Video Browsing Using Interactive Navigation Summaries

Klaus Schoeffmann and Laszlo Boeszoermenyi
Institute of Information Technology, Klagenfurt University, 9020 Klagenfurt, Austria
{ks,laszlo}@itec.uni-klu.ac.at

Abstract

A new approach for interactive video browsing is described. The novelty of the proposed approach is the flexible concept of interactive navigation summaries. Similar to time sliders, commonly used with standard soft video players, navigation summaries allow random access to a video. In addition, they also provide abstract visualizations of the content at a user-defined level of detail and, thus, quickly communicate content characteristics to the user. Navigation summaries can provide visual information about both low-level features but even high-level features. The concept fully integrates the user, who knows best which navigation summary at which level of detail could be most beneficial for his/her current video browsing task, and provide him/her a flexible set of navigation means. A first user study has shown that our approach can significantly outperform standard soft video players - the state-of-the art ”poor man's” video browsing tool.

1. Introduction

The application domain of digital video has experienced a considerable broadening during the last years. In addition to entertainment, videos are used for several other purposes these days. E-Learning, surveillance, health-care, product presentation and product usage instructions are only a few examples of application areas where masses of digital videos are produced every day. However, while recording digital videos and saving them in a video archive are quite simple tasks, browsing videos in order to find out (1) whether a video is of-interest or out-of-interest and (2) what content a video roughly contains, remains a difficult and time-consuming task. Although several approaches for video browsing have been proposed, we believe that video browsing still can be significantly improved. In particular, in order to become appropriate to the majority of users, we argue that a meaningful video browsing tool should

- be easy to use and, thus, also appropriate for users without an expertise in video-techniques. Several approaches, proposed so far, require specific knowledge often hard to understand for non-expert users.
- be suitable for several application domains (i.e. several types of video). Many video browsing approaches have been presented, which work for one specific domain only (e.g. news or sports etc.) Moreover, most of the proposed approaches are based on shots. However, in some domains (e.g. medicine, certain kinds of surveillance, e-learning) a video sequence often consists only of one single shot.
- require both low content analysis time and low additional storage space for meta-data. In the literature, some approaches have been presented, which require very high analysis time (e.g. several hours or even days) before a video sequence can be browsed. Long analysis phases avoid on-demand usage scenarios, required in many application domains.

In current praxis many users, although typically not professionally working in the area of video browsing, often use simple soft video players for browsing digital videos, as video players fulfill all three above-mentioned requirements. They provide very simple navigation means as play, pause, fast forward and rewind. Additionally, they also provide a time slider which allows jumping to a particular time position and, thus, random access. The main advantage of these navigation means is that users are very familiar with them since they have been used for several decades by VCR hardware, invented in the 1960s. However, it is obvious that video players are not very well suited for the purpose of video browsing. Fast forwarding a video only allows linear playback and it is quite stressful for humans when used with high speed. Also the time slider raises problems when used with long video sequences resulting in a coarse resolution and, thus, imprecise behavior, where short movements will cause jumps by minutes rather than seconds. Moreover, video players give no feedback to a user about already visited positions in time. For long video sequences this often leads to the situation that the user does not know where to go in order to find the (missing) searched scene. This problem has also been observed in our user study, which is described in Section 4.

We propose a concept for video browsing which strives for an improvement in several aspects. First, it is easy to use and quickly understandable to users since it is based on known interaction models (namely that of time sliders and scroll bars). Next, it enhances already proposed work ([2, 5, 7, 16] with meaningful extensions to make video browsing appropriate for several application domains (and several types of video content). In particular, we propose the concept of *interactive navigation summaries* to be used as alternatives to time sliders. The main idea is to use abstract visualization in order to give the user as much information about the content as possible and as required. Such visualization summaries can visualize meta-data ranging from simple *low-level features* - which are usually quickly extractable - to *high-level features*. Instead of favoring any type of extracted meta-data, and provide search functions based on it, we let the user determine which information suits best to his/her current browsing needs. A first user study has shown that users can understand our browsing concept very quickly and highly benefit from it when similar scenes within a video sequence should be found.

The paper is organized as follow. Section 2 summarizes related work in that area. Section 3 describes the concept of interactive navigation summaries. The result of our user study is given in Section 4 before Section 5 concludes our work.

