2

A structured video browsing tool

G. Benedetti, B. Bodin, F. Lhuisset, O. Martineau, B. Merialdo
Multimedia Communications Dept., Institut Eurecom,
BP 193, 06904 Sophia-Antipolis, France
Tel: +33 93 00 26 29, Fax: +33 93 00 26 27
email: {benedett, bodin, lhuisset, martinea, merialdo}@eurecom.fr

Abstract

This paper describes a tool for browsing video sequences. It is based on an automatic decomposition of the sequence into a number of consecutive shots. For each shot, a representative image is constructed and assigned a hypermedia link to the original video. The images are presented to the user as a document that can be browsed. Selecting an image leads to the playback of the corresponding portion of the video.

While a number of browsers have already been proposed, the novel aspect of our tool is that it is able to perform a hierarchical classification of these shots and to present a structured view of the video sequence. The user can thus get a progressive view to the parts of the video that interest him by moving inside the hierarchical structure. This hierarchical representation is first computed automatically, but it can also be hand-edited so that the user may provide his own structure to the document. The implementation that we propose uses analog video and assumes minimal computing and storage requirements, so that it can be effectively used as a practical system.

Keywords

Video browsing, video cut detection, video editing.

L. J. Bass et al. (eds.), *Engineering for Human-Computer Interaction*

1 INTRODUCTION

The development of Multimedia technologies is leading to a proliferation of documents including video sequences. While video is a very effective way of communication, video objects suffer the drawback that they currently support a limited number of operations: basic recording, playback, fast-forward and rewind, plus compression and some hand-editing. Therefore, it is important to develop systems that will allow to perform easily complex operations on multimedia documents. In the present work, our purpose is to develop a tool that will allow a user to easily get an idea of the contents of a video sequence, without the need of an exhaustive playback of the complete video.

Suppose that you are given a video sequence of unknown content (it can be either an analog video-tape or a digital video file). How can you get an overview of the content without actually playback the whole sequence? If you already know the content, how can you quickly find a particular position inside the sequence? The tool that we have developed is a video browser whose purpose is to help in answering these two questions. From the original video sequence, it automatically constructs a hypermedia index composed of representative images. Browsing at the images quickly gives an overview of the video sequence. When an image is selected, it automatically provides a position in the video sequence for immediate playback.

Digital video generally requires extensive processing and storage resources. In the implementation of our video browser, we have been careful at minimizing the system requirements of our computations, so that this tool can effectively be used even on regular workstations without the need of an exotic environment. For example, the processing of an entire video tape (two or three hours) may be performed with only a few Mbytes of disk capacity. The hypermedia index that is constructed can be constrained to fit on a regular diskette, which can then be easily distributed together with the video-tape.

The rest of the paper is organized as follows: Section 2 provides some generalities about video browsers and existing work.Section 3 describes the setup that we use for analyzing video sequences (image acquisition and segmentation). Section 4 introduces the operation of our tool for sequential presentation of video. Section 5 discusses the hierarchical presentation that our tool supports. Finally, Section 6 gives some hints about possible improvements, applications and future work.

2 VIDEO BROWSERS

The basic idea of video browsing has already been proposed by a number of people (Tonomura et al., 1994) (Smoliar and Zhang, 1994) (Hampapur et al., 1994) (Arman et al., 1994a) (the list is not exhaustive), with an emphasis on various aspects of the problem. The process of video browsing can generally be decomposed in three steps:

- segmentation of the video sequence into consecutive units,
- construction of a pictorial representation for each unit,
- presentation of the units to the user (eventually in a structured way), and navigation.

Each of these steps generates its own set of requirements, algorithms and research problems. Segmentation was the object of extensive works, based on the study of various parameters and algorithms to detect changes in the video (Nagasaka and Tanaka, 1991)

(Smoliar and Zhang, 1994), on the possibility to model various visual effects (fade, dissolve...) (Hampapur et al., 1994) (Davenport et al., 1991), or on the computational aspects of the algorithms (for example (Arman et al., 1994b) detect transition in an MPEG stream from DCT coefficient rather than image pixels).

