Title
A Survey of Advanced Content Management Tools for TV Post-Production

Permalink
https://escholarship.org/uc/item/3g327522

ISBN
9780429107900

Authors
Bailer, Werner
Schoeffmann, Klaus
Hopfgartner, Frank

Publication Date
2012-03-01

Peer reviewed

Chapter

A Survey of Advanced Content Management Tools for TV Post-Production[1]

Werner Bailer, JOANNEUM RESEARCH Forschungsgesellschaft mbH, DIGITAL - Institute of Information and Communication Technologies, Steyrergasse 17, 8010 Graz, Austria, werner.bailer@joanneum.at
Klaus Schoeffmann, Klagenfurt University, Universitätsstr. 65–67, 9020 Klagenfurt, Austria, ks@itec.uni-klu.ac.at
Frank Hopfgartner, International Computer Science Institute, 1947 Center Street, Suite 600, Berkeley, CA, 94704, USA, fh@icsi.berkeley.edu

In TV post-production, a crucial task is to efficiently organize large amounts of audiovisual content and to select the most appropriate segments for new productions. The content sets are often sparsely annotated and the process relies crucially on the knowledge of the staff about the content of the materials. This is not only a cost issue but it also hinders the transformation of production processes towards more flexible and distributed workflows. Providing additional metadata and more intuitive ways for exploring and navigations content collections is a way to overcome this problem.

Content-based tools can efficiently support specific tasks in the entire chain of manual content management (e.g., content navigation, content selection and filtering, content annotation) and play an important role in facilitating TV post-production. In this chapter we present a survey of tools and algorithms for interactive video browsing and navigation, with a focus on two important content domains in TV production: news and entertainment content. The first includes news bulletins and news magazines, and the second encompasses series, soaps and shows, but excludes types of content that are typically produced outside the TV context, such as movies. We analyze usage paradigms for interactive search, browsing and content navigation. Finally, evaluation and benchmarking initiatives related to the tools presented in the chapter are discussed. The chapter concludes by outlining possible future

[1] This is an Accepted Manuscript of a book chapter published by Routledge/CRC Press in TV Content Analysis: Techniques and Applications on 19 March 2012, available online: http://www.routledge.com/9780429107900 or http://www.crcpress.com/9780429107900

research issues.

1.1 Introduction

Users in TV post-production need to deal with large amounts of audiovisual content that is often sparsely annotated and contains a high degree of redundancy. Their task is to efficiently organize content and select the most appropriate segments for the productions they are working on, often under time pressure. Traditionally, these processes rely crucially on the knowledge and memory of people who were involved in preproduction and content capture. This is not only an issue of production costs but it also hinders the transformation of production processes towards more flexible and distributed workflows.

Digital media asset management systems (DMAMS) provide storage, search and retrieval functionality for audiovisual content items, supporting users in navigating, viewing and finding relevant content items. Today, these systems still rely mostly on textual metadata. Content-based tools can efficiently support specific tasks in the entire process of content management (e.g., content navigation, content selection and filtering, content annotation) and can play an important role in facilitating TV post-production. Most commercial systems rely on key frames displayed in result lists, as well as low-resolution proxies for preview. Browsing is mostly restricted to using directory structures or links to different versions of content. Only few systems provide storyboard or light table views in their user interfaces. Autonomy Virage [108] is one of the few commercial systems integrating a range of automatic content analysis tools.

The focus of the chapter is on tools targeting professional users in a production context and excludes tools for end users. We describe tools and algorithms for *abstracting* multimedia content in order to allow for efficient presentations of large content sets, such as interactive video browsing, navigation and content filtering. The term *video abstract* is defined in [83] as "a sequence of still or moving images presenting the content of a video in such a way that the respective target group is rapidly provided with concise information about the content while the essential message of the original is preserved". The authors of [104] use the term *video abstraction* to denote all approaches for the extraction and presentation of representative frames and for the generation of video skims. We use the term *content abstraction* to include all approaches that aim at providing condensed representations of segments of a relevant or salient single media item or a collection of such items, independent of the purpose, context, form, creation method and presentation style of the abstract.

