Case postale 56, CH-1211 Genéve 20, Switzerland, 1994.
22. A. Nagasaka and Y. Tanaka, "Automatic Video Indexing and Full-Motion Search for Object Appearances," *Proceedings of the IFIP TC2/WG2.6 Second Working Conference on Visual Database Systems,* Budapest, September 30–October 3, 1991, pp. 113–127.
23. K. Otsuji, Y. Tonomura, and Y. Ohba, "Video Browsing Using Brightness Data," *Visual Communications and Image Processing, Proc. SPIE* **1606,** 980–989 (1991).
24. H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic Partitioning of Full-Motion Video," *Multimedia Syst.* **1,** 10–28 (July 1993).
25. A. Hampapur, R. Jain, and T. Weymouth, "Digital Video Segmentation," *Proceedings of the Second ACM International Conference on Multimedia,* San Francisco, August 1994, pp. 357–364.
26. P. Aigrain and P. Joly, "The Automatic Real-Time Analysis of File Editing and Transition Effects and Its Applications," *Computer & Graph.* **18,** No. 1, 93–103 (January 1994).
27. B. Shahraray, "Scene Change Detection and Content-Based Sampling of Video Sequences," *Digital Video Compression: Algorithms and Technologies, Proc. SPIE* **2419,** 2–13 (February 1995).
28. F. Arman, A. Hsu, and M. Y. Chiu, "Image Processing on Compressed Data for Large Video Databases," *Proceedings of the First ACM International Conference on Multimedia,* Anaheim, CA, August 1993, pp. 267–272.
29. J. Meng, Y. Juan, and S. F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," *Digital Video Compression: Algorithms and Technologies, Proc. SPIE* **2419,** February 1995, pp. 14–25.
30. I. K. Sethi and N. Patel, "A Statistical Approach to Scene Change Detection," *Storage and Retrieval for Image and Video Databases III, Proc. SPIE* **2420,** 329–338 (February 1995).

31. H. C. Liu and G. L. Zick, "Scene Decomposition of MPEG Compressed Video," *Digital Video Compression: Algorithms and Technologies*, *Proc. SPIE* **2419**, 26–37 (February 1995).
32. H. J. Zhang, C. Y. Low, and S. W. Smoliar, "Video Parsing and Browsing Using Compressed Data," *Multimedia Tools and Applications*, Kluwer Academic Publishers, Vol. 1, No. 1, March 1995, pp. 89–111.
33. B. L. Yeo and B. Liu, "Rapid Scene Analysis on Compressed Videos," *IEEE Trans. Circuits & Syst. for Video Technol.* **5**, No. 6, 533–544 (December 1995).
34. L. Teodosio and W. Bender, "Salient Video Stills: Content and Context Preserved," *Proceedings of the First ACM International Conference on Multimedia,* Anaheim, CA, 1993, pp. 39–46.
35. Y. Tonomura, A. Akutsu, K. Otsuji, and T. Sadakata, "VideoMAP and VideoSpaceIcon: Tools for Anatomizing Video Content," *Proceedings of ACM INTERCHI'93, Conference on Human Factors in Computing Systems*, Amsterdam, 1993, pp. 131–136.
36. H. S. Sawhney, S. Ayer, and M. Gorkani, "Model-Based 2D & 3D Dominant Motion Estimation for Mosaicking and Video Representation," *Proceedings of the IEEE International Conference on Computer Vision*, Boston, June 1995, pp. 583–590.
37. S. Mann and R. W. Picard, "Virtual Bellows: Constructing High Quality Stills from Video," *Proceedings of the IEEE International Conference on Image Processing,* 1994, pp. 363–367.
38. R. Szeliski, "Image Mosaicking for Tele-Reality Applications," *Technical Report CRL 94/2*, DEC Cambridge Research Laboratory, Cambridge, MA, 1994.
39. K. D. Yow, B. L. Yeo, M. M. Yeung, and B. Liu, "Analysis and Presentation of Soccer Highlights from Digital Video," *Proceedings of the Second Asian Conference on Computer Vision*, Vol. II, December 1995, pp. 499–503.
40. F. Arman, R. Depommier, A. Hsu, and M. Y. Chiu, "Content-Based Browsing of Video Sequences," *Proceedings of the Second ACM International Conference on Multimedia,* San Francisco, 1994, pp. 97–103.
41. M. M. Yeung and B. Liu, "Efficient Matching and Clustering of Video Shots," *Proceedings of the IEEE International Conference on Image Processing*, Vol. I, 1995, pp. 338–341.
42. R. M. Bolle, Y. Aloimonos, and C. Fermuller, "Video Representations," *Proceedings of the Second Asian Conference on Computer Vision*, Singapore, December 1995, pp. 24–28.
43. "Britannica Online," *http://www.eb.com:180/eb.html*, November 1995.
44. J. Miller, "Moving Pictures," *Images and Understanding*, H. Arlow, C. Blakemore, and M. Weston-Smith, Eds., Cambridge University Press, October 1986, pp. 180–194.
45. M. M. Yeung and B. L. Yeo, "Video Content Characterization and Compaction for Digital Library Applications," *Storage and Retrieval for Still Image and Video Databases V*, *Proc. SPIE* **3022,** February 1997, pp. 45–58.
46. H. J. Zhang, Y. H. Gong, S. W. Smoliar, and S. Y. Yan, "Automatic Parsing of News Video," *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, Boston, 1994, pp. 45–54.
47. D. Swanberg, C. F. Shu, and R. Jain, "Knowledge Guided Parsing in Video Databases," *Storage and Retrieval for Image and Video Databases*, *Proc. SPIE* **1908,** 13–25 (1993).
48. R. Mohan, "Text Based Indexing of TV News Stories," *Multimedia Storage and Archiving Systems II*, *Proc. SPIE* **2916,** 2–13 (1996).
49. B. Shahraray and D. Gibbon, "Automatic Generation of Pictorial Transcripts of Video Programs," *Multimedia Computing and Networking 1995*, *Proc. SPIE* **2417,** 512–528 (1995).
50. B. L. Yeo, *Efficient Processing of Compressed Images and Video*, available on-line at *http://www.ee.princeton.edu/-yeo/thesis*, Princeton University, Electrical Engineering Department, 1996.
51. L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proc. IEEE* **77,** No. 2, 257–286 (February 1989).
52. Y.-P. Tan and R. M. Bolle, "Binary Video Classification," *Research Report RC-21165* (Log No. 94593), IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 1998.
53. M. M. Yeung, "Analysis, Modeling and Representation of Digital Video," Ph.D. thesis, Princeton University, Electrical Engineering Department, 1996.
54. M. M. Yeung and B. L. Yeo, "Video Visualization for Compact Presentation of Pictorial Content," *IEEE Trans. Circuits & Syst. for Video Technol.* **7,** No. 5, 771–785 (October 1997).
55. D. A. Kemp, "Text Retrieval," *Aslib Information* **17,** No. 9, 211–213 (September 1989); see also D. A. Kemp, *Computer-Based Knowledge Retrieval*, Aslib, The Association for Information Management, Staple Hall, Stone House Court, London EC3A 7PB, U.K.; *aslib@aslib.co.uk,* 1998.