The construction of a pictorial representation for a segment of video can take various aspects. It can be dynamic, for example a continuously playing window (Tonomura et al., 1990) (Brondmo and Davenport, 1990), an image extracted from the segment with various decorations indicating motions (Arman et al., 1994a) (Smoliar and Zhang, 1994), or a completely artificial image composed from the whole segment (Teodioso and Bender, 1993).

Various presentations and user interfaces have also been proposed, ranging from linear presentation of images (Arman et al., 1994a), photobooks (Tonomura et al., 1994) to structured presentations (Smoliar and Zhang, 1994). When comparing our work with other systems, we decided to focus on usability. We chose to use robust algorithms for processing, with a careful implementation so that their usage be possible even on a non-experimental setup. This lead to the choice of using analog video as input, as this is the only way to effectively store large amounts of video at reasonable quality and cost. We took care that the digital processing and storage involved in the usage of the browser could be kept to a reasonable amount, compatible with regular computer resources. We also allowed for customization of the browser views, so that the user may, if he wishes, provide his own structure to the indices based on his own understanding of the video.

3 VIDEO PROCESSING

3.1 Acquisition

Digital video is becoming popular and has a number of attractive features. However its usage still impose severe constraints on current computer systems, because of the high bandwidth, computing power and disk storage that are needed for reasonable picture quality and image size. In the current implementation of our video browser, we decided to use analog video as input source, so that our system can easily be applied to realistic sequences at a reasonable cost. While the source material remains analog, the video browser constructs a digital structure (including digital images) that allows easy access to this analog video. Of course, a digital implementation of the browser, provided that the disk, computer and bandwidth resources are available, would be straightforward.

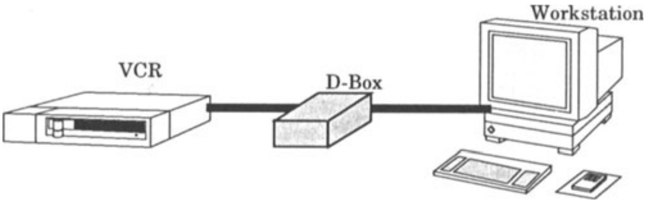


Figure 1 Overall configuration connecting workstation and VCR.

In our setup, an analog VCR is controlled from a workstation using a Control-L interface. The VCR output is connected to a SunVideo card which is in charge of video acquisition (see Figure 1). The digital video is stored on the regular Unix file system. While the VCR is playing, the video is digitized through the SunVideo at low resolution and frame rate. Our typical setup was to use a 128x96 pixel image, and a frame rate of between 10 and 15 frames per second. While these parameters are insufficient for good quality video coding, they are sufficient for reasonable sequence detection and index display. With these parameters, the regular resources of the workstation can be used without the need for a dedicated workstation or disk storage system. The images are compressed by the SunVideo card using JPEG compression method. For example, with these parameters, one minute of video sequence takes about 2 Mbytes of disk space. It is therefore conceivable to store an entire video in this format on a regular file system.

To further save space, since only a small fraction of the original video will be used in the final index, we decided to apply a progressive processing of the tape. Acquisition of a complete video sequence is performed by processing two minutes of video at a time. These two minutes are entirely digitized and indexed. Only the images that are selected for the index are kept, which amounts to an average of 20 images out of 720. The processing of a two minute period uses about 5 Mbytes of disk space, which are almost entirely freed when the index is constructed. When a two minute period is processed, the video browser starts processing the following two minute period, and this goes on until the end of the tape.

Of course, special care is taken a the boundary of these segments (a small overlap is actually processed twice) so that segmentation is not perturbed by the two minutes boundaries, and to allow to compensate for small inaccuracies in the positioning of the VCR.

3.2 Segmentation

Once a portion of video has been digitized, it is segmented into coherent shots using a histogram-based approach. For each image, colors are projected on a gray scale and an histogram is computed. It is compared with the histogram of subsequent frames.When the difference of two consecutive histograms is larger than a predefined threshold, a shot boundary is detected (see Figure 2).

