Video Abstracting

Rainer Lienhart, Silvia Pfeiffer and Wolfgang Effelsberg

University of Mannheim, 68131 Mannheim, Germany
{lienhart, pfeiffer, effelsberg}@pi4.informatik.uni-mannheim.de

1. What is a Video Abstract?

We all know what the abstract of an article is: a short summary of a document, often used to preselect material relevant to the user. The medium of the abstract and the document are the same, namely text. In the age of multimedia, it would be desirable to use video abstracts in very much the same way: as short clips containing the essence of a longer video, without a break in the presentation medium. However, the state of the art is to use textual abstracts for indexing and searching large video archives. This media break is harmful since it typically leads to considerable loss of information. For example it is unclear at what level of abstraction the textual description should be; if we see a famous politician at a dinner table with a group of other politicians, what should the text say? Should it specify the names of the people, give their titles, specify the event, or just describe the scene as if it were a painting, emphasizing colors and geometry? An audio-visual abstract, to be interpreted by a human user, is semantically much richer than a text. We define a *video abstract* to be a sequence of moving images, extracted from a longer video, much shorter than the original, and preserving the essential message of the original.

2. Why are Video Abstracts useful?

The power of visual abstracts can be helpful in many application contexts. Let us look at some examples.

Multimedia archives. With the advent of multimedia PCs and workstations, the World Wide Web and standardized 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. For instance, the on-line abstracts could support journalists when searching old video material, or when producing documentaries. Another example is the Internet movie database IMDb on the Web (http://uk.imdb.com/). It is indexed on the basis of “hand-made” textual information about the movies; sometimes, a short clip, selected at random, is also included. The index could easily be extended by automatically generated video abstracts.

Movie marketing. Trailers are widely used for movie advertising in cinemas and on television. Currently the production of this type of abstract is quite costly and time-consuming. With our system we could produce trailers automatically. In order to tailor a trailer to a specific audience, we would set certain parameters such as the desirable amount of action or of violence.

Another possibility would be a digital TV magazine. Instead of reading short textual descriptions of upcoming programs you could view the abstracts without even having to get up from your couch (supposing you have an integrated TV set and Web browser). And for digital video-on-demand systems the content provider could supply video abstracts in an integrated fashion.

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 meantime”.

Many more innovative applications could be built around the basic video abstracting technique. But let us now come to the algorithms and tools we are using to automatically produce a digital video abstract.

3. The MoCA Video Abstracting System

Types of abstracts. The definition of a video abstract we gave above is very general. In practice the purpose of an abstract can vary widely; for example viewers of documentaries might want to be informed as well as possible of all the content of the full-size video whereas the aim of a Hollywood film trailer is to lure the audience into a movie theater. Thus a documentary abstract should give a good overview of the contents of the entire video whereas a movie trailer should be entertaining in itself, and it should not reveal the end of the story.

We describe a digital video at four different levels of detail. At the lowest level, it consists of a set of *frames.* At the next higher level frames are grouped into *shots*; the term shot refers to a continuous camera recording. Consecutive shots are aggregated into *scenes* based on story-telling coherence. All scenes together compose the *video*. Note that a video comprises both the image and audio tracks.

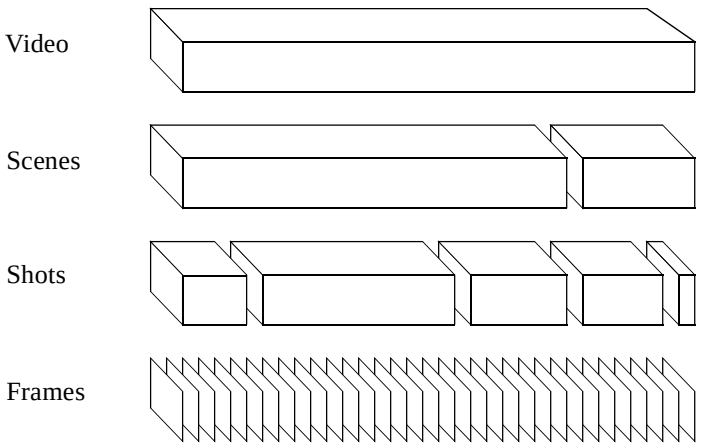

Figure 1: Our video structuring model

In this article, we call a frame sequence selected to become an element of the abstract a *clip.* A video abstract thus consists of a collection of clips.

