SOUTH AFRICA/HOLTON COLLECTION/SUPERSTOCK

Rock Painting,
South Africa

Pulling together clips of a feature film selected for such content as text, dialogs, and explosions yields a video abstract that can be put onto an html page for browsing.

VIDEO ABSTRACTING

Rainer Lienhart, Silvia Pfeiffer, and Wolfgang Effelsberg

VISUAL INFORMATION MANAGEMENT

The abstract of an article is a short summary often used to pre-select material relevant to the reader. In this case, abstract and document are the same medium, namely text. In the age of multimedia, it would be desirable to use video abstracts in a similar way—as short clips containing the essence of a longer video, without a break in the presentation medium. However, the state of the art involves textual abstracts for indexing and searching large video archives.

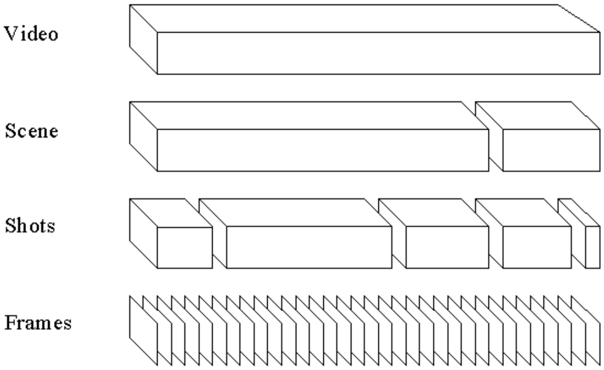

Figure 1. Our video structuring model, describing a digital video at four levels of detail: At the lowest, it consists of a set of frames; at the next higher, frames are grouped into *shots,* or continuous camera recordings, and consecutive shots are aggregated into *scenes* based on story-telling coherence. All scenes together make up the *video.* (A *clip* is a frame sequence selected to be an element of the abstract; a *video abstract* thus consists of a collection of clips.)

This media break is harmful, since it typically leads to the loss of information. For example, it is unclear to the person providing the abstract at what level of abstraction the textual description should be; if the video shows a famous politician at a dinner table with other politicians, what should the text say? The names of the people, their titles, the event, or just describe the scene as if it were a painting, emphasizing color and geometry? An audiovisual abstract, to be interpreted by a human user, is semantically much richer than a text. We define a video abstract as a sequence of moving images much shorter than the original yet preserves its essential message (see Figure 1).

The power of visual abstracts can be helpful in many application contexts.

Multimedia archives. With the advent of multimedia PCs and workstations, the World-Wide Web,

Step I	Step 2	Step 3
Video Segmentation and Analysis	Clip Selection	Clip Assembly
Extracts and analyzes • Shots and scenes • clips containing special events (close-ups of main actors, gunfire, explosions, text)	Determines video clips to be part of the abstract based on • general considerations (balanced coverage of entire input video) • special events	Assembles the selected video clips into their final format

Figure 2. The three abstracting steps

and standard video compression techniques, more and more video material is being digitized and archived worldwide. Wherever digital video material is stored, we can use video abstracts for indexing and retrieval. Online abstracts would, for example, support journalists searching old video material or producing documentaries. The Internet movie database called IMDb on the Web (http://uk.imdb.com/) is indexed on the basis of "handmade" textual information about the movies; sometimes, a short clip selected at random is also included. Such an index could be extended easily by adding automatically generated video abstracts.

Movie marketing. Trailers are widely used for movie advertising in movie theaters and on television. However, production of this type of abstract is costly and time-consuming. With our movie content analysis (MoCA) video abstracting system, anyone can produce trailers automatically.[1] In order to tailor a trailer to a specific audience, we would set certain parameters, such as the optimal amount of action or violence. Another possibility is a digital TV magazine; instead of reading short text descriptions of upcoming programs, a couch potato could view the abstracts without leaving the couch (assuming the potato has an integrated TV and Web browser). For digital video-on-demand systems, the content provider could supply video abstracts in an integrated fashion. These trailers could also be generated and played on terminals in video stores.

Home entertainment. If you miss an episode of your favorite television series, the abstracting system could perform the task of telling you briefly what happened in the missed episode. Many more innovative applications could be built around video abstracting. But what kind of algorithms and tools are needed to automatically produce a digital video abstract?