This method is relatively efficient and can be easily implemented at low cost. It is quite robust when there is a sharp transition between sequences. It is however limited in case of more complex cases such as fading, zooming or travelling. In those cases, a comparison of histograms located at a larger distance may indicate that the focus has substantially changed, but generally more complex algorithms are needed to detect the kind of transition that is taking place. In those cases, there is no precise limit between two shots, rather a continuous transition (where start and ending time can be determined).

3.3 Selection

For each video segment, a representative image is selected. We chose a simple mechanism by selecting one of the images of the sequence approximately located in the middle of the sequence. With each selected image, we keep the informations related to the video sequence: time code for beginning and ending times, sequence number etc... Some special processing is also performed to remove empty sequences corresponding to portions of the video tape which have not been recorded.

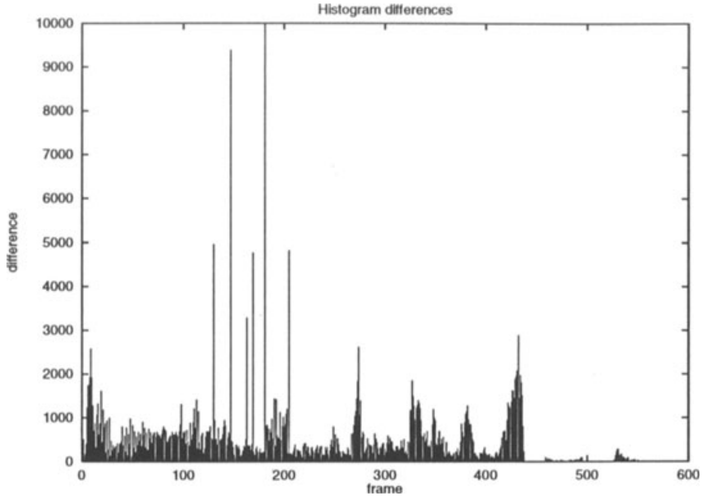


Figure 2 Histogram differences in a video sequence.

4 SEQUENTIAL BROWSING

The simplest form of video browsing is to display sequentially the sequence of images that have been kept in the index (see Figure 3). This gives a detailed static view of the video sequence. Using the scrollbar, it is possible to move forward and backward along the sequence, or to go directly at any given time location. Each image is decorated with a number of indicators: segment number, segment length, beginning timecode. Selecting an image positions the VCR automatically to the initial position of the sequence, as given by the timecode, and starts playing back. A control interface allows then to pilot the VCR directly

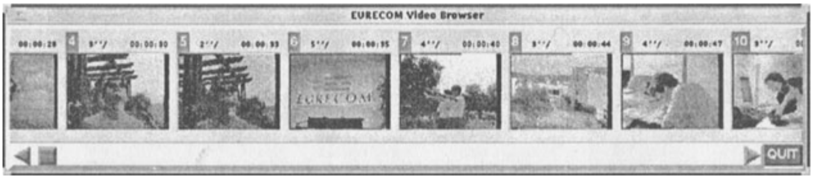


Figure 3 Sequential browsing.

It should be noted that, once the index is constructed, the video browser does require very limited resources to operate. We don't need a video card any longer, we simply need a very small amount of disk space to store the index, and the connection of the workstation to a VCR. This means that the distribution of a video tape and its index can be done to almost any computer environment.

5 HIERARCHICAL BROWSING

While sequential browsing is important because it provides a detailed view of the video sequence, it has the drawback that it is generally difficult to get a global view on the content of the tape. To improve this, we defined the notion of hierarchical view of the video sequence: a tree structure is constructed over the video shots. The leaves of the tree are the images, and the nodes represent various portions of the video sequences. An image is constructed for each node to represent the portion of video that this node covers. Each node also contains an hypermedia link to the beginning of this portion of video.

A hierarchical browser allows to interactively explore this hierarchical structure. It starts by displaying the images corresponding to the nodes at level 1 (see Figure 4). This gives an overview of the content of the entire sequence, with an indication of the content that can be found in each sub-part. By clicking on one image, the browser moves to the corresponding node and displays the corresponding daughter images.