Raw material. When we began with our movie content analysis (or MoCA) project we had to make a basic decision about the type of material that we would use as input. For example different types of material can be used for the production of a movie trailer: unchanged material from the original movie, revised material, and/or cut-out material that was not used in the final version of the movie. In our project, we use only unchanged material from the original movie. This enables our system to work with any video archive, independent of additional sources of information.

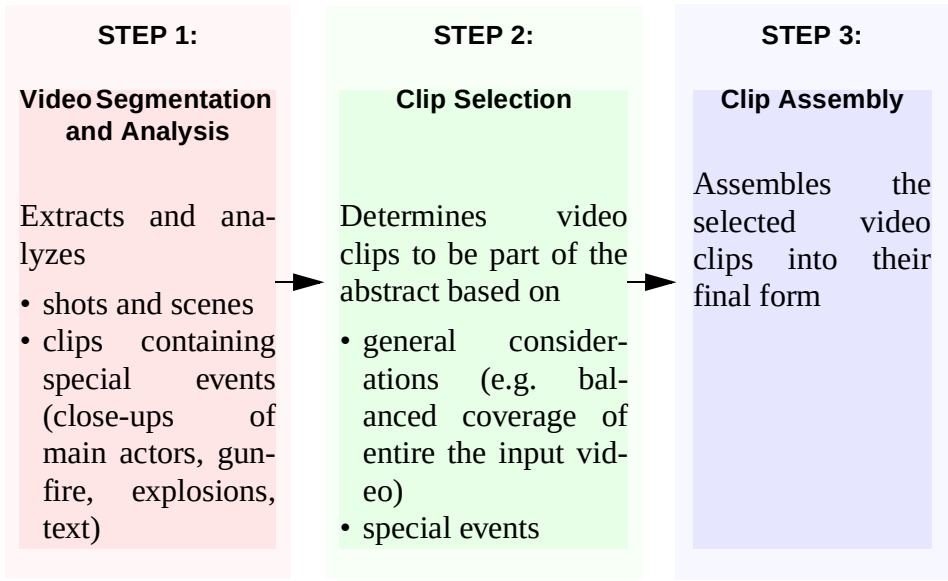

Figure 2: Overview of the three abstracting steps

Our Approach. The abstracting algorithm we have developed can be subdivided into three consecutive steps (see Figure 2). In step 1, *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, gun fires, etc. are determined. In the second step, *clip selection,* those video clips are selected which will become part of the abstract. The third step, the *clip assembly,* assembles the clips into their final form and produces the presentation layout. This implies determining the order of the video clips, the type of transitions between them, and other editing decisions.

We will now discuss the algorithms we have developed for each of these steps in detail.

Shot determination. A *shot* designates a video sequence which was recorded by an uninterrupted camera operation. Neighboring shots are concatenated by editing effects such as hard cuts, fades, wipes or dissolves. Most of the editing effects result in characteristic spatio-temporal 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 our MoCA system, we decided to use the edge change ratio parameter for cut detection, initially published in [12].

Scene determination. Usually, several neighboring shots are used to build a larger story-telling unit. The larger unit is called *scene*, *act* or just a *cluster of shots*. The clustering of shots is controlled by selectable criteria. Here are the heuristics we use for determining scene boundaries:

- Sequential shots with very 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

The MoCA Project

MoCA stands for *Movie Content Analysis*, a project started in 1995 at the University of Mannheim in Germany. The aim of the project is to automatically determine the contents of digital video, and to experiment with innovative applications based on this knowledge. Applications include the automatic recognition of film genres, the detection and recognition of TV commercials, the determination of the period in which a feature film was produced, and video abstracting.

change of camera angle usually has no influence on the main background colors.

- In different scenes the audio usually differs significantly. Therefore, a video cut not accompanied by an audio cut does not establish a scene boundary.
- A third heuristic groups consecutive shots into a scene if the shots can be identified as representing a dialog (see dialog detection).

Audio cuts. *Audio cuts* are defined as time instances delimiting time periods with similar sound. They are employed to explore the similarity of the audio track of different shots. If there is no significant change in the audio track close to a video shot boundary, i.e. if the sound is continuing 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 for each time window of the audio track, predicting its values for the next time window by exponential smoothing, and declaring an audio cut to be where the current frequency and intensity spectrum deviate considerably from the prediction.

Once the video has been segmented into its basic components, it is essential to identify semantically rich events, e.g. close-ups of main actors, gun fire, explosions, and text appearing in the video. They help us to select those sequences of frames for our clips that are important for the abstract.