The existing approaches differ in many aspects, including user interfaces and usage paradigms, requirements in terms of available metadata and the type of content for which they are designed. In the following we discuss some of these aspects that are relevant in the context of multimedia abstraction for TV post-production. For a comprehensive overview and comparison of video abstraction methods see for example [104].

Content abstraction can be done manually, automatically or semi-automatically (e.g., using user input to define examples of relevant content segments [80]). A basic aspect when creating the abstract is its **purpose**, which can be to objectively summarize the content conveying all of the original message or to deliberately bias the viewer (e.g., when creating a movie trailer or a teaser for a program, cf. [65]). In the post-production use case, the purpose is to maximize the amount of information in the abstract that is needed to judge the relevance of a content item for the current task.

Somewhat related to the purpose is the **context** of the abstract, which may be undefined and independent of the initial input of the user (e.g., when a user starts browsing), it can be defined by user input, or it can be predefined, for example, when abstracts are used for representing search results and the user's query is known [21]. Domain knowledge also contributes to the definition of the context, as it helps defining the relevance of content segments. Most video abstraction approaches for sports content exploit this knowledge (e.g., goal scenes in soccer games are relevant). The context in TV post-production is given by the current production a user is working on. However, while this context is possibly defined in scripts and storyboards, it is usually not formalized in a way that is directly usable by content management tools.

A number of aspects are related to the media type of the content to be extracted. The **dimension** of the content may be a single media item (e.g., one video) or a collection of items (an example for the visualization of a content set is presented in [85]). In the latter case all items may be of the same or of different types (e.g., a mixed collection of videos, still images and audio clips). The media type also determines whether the content set has a defined **order**. For example, a video or audio stream has an intrinsic temporal order, and it is in many cases desired to keep it in the abstract. In TV post-production the dimension is given by the set of content related to a certain production. For fictional content such as movies or series, this is often 30 times or more of the duration of the final content, and the ratio can be potentially much higher, if a production (e.g., a documentary) makes extensive use of archive content.

One of the most important aspects is the **content structure**. In [114] the authors discriminate *scripted* (e.g., movies) from *unscripted* content (e.g., sports or news). Of course, the boundaries between the two are very fuzzy. Another dimension of structure is *edited* vs. *unedited* ("rushes","raw video") content, and there exist rushes for both scripted and unscripted content. For edited scripted content the abstraction algorithm can attempt to detect and use the structure of the content (such as dialogs, e.g., in [65]), while for unscripted (and especially also unedited) content other approaches are required (e.g., [19]). Content structure does not only exist on the level of the single media item, but also on the level of the collection in the case of multi-item abstracts. In some cases the collection has a "macro-structure", such as a set of rushes produced according to a script. The content encountered in production is mostly unedited (except for existing programs, from which extracts might be needed), but depending on the type of content it can be both scripted and unscripted.

There is a large variety of approaches for the **presentation** of abstracts. It can be interactive or non-interactive, sequential or hierarchical, and different media types and visualizations can be used.

This chapter discusses tools based on different ways of usage paradigms and presentation methods. Section 1.2 starts with an overview of domain independent tools for browsing and navigating content collections in post-production. The section then focuses on tools for two important content domains in TV production: news and fictional entertainment content. Section 1.3 discusses the different usage patterns and the considerations on user interfaces that are derived from them. In Section 1.4 we provide an overview of methods and initiatives for evaluating content abstraction tools and provide information on the level of maturity. Section 1.5 concludes the chapter and outlines future research topics.

1.2 Tools for Video Browsing and Navigation

The main goals of a video browsing tool are to enable interacting users to (i) quickly get an overview of the content of a video and to (ii) quickly find specific video segments in it. The first goal is important in cases where users are not yet familiar with the content, e.g., an editor starting with a new project and trying to get an overview of the raw material. The second goal is important for nearly all use cases in TV post-production. Users need efficient ways to quickly locate specific scenes, shots, or single frames in the video through appropriate interaction means and advanced visualizations of video content.