2. Related Work

The idea of enhancing a video player's time slider with additional information has first been raised in 2001 by Barbieri et al. [2]. Similar to our concept they proposed to visualize low-level features by color bars in two time combined time sliders, together called *Color Browser*. A color bar represents information about the content (e.g. the dominant color of a frame or the corresponding audio volume). While the time-scale of the first time slider can differ from one video sequence to another, the second time slider (representing details of a zoom-window in the first time slider) should always show the same time-scale in order to allow users to "recognize patterns of colors within different programs". Moraveji [16] proposed to use visually distinct color bars in the background of the time slider whereas a color bar represents a specific time region and visualizes the most relevant content-related feature. Divakaran et al. [7] presented a content-based feature visualization, shown as timeline overlay, integrated into a personal video recorder which is based on classification of audio segments. They evaluated their concept with several sports videos in a user-study, which showed that users liked the importance level plot for its flexibility, even if the visualization resulted in mistakes. Chen et al. recently presented the *EmoPlayer* [5], a video player which can visualize affective annotations in the background of the time slider. In particular, different emotions of actors and actresses - angry, fear, sad, happy, and neutral - can be visualized for a selected character in a video based on a manually annotated XML file. Different colors are used for the different emotions. When a character is not present in a specific scene, the bar shows no color (i.e. white) for the corresponding segment. A user can, therefore, simply identify in which segments of the video sequences a particular character is present and which emotion the character expresses.

To overcome the limitations of time sliders, Hürst et al. proposed the *ZoomSlider* interface [12, 13]. The entire player window is used as a time slider with different stages of granularity in a linear way. The granularity (i.e. resolution) of the time slider is linearly dependent on the currently selected vertical position in relation to the whole height of the player window. The finest granularity is used at the top of the window and the coarsest granularity is used at the bottom of the window. This concept has been further extended in [11] where a similar mechanism for changing the playback speed of the video is proposed. The right vertical part of the player window is used to change the playback speed where the slowest speed is assigned to the top and the highest speed is assigned to the bottom of the player window. In a linear fashion the user can, therefore, select any playback speed based on the actual vertical position. An interesting approach for video browsing by direct manipulation has been presented by Kimber et al. [15] and Dragicevic et al. [8]. As a complement to the time slider they propose *relative flow dragging*, which is a technique to control moving imagery by direct mouse manipulation of content objects. Dragicevic et al. use feature-based optical flow estimization based on SIFT salient feature points of two consecutive frames. A user study has been conducted which has shown that relative flow dragging can significantly outperform the time slider on specific browsing tasks.

Several other approaches for video browsing and video skimming have been presented in the literature during the recent years. Campanella et al. [4] proposed a visualization of MPEG-7 low-level features in a 2D feature space where each axis corresponds to a specific feature, selected by the user. Each shot is represented by a small square showing the dominant color of the shot. The shots are also visualized in a temporal manner by (1) color bars showing the dominant color of the shot and (2) keyframes of a shot. Hauptmann et al. [10] proposed *Rapid Serial Visual Presentation (RSVP)* (of keyframes) of query results in order to exploit both maximal use of human perception skills and the system's ability to learn from human interaction. Goeau et al. [9] proposed the so called *Table of Video Contents (TOVC)* visualization for browsing story-based video content like news, interviews, or sports summaries. Rooij et al. [6] introduced the concept of *video threads*, where a video thread is a sequence of feature-based similar shots from several videos in some specific order. A visualization of video threads for the purpose of explorative browsing has been developed as a *CrossBrowser* and a *RotorBrowser*. Based on a *tempo function*, Adams et al. [1] proposed *temporal semantic compression* for video browsing. Their video browsing prototype allows shot based navigation, whereas only a few shots are shown at a glance containing the selected shot in the center. Several different compression modes can be chosen. While *linear compression* simply speeds up playback, *mid-shot constant* takes a constant amount from the middle of a shot at constant playback rate. *Pace proportional* uses variable playback rate based on the frame-level tempo and *interesting shots* discard shots with low tempo values according to the selected compression rate. Jansen et al. [14] proposed *VideoTrees*, as alternatives to storyboards, for the purpose of video browsing. A VideoTree is a hierarchical tree-like temporal presentation of a video through keyframes. With each depth level the level of detail increases as well (until shot granularity). The current selected node in the tree is always centered, showing the context (i.e. a few of the adjacent nodes) in the surrounding area. In a user study with 15 participants they showed that the VideoTrees can outperform storyboards regarding the search time (1.14 times faster). However, the study also revealed that users found the classical storyboard much more easy and clear.