Finding Faces of Actors and Identifying Dialogs

In many video genres the cast is an essential piece of information. This is particularly true for feature films. Our abstracting system must understand where the main actors appear in the video. We have implemented a face detection algorithm and a method for recognizing the face of the same actor again, even across shot boundaries.

Face detection. An excellent face detection algorithm was developed by Rowley, Baluja and Kanade [10]. Their system recognizes about 90% of all upright and frontal faces in images (e.g. 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 is able to detect new faces in arbitrary images very reliably.

We have implemented our own neural network and trained it with approximately 1000 faces, in very much the same way. To increase the range of detectable faces, our implementation also searches for slightly tilted faces (± 30 degrees). This modification was necessary because the faces of actors in video are hardly ever upright, in contrast to faces in still images. To speed up processing, we only pass frame regions to the face detector in which the pixel colors are close to human skin color. This filter reduces the number of candidate face regions by more than 80%. Moreover, face detection is only run on every third frame of the video sequence. The result of face detection is a set of regions in frames where faces appear.

Face-based classes. 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 only interested in the main actors and therefore consider only faces larger than 30% of the frame size (i.e. faces in close-up shots). In a first step, faces within shots are related to each other according to the similarity of their position and size in neighboring frames, assuming that these features change only slightly from frame to frame. This is especially true for dialog scenes. In addition, we dispose of

accidental mis-classifications of the face detector by discarding all face-based classes with fewer than three occurrences of a face, and by allowing up to two drop-outs in the face-tracking process. In a second step, face-based classes with similar faces are merged by face recognition algorithms [4] in order to obtain face-based classes that are as large as possible.

Main actors. 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. It describes where, when and in what size that actor appears in the video.

Dialog detection. 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 cross-over relations between different actors. For example a male and a female actor could appear in an m-f-m-f sequence. An example of a dialog automatically detected in this way is shown in Figure 3.

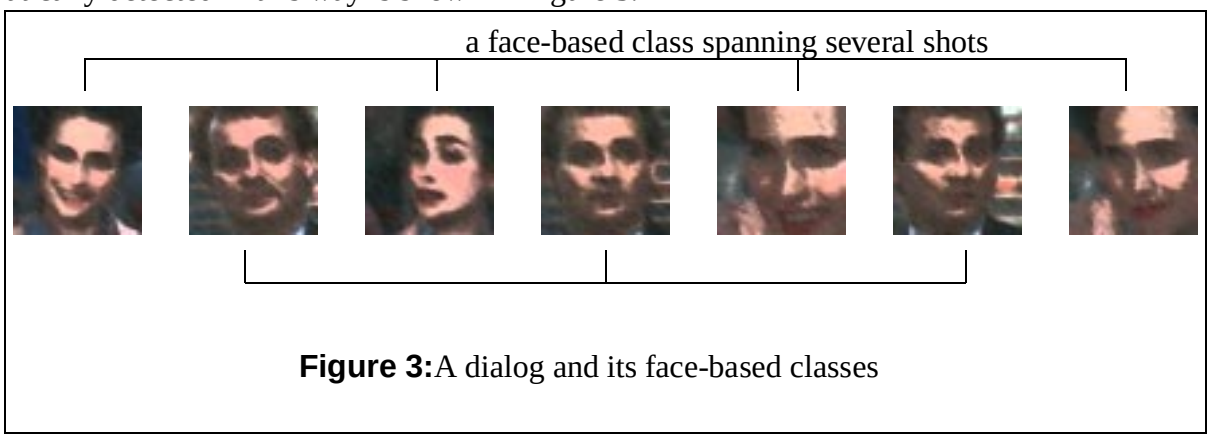

Figure 3: A dialog and its face-based classes

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 pieces 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]. They extract the bitmaps of text appearing in title sequences and translate their content to ASCII.

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 be able 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, those 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, and
- all characters of a cluster are contained in the segmented text regions for at least one frame.

Our experience shows 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.

A very 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 or the longest text line. This allows us to automatically extract the title in many practical cases.

Identifying Gun Fire and Explosions

Attractiveness to the user is an important design criterion for an abstract of a feature film. Action films often contain *explosions* and *gun fire;* we can recognize these events automatically. The 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 the distribution of the indicators with a database of known distributions for explosions and gun fires. If the distribution is found in the database a gun fire or explosion is recognized [7].