MoCA Video Abstracting

The purpose of an abstract varies widely; for example, viewers of documentaries may want to be told about all the content on the full-size video, whereas a Hollywood film trailer seeks to lure the audience into a movie theater. Thus, a documentary abstract should give a good overview of the contents of an entire video, whereas a movie trailer should be entertaining in itself while not revealing the end of the story.

When we began the MoCA project, we made a basic decision about the type of material we would use as input. For example, different types of material can be used for producing a movie trailer—unchanged material from the original movie, revised material, and outtakes not used in the movie's final version. However, we use only unchanged material from the original movie, enabling our system to work with any video archive, independent of additional sources of information.

The abstracting algorithm we developed can be subdivided into three consecutive steps (see Figure 2). In the first step, video segmentation and analysis, the input video is segmented into its shots and scenes. At the same time, frame sequences with special events, such as text appearing in the title sequence, close-up shots of the main actors, explosions, and gunfire, are identified. In the second, clip selection, video clips are selected for inclusion in the abstract. The third, clip assembly, assembles the clips into their final sequence and produces the presentation layout; this step involves determining the order of the video clips, the type of transitions between them, and other editing decisions.

Video Segmentation

A *shot* designates a video sequence recorded by a camera's uninterrupted operation. Neighboring shots are concatenated through such editing effects

[1] The MoCA project, begun in 1995 at the University of Mannheim in Germany, seeks ways to automatically determine the contents of digital video and experiment with innovative applications based on this content, including automatic recognition of film genres, detection and recognition of television commercials, determination of the period in which a feature film was produced, and video abstracting

as hard cuts, fades, wipes, and dissolves. Most editing effects result in characteristic spatiotemporal changes in subsequent frames of the video stream and can therefore be detected automatically. Various methods have been proposed and implemented successfully (see [3] for examples). In MoCA, we use the edge-change-ratio parameter, initially published in [12], for cut detection.

Several neighboring shots are usually used to build a larger story-telling unit, called a scene [1], an act, or just a cluster of shots. The clustering of shots is controlled by selectable criteria. To determine scene boundaries, we use several heuristics:

- Sequential shots with similar color content usually belong to a common scene because they share a common background [11]. The color content of the frames changes much more drastically at the end of a scene than within the scene. A change of camera angle usually has no influence on the main background colors.
- In different scenes, the audio content usually differs significantly. Therefore, a video cut not accompanied by an audio cut does not establish a scene boundary.
- Consecutive shots are grouped into a scene if the shots can be identified as representing a dialog.

We expect our tools to be used for large multimedia archives in which video abstracts would be a much more powerful browsing technique than textual abstracts.

Audio cuts are defined as time instances delimiting time periods with similar sound and are used to explore the similarity of the audio track in different shots. If there is no significant change in the audio track close to a video shot boundary, that is, if the sound continues across a video shot boundary, we consider both shots to belong to the same scene. Audio cuts are determined by calculating the frequency and intensity spectrum of each time window of the audio track, predicting its values for the next time window through exponential smoothing and declaring an audio cut to be where the current frequency and intensity spectrum deviate considerably from the prediction.

Finding Special Events

Once the video is segmented into its basic components, we identify semantically rich events, such as close-ups of main actors, gunfire, explosions, and text appearing in the video. These events help us select the sequences of frames for our clips that are important for the abstract.

Finding actors' faces and identifying dialog. In many video genres, the cast is an essential piece of information, particularly in feature films. Our abstracting system has to understand where the main actors appear in the video. Therefore, we use a face-detection algorithm and a method for recognizing the face of the same actor again, even across shot boundaries.

An excellent face-detection algorithm developed by Rowley, Baluja, and Kanade [10] recognizes about 90% of all upright and frontal faces in images (photos, newspapers, and single video frames) while hardly ever identifying non-face regions of a frame as a face. The basic idea is to train a neural network with hundreds of example faces in which the eyes and the nose are manually marked. After the learning phase, the neural network can reliably detect new faces in arbitrary images.

We have implemented our own neural network and trained it with approximately 1,000 faces in much the same way. To increase the range of detectable faces, our implementation also searches for slightly tilted faces. This modification was necessary because the faces of actors in videos are rarely upright, in contrast to faces in still images. To speed processing, we pass to the face detector only frame regions in which the pixel colors are close to human skin colors. This filter reduces the number of candidate face regions by more than 80%. Moreover, face detection is run only on every third frame of the video sequence. The result is a set of regions in frames in which faces appear.