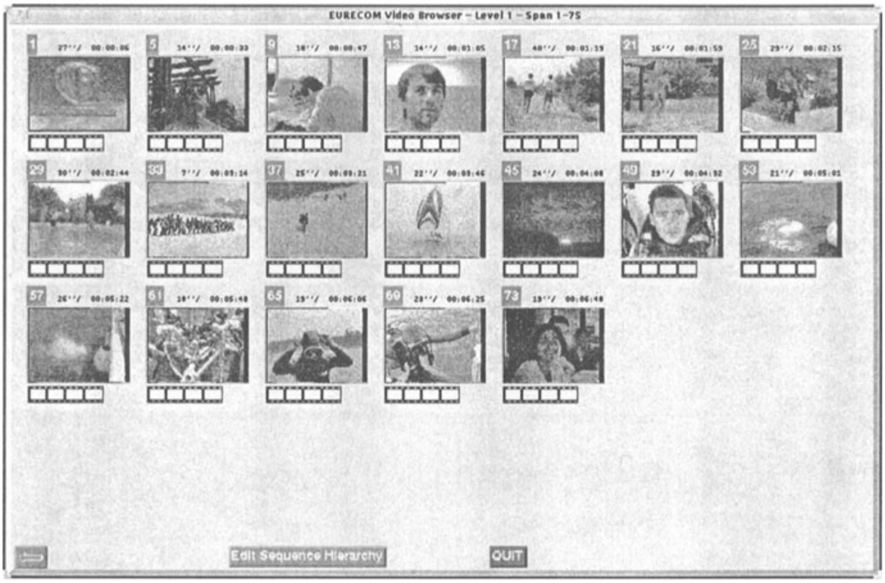


Figure 4 Hierarchical browsing.

In the display, the images are decorated with the same kind of informations as described previously in the sequential browser. A caption field is also available, and its usage will be described later. Note that there are now two operations that can be accomplished when the image for a node is selected: it can be either to display the corresponding portion of the video tape, or to go down one level in the hierarchical structure. For compatibility with the sequential mode, the playback of the videotape is always activated by clicking directly on the image, while the button located below the image allows to move inside the structure.

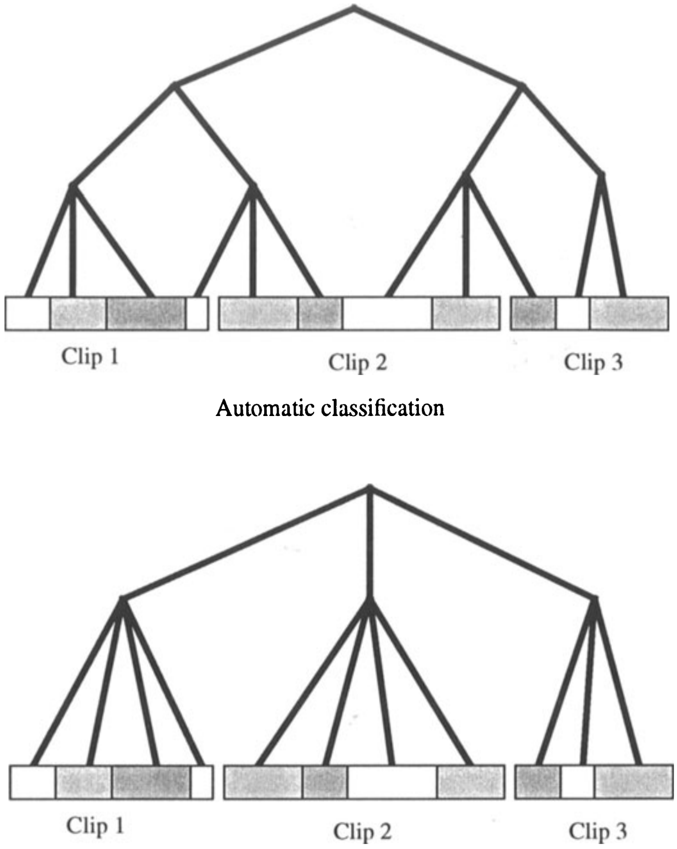
The hierarchical classification is actually performed using only time information, so that the portions covered by nodes at the same level are approximately of the same length (in time). The number of classes that are constructed for each level is chosen so that all the images for any node can be displayed in a single window. With this mechanism, the images that are displayed at the root of the hierarchy provide an overview of the complete video in one glance. Another question is how to select a representative image for a daughter node. For a leaf of the structure, which corresponds to a single shot, it is easy because only one image was kept in the index for this shot. For interior nodes, which cover several shots, the image corresponding to the longest shot covered by the node is kept as a representative for the node.