Generating the Video Abstract

We now concentrate on the generation of trailers for movies as a very important type of abstract. A movie trailer is a short appetizer for a movie, made to attract the attention of the viewer. Such a trailer requires the inclusion of eye-catching clips into the abstract. Again we use heuristics over the basic physical parameters of the digital video to select the clips for our trailer:

(1) *Important objects and people*: The most important objects and actors appearing in the original video should also appear in the trailer. Starring actors are especially important since potential viewers often have preferences for specific actors.

(2) *Action*: If the film contains explosions, gun fire, car chases or violence, some of these should be in the trailer. They attract attention and make viewers curious.

(3) *Dialogs*: Short extracts from dialog scenes with a starring actor stimulate the watchers' fantasy and often carry important messages.

(4) *Title text and title music:* The title text and parts of the title music should be contained in the trailer. Optionally, the names of the main actors from the opening sequence can be shown.

A special feature of our trailer generation technique is that the end of the movie is not revealed; we simply do not include clips from the last 20% of the movie. This guarantees that we don't take away 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 above heuristics. This is done in an iterative way. Initially, all scenes of the first 80% of the movie are in the scene candidate set. All decisions have to be based on physical parameters of the video because only those can be derived automatically. Thus the challenge is to determine relevant scenes, and a good clip as a subset of frames of each relevant scene, based on computable parameters.

We use two different mechanisms to select relevant scenes and clips. The first mechanism extracts *special events and texts* from the video, such as gun fire, explosions, cries, close-up shots, dialogs of main actors, and title text. We claim that these events and texts summarize the video well, and

are suited to attract the viewer's attention (see properties (1)-(4) above). The identification of the relevant sequences of frames is based on the algorithms described above, and is fully automatic.

The percentage of special events to be contained in the abstract can be specified as a parameter by the user. In our experiments it was set to 50%. If the total length of special event clips selected by the first mechanism is longer than desired, scenes and clips are chosen uniformly and randomly from the different types of events. The title text, however, will always be 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. Clips are then selected repeatedly from those 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.

In general clips must be much shorter than scenes. So how is a clip extracted from a scene? We have tried out two heuristics. With the first one, we pick those shots with the highest amount of action, and with the same basic color composition as the average of the movie. More details can be found in [8]. Action is defined through motion, either object motion or camera motion, and the amount of motion in a sequence of frames can easily be computed based on motion vectors or on the edge change ratio. The action criterion is motivated by the fact that actions 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 the perception of a video's mood and color composition should thus be preserved in the trailer.

The second heuristic takes a completely different approach. It uses the results of our MoCA genre recognition project. The basic idea of that project is to compute a large number of audio-visual 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 beforehand and stored in a database. The results of this project can now be used to select clips for the trailer in a more sophisticated way: Those 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 will automatically tailor the selection process to a specific genre provided that 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 et. al. showed that the sequencing of clips strongly influences the viewer's perception of their meaning [9]. Therefore ordering of the clips must be done very carefully. We first group the video clips into four classes. The first class or *event class* contains the special events, currently gun fires and explosions. The second class consists of *dialogs*, while the *filler clips* constitute the third class. The extracted *text* (in the form of bitmaps and ASCII text) falls into the