So far, each detected face is isolated and unrelated to other faces in the video. The next task is to classify frames with similar faces in order to find groups of frames showing the same actors. Such a group of related frames is called a *face-based class*. We are interested only in the main actors and therefore consider only faces larger than 30% of the frame size, that is, faces in close-up shots. In a first step, faces within shots are related to each other according to the similarity of their positions and sizes in neighboring frames, assuming that these features change only slightly from frame to frame. This assumption is especially true for dialog scenes. In addition, we dispose of accidental misclassifications of the face detector by discarding all face-based classes with fewer than three occurrences of a face and by allowing up to two dropouts in the face-tracking process. In a second step, face-based classes with similar faces are merged by face recognition algorithms [4] to obtain face-based classes as large as possible.

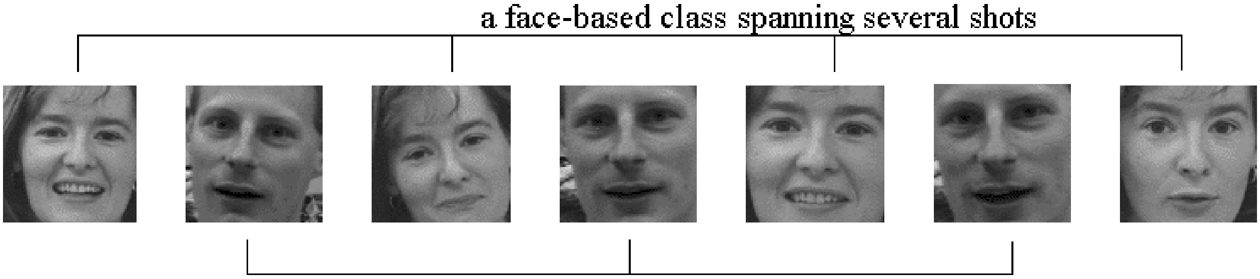

Figure 3. A dialog and its face-based classes

The same face recognition algorithms are used to identify and merge face-based classes of the same actor across shots throughout the video, resulting in so-called *face-based sets*. There is a face-based set for each main actor, describing where, when, and at what size that actor appears in the video.

It is now easy to detect typical shot/reverse-shot dialogs and multi-person dialogs. We search for sequences of face-based classes, close together in time, with shot-overlapping face-based classes of the same actor and crossover relations between different actors. For example, a male and a female actor could appear in an m-f-m-f sequence. See Figure 3 for an example of a dialog automatically detected in this way.

Extracting text from the title sequence. In the opening sequence of a feature film, important information appears in the form of text. Examples include the title and the names of the main actors. Both types of information should be stored with the video abstract itself, as well as in a search index for a set of abstracts. For this purpose, we use our own text segmentation and text recognition algorithms described in [5].

The text segmentation step results in a list of text regions per frame and a list of their motion paths throughout the sequence. In order to extract the bitmaps of the title and the names of the main actors, character regions within each frame are clustered into words or text lines based on their horizontal distance and vertical alignment. Next, the clusters connected via the motion path of at least one character region are combined into a text line representation. For each text line representation, a time-varying (one per frame) bounding box is calculated. The content of the original video framed by the largest bounding box is chosen as the representative bitmap of the text line. This method works well under the following assumptions:

- The text line is stationary or moving linearly.
- All characters of a cluster are contained in the segmented text regions for at least one frame.

We have found that these assumptions are true for most occurrences of text in feature films. The largest bounding box will then enclose the text, and we can perform OCR-style text recognition on the box in