In the audiovisual media production process, different classes of content need to be treated differently. The reasons are the different amount of metadata that are available (or that one can afford to annotate), the different temporal granularity of productions (i.e., the lengths of clips being used), the level of reuse of content and the production schedules. Two important classes of audiovisual content, also commercially, are news and fictional entertainment content, such as TV movies or series. The first consists of rather short segments, that are heavily reused (especially when they are new, but sometimes also over longer periods), well annotated and typically need to be brought into productions in a hurry. The second consists of rather longer segments, specifically produced for a production according to a script, but otherwise typically not well annotated. This section first discusses approaches that are not targeting a specific content domain, and then discusses tools focusing on news and fictional entertainment content respectively. Sports is the third important content domain in TV production, but due to the large amount of quite specific approaches for different types of sports, there is not enough room to include the topic in this chapter.

A remarkable amount of work can be found in the literature that focuses on automatic detection of content classes (e.g., sports, news, fictional entertainment, etc.). These *multimedia genre classification* methods usually employ machine learning algorithms (such as Support Vector Machines [40]) on content descriptors (e.g., bag-of-visual-words [115]) in order to train classifiers that are able to automatically detect different genres, often also with fine granularity (e.g., different kinds of sports content). However, multimedia genre classification is a large field of research; a complete survey on that topic is out of the scope of this chapter. The interested reader is referred to recent literature, such as [58, 72, 110].

1.2.1 Domain Independent Approaches

Simple Video Players

An example of a simple but commonly used video browsing tool is a video player providing a time-slider, often also a fast-forward feature, in addition to the playback feature. Such a video player is a very convenient tool for video browsing as it uses simple and well-known interaction means. Maybe more important, it is immediately usable for any video file without any preprocessing delay, which an advanced video retrieval tool would require for content analysis. On the other hand, however, a common video player is very poorly suited for the task of quickly locating specific parts in a long video, because the resolution of the time-slider is way too coarse in order to be efficiently used for frame-based search. Moving the time-slider by just one pixel may result in a jump of several seconds or minutes instead of only one frame.

This type of video player with a time line is commonly used in content management tools, and is also included it all 28 commercial media asset management systems we surveyed.

Enhanced Video Players

In order to overcome the limitations of a common time-slider (also known as *seeker-bar*), several improvements of the interaction model have been proposed [52, 51, 30, 87]. Hürst et al. [50, 51, 52] propose the *ZoomSlider* (see Figure 1.1), which is a virtual hidden time-slider available on the entire player window. When the user clicks on any position in the player window, a time-slider for moving backward or forward appears. The granularity of that time-slider is dependent on the vertical position of the mouse in relation to the entire height of the player window. When the mouse is moved in vertical direction the scaling of the time-slider changes in a linear way. The finest granularity is used at the top and the coarsest granularity is used at the bottom of the window. Therefore, a user can zoom-in or zoom-out the scaling of the time-slider by selecting different vertical mouse positions.