3. Interactive Navigation Summaries

We define navigation summaries as an extension (or maybe replacement) to the usual time slider. A navigation summary behaves very similar to a usual time slider, i.e. when a user clicks on any horizontal position, the playback for the video starts from the corresponding time position. When the user moves the mouse over a navigation summary, the corresponding playback time and a preview picture of the underlying content is shown (either in an own area of the window or immediately at the mouse position). Thus, a navigation summary enables random access to the video with preview information. Most importantly, a navigation summary is used as a video abstraction means and, therefore, visualizes extracted meta-data in a temporal manner. By providing several navigation summaries and let the user choose which one suits best to his/her current browsing task, this concept becomes a very flexible way of browsing. Navigation summaries can visualize very simple low-level content characteristics (e.g. motion intensity, dominant colors, etc.), which can be extracted from a video in a quick analysis phase. However, navigation summaries can also visualize high-level information (e.g. positions of shot-boundaries or commercial breaks, sports events identified by audio analysis as proposed in [7], or actor/actress emotions like proposed in [5], etc.)

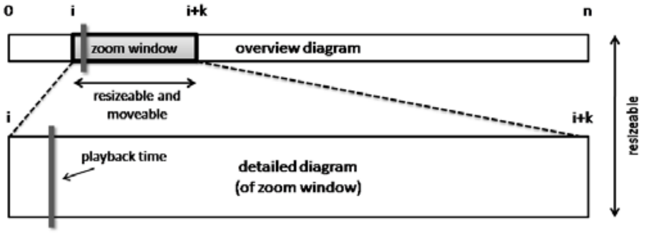

Figure 1. Interaction model of a navigation summary

Figure 1 shows the model of a navigation summary. As shown in the figure, a navigation summary consists of two horizontal diagrams (1) a small-sized *overview diagram* and (2) a *detailed diagram*. In general, both diagrams show the same abstract content visualization but use different levels of detail. In order to express the detailedness of a diagram we introduce the notion of *Frames-Per-Pixel* for both diagrams (denoted as φ_O and φ_D; the higher φ, the coarser, less detailed is the presentation), defined by the following simple equations, whereas W represents the available width of the diagram:

$$\varphi_O = \frac{n}{W} \tag{1}$$

$$\varphi_D = \frac{k}{W} \tag{2}$$

The overview diagram shows a small-sized content visualization for all n frames and, therefore, has usually a quite high value of φ_O. The detailed diagram shows content visualization for a particular number of k frames according to a user-defined *zoom window* with a maximum detailedness of $\varphi_D = 1$. Our current implementation uses simple linear frame subsampling to display lower levels of detail (when $\varphi_O < 1$ or $\varphi_D < 1$).

The user can change (1) the position of the zoom window, (2) the horizontal size of the zoom window, and (3) the vertical size of the whole navigation summary. The size of the zoom window is limited by the following rule which defines a maximum size of $\varphi_D = \varphi_o$ and a minimum size of $\varphi_D = 1$:

$$W \leq k \leq n \tag{3}$$

Whenever position or size changes, the content of the detailed diagram is updated immediately. The reason for using two diagrams is that it enables convenient navigation even for long video sequences, always preserving the context of the detailed view.

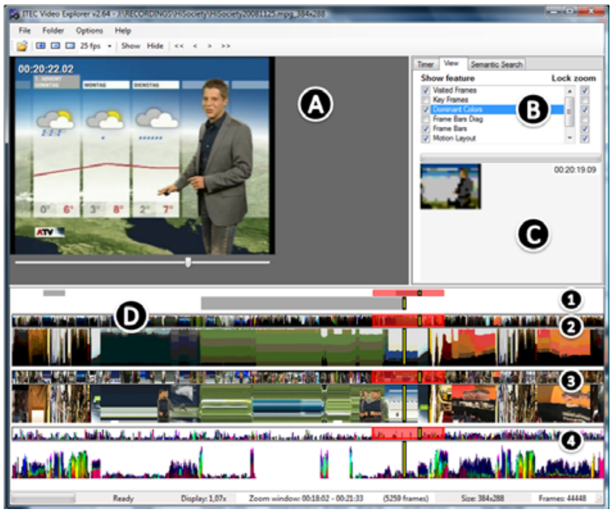

Figure 2. Our video browsing prototype using interactive navigation summaries

As already mentioned, several navigation summaries can be enabled for a specific browsing task. It is, therefore, possible that every navigation summary can be used with a different zoom window size, i.e. with different values for φ_D. For some specific browsing tasks this could be very useful, however, we also provide the possibility to lock the size and position of the zoom window across specific navigation summaries. This enables a user to see several different content abstractions in temporal combination at one glance (e.g. dominant color together with motion intensity or shot boundaries etc.) As our implementation uses a plug-in architecture with a well-defined interface, it is easy to add new interactive navigation summaries to the application.