A major feature of our system is that the hierarchical structure can be hand-edited by the user. The informations related to the hierarchy are included in a human-readable file which contains one line per node of the tree. This line contains:

- the absolute segment number,
- the position of the node in the structure, written as a dotted number, so that 1.2.1 is the first daughter of the second daughter of the first daughter of the root,
- the length of the portion of video covered by the node,
- the timecode of the beginning of this sequence,
- the name of the file containing the representative image for the node,
- some text which is displayed as a caption by the browser. This text is initially blank, and can be modified by the user to provide comments or indications about the video.

The hierarchical structure as it is described is actually quite flexible. The first intended usage is that the user would precisely determine the portions of video that he wants to be able to display and modify the structure accordingly. For example, we have a video tape containing a series of clips (a few minutes each) about various projects and demonstrations that we have been doing in our laboratory. Editing the hierarchy allows us to construct one node at level 1 for each video-clip, to select the proper timecode to start the clip, and to write a caption that provides the name of the project or demonstration (see Figure 5). Depending on the amount of data, we are even able to create a multi-level hierarchy if, for example, we first want to classify our video-clips into broad categories.

Finally, it should be noted that, while the initial hierarchy that is automatically constructed by the browser is a real tree-structure which respects the time ordering of the video shots corresponding to the leaves, there is nothing in the operation of the browser that prevents this tree-structure to be changed into a trellis-structure if needed by the user. This has several advantages. Certain shots can be accessed from several paths in the structure, for example, if a project would fit into two or more categories, we would point to the same video-clip from two different nodes. Also, it may allow to construct nodes that cover non-sequential video shots, for example to prepare a presentation which will include a given sequence of demonstrations in any desired order, even if the video clips are not recorded sequentially on the tape.



Automatic classification

Hand-edited classification

Figure 5 Editing a hierarchical structure over a video sequence.

6 ADVANCED FUNCTIONS

The functions that are provided by the hierarchical structure are quite important from a practical point of view. However, the current way of hand-editing the file describing the structure is not the most convenient. We are planning to move to a graphical interface to allow structure editing, which will allow modifications by moving images from one to another, creating, moving and deleting nodes.

Another facility will be to use this browsing interface as a tool for video editing, where cut-and-paste operations on images will be automatically translated to play-back and recording functions for two or more VCRs. This will provide an easy way to construct a well-balanced video from a series of raw recordings.

Other improvements can be envisioned by using more advanced image processing algorithms. For example, the work described in (Arman et al., 1994a) allows to automatically find images that a similar to the image currently selected. This can be used as a "direct goto" based on image content. Another possibility is to classify all the images appearing in the index, removing those that are similar and displaying only the remaining ones. This would provide a list of all different images in the video, and could be useful if, for example, we are looking for the appearance of a certain character in the video.

Finally, we envision that such a tool can be completed with advanced multimedia indexing tools such as word-spotting (on the audio channel), face recognition, movement identification etc... so that the index structure can be augmented with more and more relevant informations.

7 CONCLUSION

We have presented a video browsing tool that automatically constructs an hypermedia index to a video sequence. The index is composed of images extracted from shots of the original sequence which are linked to the corresponding portion of the video. Our tools allows two modes for browsing:

- sequential browsing, where images are presented in the same order as the video, allows a detailed presentation of the content of the video,
- hierarchical browsing, where a tree-structure is constructed so that the presentation of the root gives an immediate overview of the content of the entire video. This structure can be hand-edited to meet specific desire of the user, and may even be extended to a trellis-structure.