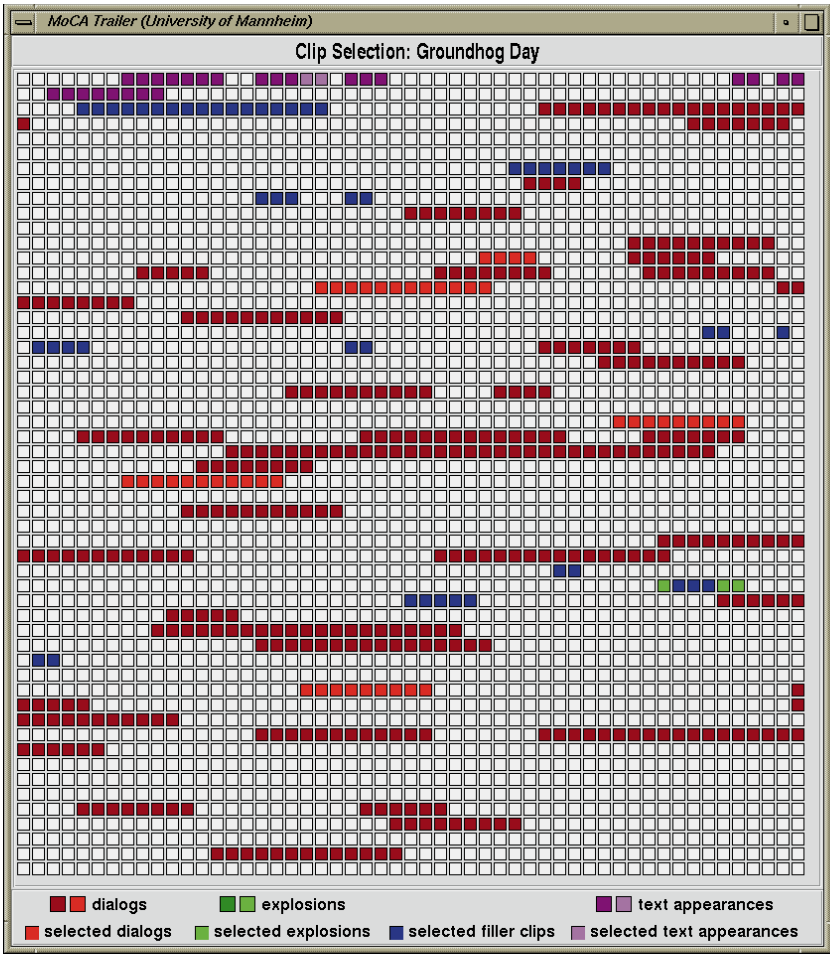
MoCA Trailer (University of Mannheim)
Clip Selection: Groundhog Day
dialogs
explosions
text appearances
selected dialogs
selected explosions
selected filler clips
selected text appearances

Figure 4. The temporal distribution of the detected video and audio events in the movie "Groundhog Day," along with the events chosen during the clip-selection process to be part of the trailer. Note: Since "Groundhog Day" is not an action movie, there are only two explosions and no gunfire. Each box represents two seconds (2,828 boxes total). Time passes from left to right and top to bottom.

order to translate the recognized text into ascii.

Automatic detection of the movie title is most desirable. A simple heuristic is that the title can be distinguished from other text in the opening sequence because it is centered on the screen and is in the largest font size or the longest text line. This heuristic allows us to automatically extract the title in many practical cases.

Identifying gunfire and explosions. Attracting the user's eye is an important design criterion for an abstract of a feature film. Action films often contain explosions and gunfire—events that can be recognized automatically. Distribution of the audio parameters loudness, frequencies, pitch, fundamental frequency, onset, offset, and frequency transition are calculated for short time windows of the audio track. For each time window, we compare distribution of the indicators with a database of known distributions for explosions and gunfire. If the distribution is found in the database, gunfire or explosions are recognized [7].

We used these various functions to analyze the movie "Groundhog Day;" Figure 4 shows the general distribution of detected special events.

Generating a Video Abstract

A movie trailer is a short appetizer for a movie, intended to attract the viewer's attention. Trailers require inclusion of eye-catching clips into the abstract. We again use heuristics over the basic physical parameters of the digital video to select the clips for our trailer:

- *Important objects and people*. The most important objects and actors in the original video should also appear in the trailer. Starring actors are especially important, since potential viewers often have preferences for specific actors.
- *Action*. If the film contains explosions, gunfire, car chases, or violence, some of these events should be in the trailer. They attract attention and make viewers curious.
- *Dialog*. Short extracts from dialog scenes with a starring actor stimulate viewer fantasy and often carry important messages.
- *Title text and title music*. The title text and parts of the title music should be included in the trailer. The names of the main actors from the opening sequence can also be shown.

A special feature of our trailer-generation technique is that the end of the movie is not revealed; we do not include clips from the last 20% of the movie, guaranteeing that we don't neutralize the suspense.

Clip selection. The user of our abstracting system can specify a target length not to be exceeded by the video abstract. When selecting clips, the system has to come up with a compromise between the target length and the heuristics. This compromise is done iteratively. Initially, all scenes from the first 80% of the movie are in the scene candidate set. All decisions have to be based on the physical parameters of the video, because only they can be derived automatically. Thus, the challenge is to determine relevant scenes, and a good clip as a subset of the frames of each relevant scene, based on computable parameters.