Figure 2 shows a screenshot of a video browsing prototype using navigation summaries. It consists of four main areas. In (A) the video frames and a common time slider are displayed. (B) shows a list of available navigation summaries for the opened video sequence. Here the user can select which navigation summaries should be displayed and for which ones the zoom window should be locked (i.e. combined). (C) always shows a low-resolution preview image and the related playback time when the user moves the mouse over an overview or detailed diagram of a navigation summary. Finally, (D) shows the navigation summaries, whereas four different navigation summaries have been selected: (1) visited frames, (2) dominant colors, (3) frame stripes, and (4) motion layout. The first navigation summary looks very different, as this is not an abstract visualization of content characteristics but rather a video browsing log which shows which parts of the video have been already visited (grey in the figure) or not (white). It behaves exactly the same way as all other navigation summaries. Due to space limitations, it is not part of this paper to describe what information the other selected navigation summaries exactly show, how they are extracted and for which purpose of browsing they are suited best.

4. User Study

A user study has been conducted to evaluate the performance of our video browsing approach. In this first user study we wanted to compare its performance in relation to a standard soft video player. Comparison with other video browsing approaches, proposed in the literature, will be part of further work. We used only one single navigation summary, namely the *motion layout* summary, described in [17]. The reason is that we first wanted to find out if users understand the interaction model for a single navigation summary rather than several (maybe combined) ones. Because we also wanted to test the performance of the motion layout summary itself, we disabled the panel showing preview images on mouse moves, i.e. (C) in Figure 2. We used the following two test videos:

1. A 92 minutes long evening show (Austrian version of ”*Who wants to be a millionaire*”). Number of scenes we are looking for: 4. Average scene length: 32 seconds. Average distance between ”adjacent scenes”: 22 minutes and 54 seconds.

2. A 23.5 minutes long recording of a *ski-jumping* event. Number of scenes we are looking for: 17. Average scene length: 22 seconds. Average distance between ”adjacent scenes”: 40 seconds.

The first video was four times longer in duration than the second one and contained only a few scenes of interest with long distance in-between. The second video contained, however, roughly four times more scenes of interest than the first video with - indeed - very short distance in-between. A time limit of 10 minutes for each task has been used. The tests have been performed on a Intel(R) Core(TM)2 Duo E6550 machine with 4 GB RAM running Windows Vista 64 Bit SP1 with a screen resolution of 1280x1024 pixels.

4.1. Selected Subjects

16 subjects have been tested, 10 men and 6 women. The average age of the subjects was 28.69; the youngest subject was 24, the oldest one was 40. All subjects characterized themselves to have ”good computer skills”, 14 of them were students of informatics. Two search tasks had to be done, one with a standard video player (VLC v0.9.2) and the other one with our tool. Both tasks were executed with two different videos, whereas both the chosen tool and the

chosen video have been permutated with the number of subjects, with a latin square principle, to avoid familiarization effects. Subjects with odd sequence numbers used the VLC player for video-1 and our tool for video-2. Subjects with even sequence numbers used our tool for video-1 and the VLC player for video-2. Before starting a browsing task a subject got an introduction into the usage of the related tool by watching a three-minutes tutorial video. After that, we showed the subject a scene of interest and told him/her that (s)he had to find all the other similar scenes in the video, whereas the number of such scenes were also told. Thus, when our tool has been chosen for a task, a subject also knew how the scene of interest is visualized in the motion layout summary (the searched scenes had a special motion characteristic). We did not show subjects the advanced navigation features of the VLC player (i.e. short/medium/long jumps backward or forward by specific keyboard shortcuts) in the tutorial video, although they were free to use it if they knew some shortcuts. However, it turned out that no single subject - with good computer skills - knew any of these keyboard shortcuts.