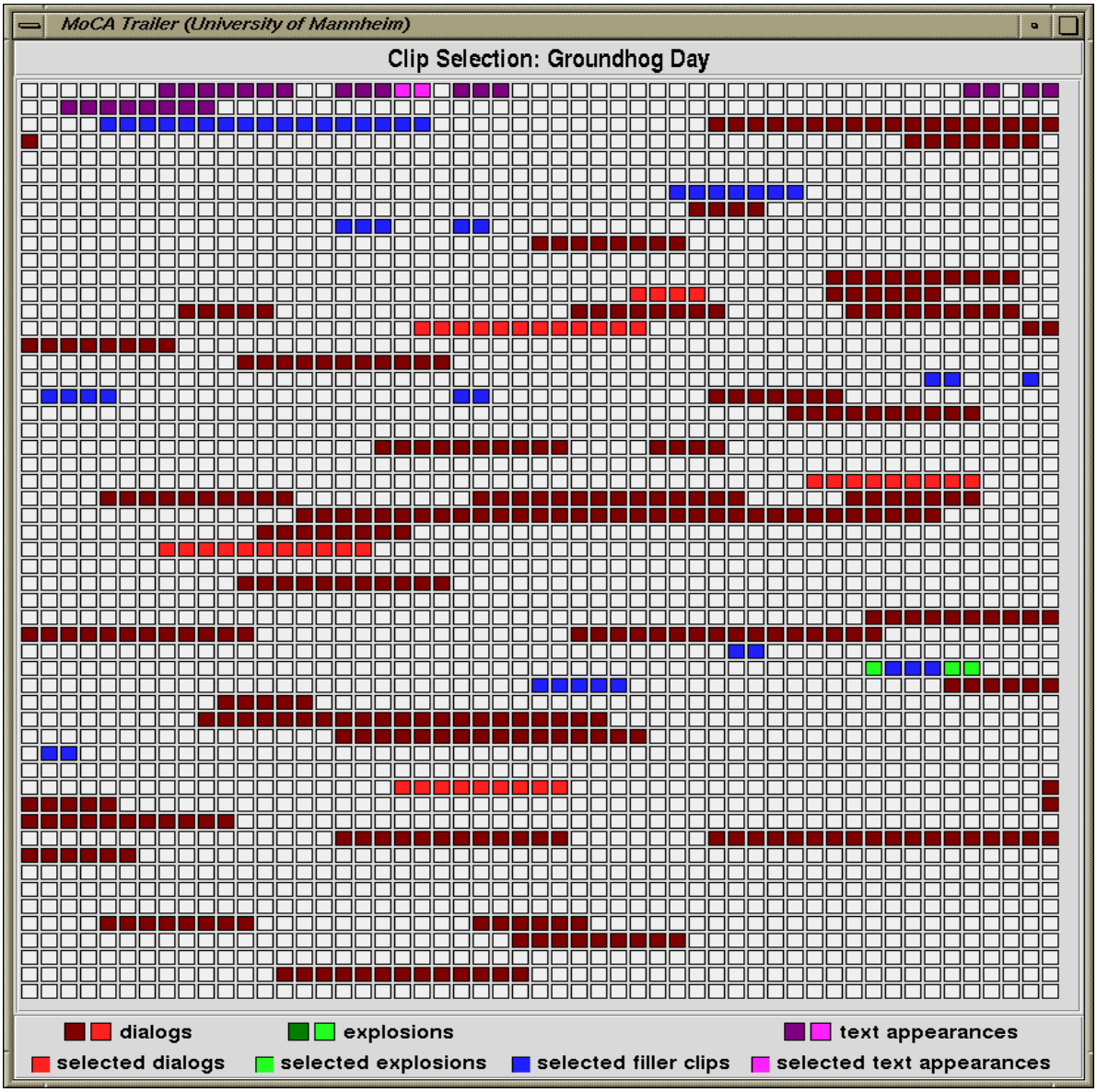

Figure 4: This graph shows the temporal distribution of the detected video and audio events of the movie “Groundhog Day” as well as those which have been chosen during the clips selection process to be part of the trailer. Note, since “Groundhog Day” is not an action movie, there are only two explosions and no gun fire. Each box represents two seconds (2828 in total). Time passes from left to right and top to bottom.

fourth class. Within each class the original temporal order is preserved.

Dialogs and event clips are assembled in turn into so-called *edited groups*. The maximum length of an edited group is a quarter 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 class four usually show the film title and the names of the main actors. The title bitmap is always added to the trailer, cut to a length of one second. Optionally, the actors' names can be added to the trailer.

Edits. We apply three different types of video edits in the abstract: hard cuts, dissolves, and wipes. Their usage is based on general rules derived from knowledge elicited from professional cutters [6]. This is 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 insert soft cuts (dissolves and wipes) between calmer clips only, such as dialogs. Table 1 shows the possible usage of edits in the different cases. A much more sophisticated approach for automatic video editing of humorous themes can be found in [6].

	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

Table 1: Edits in an abstract

Interestingly audio editing is much more difficult. A first attempt to simply concatenate the sound tracks belonging to the selected clips produced terrible audio. In dialog scenes it is especially important that audio cuts have priority over video cuts. The construction of the audio track of the abstract is currently performed as follows:

- The audio of special event clips is used as it is in the original.
- The audio of dialogs respects audio cuts in the original. The audio of every dialog is cut in length as much as to fill the gaps between the audio of the special events. 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 are planning to experiment with speaker recognition and with 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.

4. Experimental Results

In order to evaluate the MoCA video abstracting system, we ran a series of experiments with video sequences recorded from German television. We quickly found out 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 observe that the shots extracted by a human for an abstract depend to a large extent on the purpose of the abstract: For example, a trailer for a movie often emphasizes thrill and action without giving away the end, 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 the story. We conclude that automatic abstracting should be controlled by a parameter describing the purpose of the abstract.