Received October 15, 1996; accepted August 27, 1997

Ruud M. Bolle *IBM Research Division, Thomas J. Watson Research Center, P.O. Box 218, Yorktown Heights, New York 10598 (bolle@watson.ibm.com).* Dr. Bolle received the Bachelor's degree in analog electronics in 1977 and the Master's degree in electrical engineering in 1980, both from Delft University of Technology, Delft, The Netherlands. In 1983 he received the Master's degree in applied mathematics and in 1984 the Ph.D. in electrical engineering from Brown University, Providence, Rhode Island. In 1984 Dr. Bolle became a Research Staff Member at the IBM Thomas J. Watson Research Center in the Artificial Intelligence Department of the Computer Science Department. In 1988 he became manager of the newly formed Exploratory Computer Vision group, which is part of IBM's Digital Library effort. Currently, his research interests are focused on video database indexing, video processing, and biometric applications: the multiple-modality document area. Dr. Bolle is a Fellow of the IEEE. He is on the Advisory Council of IEEE TPAMI, and he is Area Editor of *Computer Vision and Image Understanding* and an Associate Editor of the *Journal of Mathematical Imaging and Vision.* He is Guest Editor of a special issue on Computer Vision Applications for Network-Centric Computing of *Computer Vision and Image Understanding,* which will appear later in 1998.

Boon-Lock Yeo *Microcomputer Research Laboratories, Intel Corporation, 2200 Mission College Blvd., Santa Clara, California 95052 (Boon-Lock_Yeo@ccm.sc.intel.com).* Dr. Yeo received the B.S.E.E. degree in electrical engineering (with highest distinction) from Purdue University in August 1992, and the M.A. and Ph.D. degrees from Princeton University in June 1994 and January 1996, respectively. He received the 1996 IEEE Circuits and Systems Society Video Technology Transactions Best Paper Award, the Wallace Memorial Fellowship in Engineering (1995–96), and the IBM Graduate

Fellowship (1994–95) and was a Singapore Technologies Overseas Scholar (1989–91). He has held summer and visiting positions at C-Cube Microsystems, Siemens Corporate Research, and AT&T Bell Laboratories. In December 1995, Dr. Yeo joined the IBM Thomas J. Watson Research Center, Hawthorne, New York, as a Research Staff Member and later became a manager. Since January 1998, he has been with the Microcomputer Research Laboratories, Intel Corporation, Santa Clara, California, managing the Video Technology Department. Dr. Yeo is serving as a guest editor of a special issue of *Computer Vision and Image Understanding* on Computer Vision Applications for Network-Centric Computing. His research interests include signal and image processing, data compression and communications, visualization, computer vision, and problems related to multimedia information systems.

Minerva M. Yeung *Microcomputer Research Laboratories, Intel Corporation, 2200 Mission College Blvd., Santa Clara, California 95052 (Minerva_Yeung@ccm.sc.intel.com).* Dr. Yeung received the Ph.D. and M.A. degrees from Princeton University in 1996 and 1994, respectively, and a B.S.E.E. (with highest distinction) from Purdue University in 1992. From August 1996 to December 1997, she was a Research Staff Member at the IBM Thomas J. Watson Research Center in the Image Applications group. Dr. Yeung joined the Microcomputer Research Laboratories of Intel Corporation, Santa Clara, California, in January 1998. She is currently leading the efforts to build research projects in image processing, and applications of digital image and video. Dr. Yeung is a guest editor of a special issue of the *Communications of the ACM* on digital watermarking and an associate chair of ACM Multimedia '98. She is involved in the program committees of several major conferences in image processing and computer vision. Dr. Yeung has co-authored about 30 papers, holds one patent, and has seven pending applications in the area of video processing, presentation, and digital watermarking. Her research interests are in the general areas of image/video processing, watermarking, computer–human interaction, and database retrieval.