4.2. Results

The user study revealed a lot of interesting findings. First, it showed that when looking for a searched scene in a video a user first tries to develop a search strategy and follows that strategy by trial-and-error principle. When the strategy turns out to be not (fully) valid, a user typically has a decision problem whether (s)he should completely restart from the beginning with the same strategy, because (s)he may have overlooked some scenes, or whether (s)he should develop a new search strategy. This clearly reflects the need for a ”visited frames” navigation summary, as described in Section 3 and illustrated in Figure 2. Moreover it showed that with the first video sequence subjects had much more navigation problems when using the standard video player. The reason is that the video is quite long and, thus, the resolution of the time slider is much coarser, making precise time jumps almost impossible. Moreover, since the distance between the searched scenes is much higher, a random hit is more unlikely. In addition a fast-forward would only be of minor help since it either requires long time to finish (”slow” fast-forward) or it may miss a short scene of interest (”high” fast forward).

The user study has shown that people can quickly understand our video browsing concept - as it uses familiar interaction models - and highly benefit from it. As shown in Figure 3, users of our tool were able to solve the search task 3.72 times faster for video-1 (94 secs vs. 351 secs), and 1.14 times faster for video-2 (189 secs vs. 215 secs), than users of the VLC player. However, it should be noted that the second video was very well suited for browsing with VCR-like navigation features, as provided by a standard video player, since searched scenes were very close to each other (i.e. scene $s+1$ started 40 seconds to the end of scene s, in average). Thus, a fast-forward with highest possible speed (e.g. 8x in VLC player) would be a possible strategy to find all 17 searched scenes in about 176.25 seconds (in fact, some subjects used that search strategy). Moreover, due to the high number of searched scenes and the relatively low duration of the whole video (26.5 percent of the content was part of a searched scene), random hits in video-2 were more probable than in video-1 (only 2.34 percent). The short duration of the entire video was also advantageous with regard to the resolution of the time slider. The user study has, thus, demonstrated that even for video sequences well suited for player-like navigation, a user can be faster with our tool.

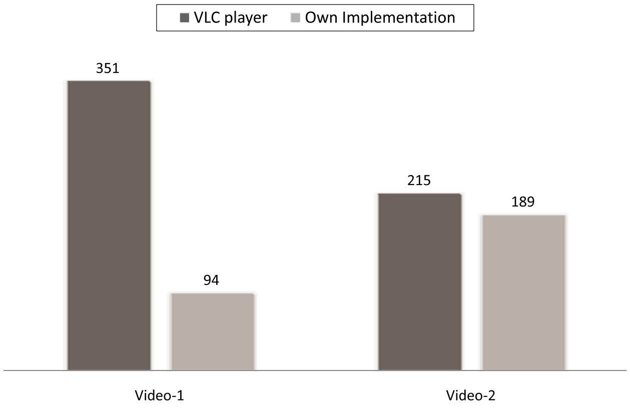

Figure 3. Result of user study: average time (in seconds) to solve a search task

4.3. System Usability Score (SUS)

At the end of the test each participant had been requested to fill out a questionnaire with some standardized questions, used to compare the System Usability Score (SUS)[3]. Altogether, users rated the VLC player with a total SUS score of 74 and our video browsing tool with a total SUS score of 79, which means that in average users found our tool more usable than the VLC player. In Figure 4, the average SUS ratings for all questions are shown[1].

While many users found that they need to learn more for using our video browsing tool (*Question 10*) and they might need the help of a technical person to use the system (*Question 4*), they also felt confident using the system

[1] This diagram shows the average of the plain rating value per question without the value correction (i.e. 5-value and value-1) and the multiplication with 2.5 (as used for the overall SUS value [3]). The score contribution for questions 1,3,5,7,9 is rated value minus 1 and for questions 2,4,6,8,10 it is 5 minus rated value.

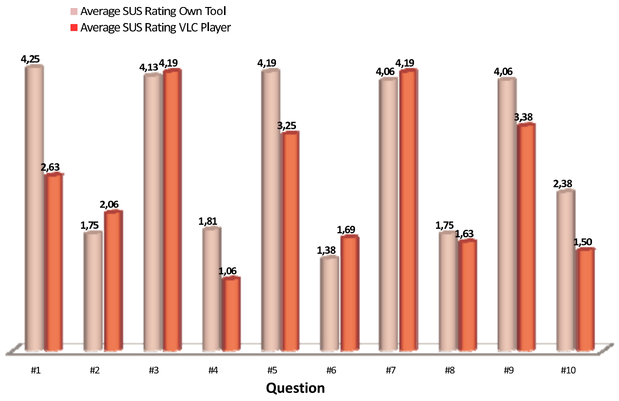

Figure 4. System Usability Score (SUS): average rating according to question number

(*Question 9*) and thought that they would like to use our tool frequently (*Question 1*). Moreover, they users imagined that most people would learn to use our tool very quickly (*Question 7*) and found it almost equal easy to use than the VLC player (*Question 3*). Interestingly, users also found that the VLC player is more unnecessarily complex than our video browsing tool (*Question 2*) (details about the standardized questions of SUS tests can be found in [3]).