When we compared the abstracts generated by our system with abstracts actually shown on television, we observed no obvious difference in quality (at least within the picture track - the audio track usually contains material which was originally not part of the video). In the case of the reviews for last week's episode of a TV series the scenes generated by our tool were very similar to the ones 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 persons. Even if the generated abstracts were quite different from the man-made ones, they could not tell which were better (see [8]).

For browsing and searching large information archives, many users are now familiar with the popular WWW interfaces. Therefore our abstracting tool can compile its results into an HTML page, including the anchors for playback of the short video clips. An example is given in Figure 5. The top of the page 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. The special events are followed by a 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 some statistics.

Figure 5: Result of video abstracting, compiled into an HTML page

5. Related Work

Video abstracting is a very young research field. We would like to mention two other systems suitable to create abstracts of long videos. The first is called video skimming [2]. It mainly aims at abstracting documentaries and newscasts. Video skimming 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 (such as Term Frequency Inverse Document Frequency) 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, the video clips are selected from the surrounding frames.

The second approach is based on the image track only. It does not generate 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 key frames. It allows to interactively descend into the story by selecting a story unit of the graph [11].

6. Conclusions and Outlook

We have presented a set of algorithms for the automatic generation of video abstracts. In a first step we decompose the input video into semantic units, called shot clusters or scenes. In a second step we detect and extract semantically rich pieces, in particular text from the title sequence and special events such as dialogs, gun fires, and explosions. Video clips, audio pieces, images and text are extracted and composed to an abstract. The output can be compiled into an HTML page for easy access through browsers.

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

Acknowledgments

Much of our work on movie content analysis was done jointly with Stephan Fischer whose contributions to the MoCA project we gratefully acknowledge. We would also like to thank Ramesh Jain of UC San Diego for his assistance in the preparation of this paper.

References

[1] D. Bordwell, K. Thompson: Film Art: An Introduction. 4th ed., *McGraw-Hill*, 1993.

[2] M. Christel, T. Kanade, M. Mauldin, R. Reddy, M. Sirbu, S. Stevens, and H. Wactlar. Informedia Digital Video Library. *Communications of the ACM*, 38(4):57-58 (1995).

[3] A. Dailianas, R. B. Allen, P. England: Comparison of Automatic Video Segmentation Algorithms. Proc. SPIE 2615, Photonics East 1995: *Integration Issues in Large Commercial Media Delivery Systems*, Andrew G. Tescher; V. Michael Bove, Eds., pp. 2-16

[4] S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back: Face Recognition: A Convolutional Neural Network Approach. *IEEE Trans. Neural Networks, Special Issue on Neural Network and Pattern Recognition*, 1997, to appear

[5] R. Lienhart. Automatic Text Recognition for Video Indexing. *Proc. ACM Multimedia 1996*, Boston, MA, pp. 11-20

[6] F. Nack, A. Parkes: The Application of Video Semantics and Theme Representation in Automated Video Editing. *Multimedia Tools and Applications*, Vol. 4, No. 1 (1997), pp. 57-83

[7] S. Pfeiffer, S. Fischer, W. Effelsberg: Automatic Audio Content Analysis. *Proc. ACM Multimedia 1996*, Boston, MA, pp. 21-30

[8] S. Pfeiffer, R. Lienhart, S. Fischer, W. Effelsberg: Abstracting Digital Movies Automatically. In *J. Visual Communication and Image Representation*, Vol. 7, No. 4 (1996), pp. 345-353

[9] C. Pryluck, C. Teddlie, R. Sands: Meaning in Film/Video: Order, Time and Ambiguity. *J. Broadcasting* **26** (1982), pp. 685-695

[10] H. A. Rowley, S. Baluja, T. Kanade: Human Face Recognition in Visual Scenes. *Technical Report, Carnegie Mellon Uuniversity,* CMU-CS-95-158R, School of Computer Science, November 1995

[11] M. Yeung, B.-L. Yeo, B. Liu: Extracting Story Units form Long Programs for Video Browsing and Navigation. *Proc. IEEE Multimedia Computing & Systems 1996,* Hiroshima, Japan, pp. 296-305

[12] R. Zabih, J. Miller, K. Mai; A Feature-Based Algorithm for Detecting and Classifying Scene Breaks. *Proc. ACM Multimedia 1995,* San Francisco, CA, pp. 189-200