Particular care has been taken to ensure that a minimum of computer and storage resources are required in each step of the processing. In particular, once the index is constructed, it can be stored on a diskette, distributed with the videotape and used on almost any regular workstation.

With the development of multimedia technology, we believe that video browsers will become common tools to manipulate video sequences. Yet much research remains to be done to construct powerful algorithms that will allow a more efficient indexing.

8 BIBLIOGRAPHY

Arman, F., Depommier, R., Hsu, A., and Chiu, M.-Y. (1994a). Content-based browsing of video sequences. *ACM Multimedia Conference*, pages 97–103.

Arman, F., Hsu, A., and Chiu, M.-Y. (1994b). Image processing on encoded video sequences. *Multimedia Systems Journal*, 1(5):211–219.

Brondmo, H. and Davenport, G. (1990). *Creating and Viewing the Elastic Charles - a*

Hypermedia Journal. Intellect Ltd, Oxford, England.

Davenport, G., Smith, T. A., and Pincever, N. (1991). Cinematic primitives for multimedia. *IEEE Computer graphics and Applications*, pages 67–74.

Hampapur, A., Jain, R., and Weymouth, T. (1994). Digital video segmentation. *ACM Multimedia Conference*, pages 357–364.

Nagasaka, A. and Tanaka, Y. (1991). Automatic video indexing and full-video search for object appearances. *2nd IFIP Working Conference on Visual Database Systems*, pages 119–133.

Smoliar, S. W. and Zhang, H. (1994). Content-based video indexing and retrieval. *IEEE Multimedia*, pages 62–72.

Teodioso, L. and Bender, W. (1993). Salient video stills: Content and context preserved. *ACM Multimedia 93*, pages 39–46.

Tonomura, Y., Akutsu, A., Otsuji, K., and Sadataka, T. (1990). Content oriented visual interface using video icons for visual database systems. *Journal of Visual Language and Computing*, 1:183–198.

Tonomura, Y., Akutsu, A., Taniguchi, Y., and Suzuki, G. (1994). Structured video computing. *IEEE Multimedia*, pages 34–43.

9 BIOGRAPHIES

G. Benedetti, B. Bodin, F. Lhuisset, O. Martineau, are students from the Ecole Nationale Supérieure des Telecommunications (ENST Paris), and follow a specialization in the Multimedia Communications Dept at the Institut EURECOM.

B. Merialdo is an Associate Professor in the Multimedia Communications Dept at the Institut EURECOM.

Discussion

Hong-Mei Garcia: What is the research contribution here, given that you can only search one video at a time.

Bernard Merialdo: The contribution is to know what is in a single videotape and to give easy access to it. The indices of all videotapes can be kept on-line since they are not large. Our future work is focused on more generic indexing strategies.

Helmut Stiegler: Do you have any position whether the proper device for looking at videos in the future will be the TV with a set-top box (as proposed by Oracle) or a PC (as proposed by Microsoft)?

Bernard Merialdo: No opinion.

Jim Larson: What has been the experience of the users of this system? Have you done any user testing?

Bernard Merialdo: No. We are the only users thus far.

Christopher Ahlberg: A comment: lots of user groups will be interested in this. However, creating libraries will requires different techniques.

Joëlle Coutaz: This is an excellent domain for computer vision. As you point out, there have been thousands of papers on video segmentation. The conclusion has been that the way you segment a video depends on the application. It requires interpretation and interaction with the user.

Wendy Mackay: This is designed for highly structured video. Perhaps ultimately it would be better to capture the video information at recording and editing time. Also, refer to EVA (Experimental Video Annotator) from MIT & other work at MIT on the use of video streams to access video.

Len Bass: News programs such as CNN use very distinctive, stylized presentations. Have you looked at a style based manner of segmentation?

Bernard Merialdo: I am aware of a grammar-based parsing scheme for dealing with this, but we have not yet taken advantage of it.

Matthew Fuchs: Soon it will be cheap enough to store the video digitally.

Bernard Merialdo: Yes, I believe it will be feasible in 2-3 years.

Matthew Fuchs: Also, have you considered storing the index directly on the tape?

Bernard Merialdo: No.