Table 1. Edits in an abstract

	Event Clips	Dialog Clips	Other Clips
Event Clips	hard cut	hard cut	hard cut
Dialog Clips	hard cut	dissolve, wipe, fade	hard cut, dissolve, wipe, fade
Other Clips	hard cut	hard cut, dissolve, wipe, fade	hard cut, dissolve, wipe, fade

We use two different mechanisms for selecting relevant scenes and clips. The first extracts special events and texts from the video, such as gunfire, explosions, cries, close-up shots, dialogs with main actors, and title text. These events and texts summarize the video well and are suited for attracting viewer attention. Identification of the relevant sequences of frames is based on the algorithms and is fully automatic.

The user can specify the share of special events in the abstract. In our experiments, we set it at 50%. If the total length of special-event clips is longer than desired, scenes and clips are chosen uniformly and randomly from the different types of events. The title text, however, is always contained in the abstract.

The second mechanism adds filler clips from different parts of the movie to complete the trailer. To do so, the remaining scenes are divided into several non-overlapping sections of about the same length. We have used eight sections in our experiments. The number of clips and their total length within each section are determined first. Clips are then selected repeatedly from the sections with the lowest share in the abstract so far, until the target length of the trailer is reached. This mechanism ensures good coverage of all parts of the movie even if special events occur only in some sections.

Clips should generally be much shorter than scenes. But how can a clip be extracted from a scene? We have tried out two heuristics. With the first, we pick the shots with the most action and with the same basic color composition as the average throughout the movie [8]. Action is defined through motion (object motion or camera motion), and the amount of motion in a sequence of frames can be computed easily based on motion vectors or on the edge-change ratio. The action criterion is motivated by the fact that action clips are often more interesting and carry more content in a short time than calm clips. The idea behind the color criterion is that colors are an important component for perceiving a video's mood and that color composition should thus be preserved in the trailer.

The second heuristic takes a completely different approach, using the results of our MoCA genre recognition project. The project's basic goal is to compute a large number of audiovisual parameters from an input video and use them to classify the video into a genre, such as newscast, soccer, tennis, talk show, music clip, cartoon, feature film, or commercial. The classification is based on characteristic parameter profiles, derived in advance and stored in a database. The results of the project can now be used to select clips for the trailer in a more sophisticated way. The clips closest in parameter values to the characteristic profile of the entire movie are selected. The advantage of this clip-selection process is that it automatically tailors the selection process to a specific genre provided we have a characteristic parameter profile for it.

Clip assembly. In the assembly stage, the selected video clips and their respective audio tracks are composed into the final form of the abstract. We have experimented with two degrees of freedom in the composition process: ordering and edits (types of transition) between the clips.

Ordering. Pryluck, Teddlie, and Sands [9] showed that the sequencing of clips strongly influences the viewer's perception of their meaning. Therefore, the ordering of the clips must be done very carefully. We first group the video clips into four classes. The first, also called the *event class*, contains the special events, currently gunfire and explosions. The second consists of dialogs. The filler clips constitute the third class. The extracted text (in the form of bitmaps and ascii text) falls into the fourth class. Within each class, the original temporal order is preserved.