5. Conclusions

We have presented a novel tool for interactively browsing videos, which allows flexible and extendable usage of interactive navigation summaries. The basic idea behind is to integrate the user's potential knowledge about the content being browsed and provide him/her simple but flexible interaction means. The concept of navigation summaries can show several types of low-level content features as well as high-level content features and, therefore, allows to implement customizable user interfaces. A first user study has shown that users highly accept our proposed approach and that they can quickly use it for beneficial video browsing. The next step is to define a meaningful set of navigation summaries and evaluate them in further studies.

References

[1] B. Adams, S. Greenhill, and S. Venkatesh. Temporal semantic compression for video browsing. In *Proceedings of the 13th international conference on Intelligent user interfaces*, pages 293–296. ACM New York, NY, USA, 2008.

[2] M. Barbieri, G. Mekenkamp, M. Ceccarelli, and J. Nesvadba. The color browser: a content driven linear video browsing tool. *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pages 627–630, 2001.

[3] J. BROOKE. SUS: a'quick and dirty'usability scale. *Usability Evaluation in Industry*, 1996.

[4] M. Campanella, R. Leonardi, and P. Migliorati. The future-viewer visual environment for semantic characterization of video sequences. In *Proceedings of the 2005 International Conference on Image Processing (ICIP 2005), Genoa, Italy, September 11-14*, volume 1, pages 1209–1212. IEEE, 2005.

[5] L. Chen, G. Chen, C. Xu, J. March, and S. Benford. EmoPlayer: A media player for video clips with affective annotations. *Interacting with Computers*, 20(1):17–28, 2008.

[6] O. de Rooij, C. Snoek, and M. Worring. Query on demand video browsing. In *Proceedings of the 15th international conference on Multimedia*, pages 811–814. ACM Press New York, NY, USA, 2007.

[7] A. Divakaran and I. Otsuka. A Video-Browsing-Enhanced Personal Video Recorder. In *Image Analysis and Processing Workshops, 2007. ICIAPW 2007. 14th International Conference on*, pages 137–142, 2007.

[8] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh. Video browsing by direct manipulation. 2008.

[9] H. Goeau, J. Thievre, M. Viaud, and D. Pellerin. Interactive Visualization Tool with Graphic Table of Video Contents. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 807–810, 2007.

[10] A. Hauptmann, W. Lin, R. Yan, J. Yang, and M. Chen. Extreme video retrieval: joint maximization of human and computer performance. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 385–394. ACM Press New York, NY, USA, 2006.

[11] W. Hürst. Interactive audio-visual video browsing. In *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 675–678. ACM New York, NY, USA, 2006.

[12] W. Hürst, G. Gotz, and T. Lauer. New methods for visual information seeking through video browsing. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, pages 450–455, 2004.

[13] W. Hürst and P. Jarvers. Interactive, Dynamic Video Browsing with the ZoomSlider Interface. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, pages 558–561, Amsterdam, The Netherlands, 2005. IEEE.

[14] M. Jansen, W. Heeren, and B. van Dijk. Videotrees: Improving video surrogate presentation using hierarchy. In *Content-Based Multimedia Indexing, 2008. CBMI 2008. International Workshop on*, pages 560–567, 2008.

[15] D. Kimber, T. Dunnigan, A. Girgensohn, F. Shipman, T. Turner, and T. Yang. Trailblazing: Video Playback Control by Direct Object Manipulation. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1015–1018, 2007.

[16] N. Moraveji. Improving video browsing with an eye-tracking evaluation of feature-based color bars. *Digital Libraries, 2004. Proceedings of the 2004 Joint ACM/IEEE Conference on*, pages 49–50, 2004.

[17] K. Schoeffmann, M. Lux, M. Taschwer, and L. Boeszoermenyi. Visualization of Video Motion in Context of Video Browsing. *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, July 2009.