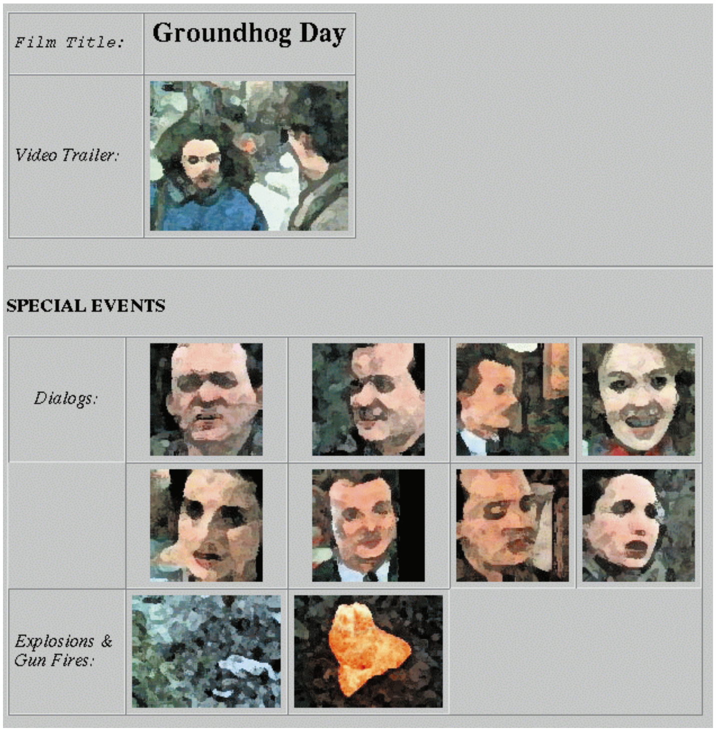
Film Title: Groundhog Day

Figure 5. Result of video abstracting compiled into an html page

Dialogs and event clips are assembled in turn into so-called *edited groups*. The maximum length of an edited group is 25% of the length of the total share of special events. The gaps between the edited groups are filled with the remaining clips, resulting in a preliminary abstract.

The text occurrences in the fourth class usually show the film's title and the names of the main actors. The title bitmap is always added to the trailer, cut to a length of one second. The actors' names can also be added to the trailer.

Edits. We apply three different types of video edits in the abstract: hard cuts, dissolves, and wipes. Their use is based on general rules derived from knowledge elicited from professional video and film editors [6]—a research field in its own right. As a preliminary solution, we found it reasonable to concatenate special-event clips with every other type of clip by means of hard cuts and to insert soft cuts (dissolves and wipes) between calmer clips, such as dialogs. Table 1 shows possible use of edits in the various cases. For automatic video editing of humorous themes, a much more sophisticated approach can be found in [6].

Audio editing is much more difficult. A first attempt to concatenate the soundtrack segments of the selected clips produced terrible audio. It is especially important in dialog scenes that audio cuts have priority over video cuts. Construction of an abstract's audio track is currently performed as follows:

- The audio segments of special-event clips are used as they are in the original.
- The audio of dialogs respects the audio cuts in the original. The audio of every dialog is cut in length as much as necessary to fill the gaps between the audio segments of the special events. Audio dissolves are the primary means of concatenation.
- The entire audio track of the abstract is underlaid by the title music. During dialogs and special events, the title music is reduced in volume.

We plan to experiment with speaker recognition and speech recognition to be able to use higher-level semantics from the audio stream. The combination of speech recognition and video analysis is especially promising.

Experimental Results

To evaluate the MoCA video abstracting system, we ran a series of experiments with video sequences recorded from German television. We quickly found that there is no absolute measure for the quality of an abstract; even experienced movie directors told us that making good trailers for a feature film is an art, not a science. It is interesting to note that the shots extracted by a human for an abstract depend on the purpose of the abstract. For example, a trailer for a movie often emphasizes thrill and action without giving away the ending; a preview for a documentary on television attempts to capture the essential contents as completely as possible; and a review of last week's soap opera highlights the most important events of that particular episode. We conclude that automatic abstracting should be controlled by a parameter describing the purpose of the abstract.

Comparing the abstracts generated by our system with commercial abstracts, we found no obvious difference in quality (at least within the picture track; the audio track of the commercial abstracts usually contains material originally not part of the video). In the case of the reviews for last week's episode of a television series, the scenes generated by our tool were similar to those shown on television.

Since there is no mathematical measure for the quality of a video abstract, we presented the abstracts to a set of test subjects. Even if the generated abstracts were quite different from those made directly by humans, the subjects could not tell which were better [8].

For browsing and searching large information archives, many users are familiar with Web interfaces. Therefore, our abstracting tool can compile its results into an html page, including the anchors for playing short video clips (see Figure 5). The top of the page in Figure 5 shows the film title, an animated gif image constructed from the text bitmaps (including the title), and the title sequence as a video clip. This information is followed by a randomly selected subset of special events, which are followed by a temporally ordered list of the scenes constructed by our shot-clustering algorithms. The bottom part of the page lists the creation parameters of the abstract, such as creation time, length, and statistics.

Video abstracting is a young research field. We would like to mention two other systems suitable for creating abstracts of long videos. The first is video skimming [2], which mainly seeks to abstract documentaries and newscasts. It assumes that a transcript of the video is available; the video and the transcript are then aligned by word spotting. The audio track of the video skim is constructed by using language analysis to identify important words in the transcript; audio clips around those words are then cut out. Based on detected faces [10], text, and camera operation, video clips are selected from the surrounding frames.

The second system is based on the image track only, generating not a video abstract but a static scene graph of thumbnail images on a 2D "canvas." The scene graph represents the flow of the story in the form of keyframes, allowing users to interactively descend into the story by selecting a story unit of the graph [11].

Conclusions

Using the algorithms discussed here for automatically generating video abstracts, we first decompose the input video into semantic units, called "shot clusters" or "scenes." Then we detect and extract semantically rich pieces, especially text from the title sequence and special events, such as dialog, gunfire, and explosions. Video clips, audio clips, images, and text are extracted and composed into an abstract. The output can then be compiled into an html page for easy access through browsers.

We expect our tools to be used for large multimedia archives in which video abstracts would be a much more powerful browsing technique than textual abstracts. For example, broadcast stations today sit on a gold mine of archived difficult-to-access video material. Another application of our technique could be to create an online TV guide on the Web, with short abstracts of upcoming shows, documentaries, and feature films. Just how well the generated abstracts capture the essentials of all kinds of videos remains to be seen in a larger series of practical experiments. C

References

1. Bordwell, D., and Thompson, K. *Film Art: An Introduction.* 4th ed. McGraw-Hill, New York, 1993.
2. Christel, M., Kanade, T., Mauldin, M., Reddy, R., Sirbu, M., Stevens, S., and Wactlar, H. Informedia digital video library. *Commun. ACM 38*, 4 (Apr. 1995), 57–58.
3. Dailianas, A., Allen, R., and England, P. Comparison of automatic video segmentation algorithms. In *Proceedings of SPIE 2615, Photonics East 1995: Integration Issues in Large Commercial Media Delivery Systems*, A. Tescher and V. Michael Bove, Eds. (Philadelphia, Oct. 22–26, 1995), SPIE, Bellingham, Wa., 1995, pp. 2–16.
4. Lawrence, S., Giles, C., Tsoi, A., and Back, A. Face recognition: A convolutional neural network approach. *IEEE Trans. Neural Networks 8*, 11 (Nov. 1997), 98–113.
5. Lienhart, R. Automatic text recognition for video indexing. In *Proceedings of ACM Multimedia 1996* (Boston, Nov. 18–22, 1996), ACM Press, New York, 1996, pp. 11–20.
6. Nack, F., and Parkes, A. The application of video semantics and theme representation in automated video editing. *Multimedia Tools Appl. 4*, 1 (Jan. 1997), 57–83.
7. Pfeiffer, S., Fischer, S., Effelsberg, W. Automatic audio content analysis. In *Proceedings of ACM Multimedia 1996* (Boston, Nov. 18–22, 1996), ACM Press, New York, 1996, pp. 21–30.
8. Pfeiffer, S., Lienhart, R., Fischer, S., and Effelsberg, W. Abstracting digital movies automatically. *J. Visual Commun. Image Represent.* 7, 4 (Dec. 1996), 345–353.
9. Pryluck, C., Teddlie, C., and Sands, R. Meaning in film/video: Order, time, and ambiguity. *J. Broadcast. 26* (1982), 685–695.
10. Rowley, H., Baluja, S., and Kanade, T. Human face recognition in visual scenes. Rep. CMU-CS-95-158R, School of Computer Science, Carnegie-Mellon Univ., 1995.
11. Yeung, M., Yeo, B.-L., and Liu, B. Extracting story units from long programs for video browsing and navigation. In *Proceedings of IEEE Multimedia Computing & Systems 1996* (Hiroshima, Japan, June 17–23 1996), IEEE Computer Society Press, Los Alamitos, Calif., 1996, pp. 296–305.
12. Zabih, R., Miller, J., and Mai, K. A feature-based algorithm for detecting and classifying scene breaks. In *Proceedings of ACM Multimedia 1995* (San Francisco,Nov. 5–9, 1995), ACM Press, New York, 1995, pp. 189–200.

Rainer Lienhart (lienhart@pi4.informatik.uni-mannheim.de) is a Ph.D. candidate and research assistant in the chair of Praktische Informatik IV at the University of Mannheim in Germany.
Silvia Pfeiffer (pfeiffer@pi4.informatik.uni-mannheim.de) is a Ph.D. candidate and research assistant in the chair of Praktische Informatik IV at the University of Mannheim in Germany.
Wolfgang Effelsberg (effelsberg@pi4.informatik.uni-mannheim.de) is a professor of computer science and heads the chair of Praktische Informatik IV at the University of Mannheim in Germany.