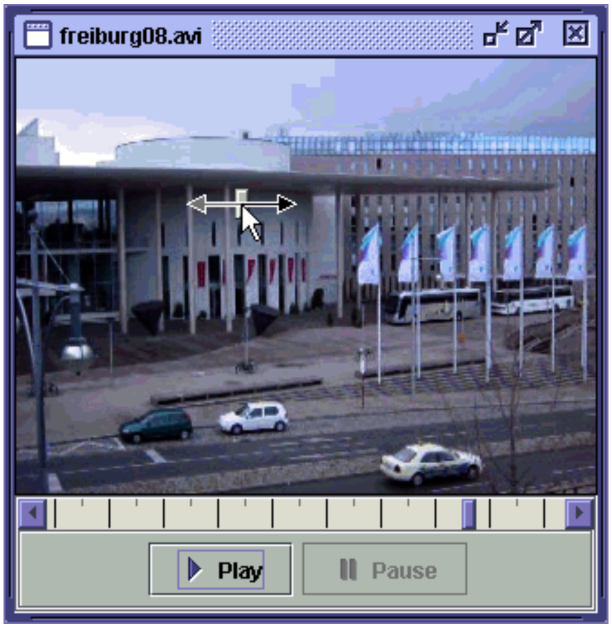


Figure 1.1: Video Navigation with the *Zoom Slider* [51], ©2005 IEEE

Traditional video tape recorders also allow improved navigation in video by *shuttle* and *jog* controls. A shuttle control enables to easily change playback speed through wheel-rotation. Complementary, a jog control enables to quickly change direction of play (forward, backward) through an additional wheel. The combined usage of shuttle/jog controls allows for a convenient navigation through a video. This interaction model has been adopted by video players and video editing products (e.g., Apple's Final Cut [54], Avid's DS System [55], or MAGIX's Movie Edit Pro [3]) that provide a virtual wheel for improved navigation. It allows a user to more accurately locate specific positions/frames in video.

Dragicevic et al. [31] propose *relative flow dragging* as an alternative to improved time-sliders and shuttle/jog controls for frame-accurate positioning. Relative flow dragging is a technique to move forward and backward in a video by direct mouse manipulation of content objects. They use an optical flow estimation algorithm based on SIFT [67] salient feature points of two consecutive frames, which is very computationally intensive and, thus, time-consuming. A user study was conducted, where specific events in videos had to be found (e.g., *Where does the car starts to move?* or *Where does the ladybug pass over a specific point?*). In a *2 technique* × *2 task* within-participant design, subjects were asked to solve tasks either with relative flow dragging or with a traditional seeker-bar. The results of their study show that participants were at least 2.5 times faster with relative flow dragging than with the traditional seeker-bar.

Pongnumkul et al. [87] propose an interaction model for a time-slider that integrates both *low-speed* navigation and *high-speed* navigation into one single control element. Their *elastic timeline* dynamically switches navigation granularity based on the horizontal and vertical distance of the mouse pointer to the handle (thumb) of the time-slider. If the user keeps the mouse pointer close to the handle, the time-slider works in low-speed browsing mode. When the user pulls the mouse pointer far away from the handle, the time-slider enters high-speed browsing mode.

Such improved interaction models allow a more flexible navigation in the video and/or more accurate locating of specific content units (scenes, shots, frames). However, they don't give any additional visual information about the content structure. Therefore, several visual enhancements of the common time-slider have been proposed in the literature. For instance, Barbieri et al. [11] propose to enhance the background of the time-slider with vertical color lines (see Figure 1.2), representing information about the content (e.g., the *dominant color* of every frame, or the *volume level* of the corresponding audio channel). This idea helps to reduce the number of interactions with the time-slider as it can directly show where shots start or stop and where segments of a specific dominant color of audio volume are located. In order to be appropriate for both short and long videos they propose to use two time-sliders of different sensibility, e.g., the first one for fast navigation and the second one for slow navigation in a particular segment. A similar idea has been presented by Moraveji [75], who propose to display distinctive colors in the background of the time-slider as abstraction for different semantic concepts (e.g., cars, persons, faces, etc.). Chen et al. [17] use the same idea to visualize emotions of a specific actor/actress in sitcoms with the *EmoPlayer*. Rehatschek et al. [91] use *frame stripes* as time-slider to improve navigation in a video. Frame stripes are images constructed by adjacent visualizations of the center columns of frames. This kind of content abstraction implicitly conveys information about content structure and motion of a scene (in addition to the general color tone of a scene).

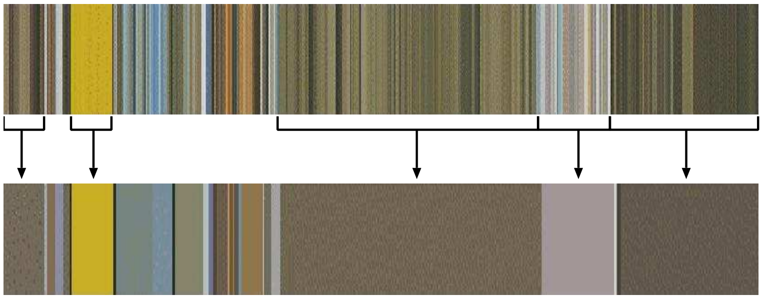


Figure 1.2: Visualization of the *ColorBrowser* without(above) and with(below) a smoothening filter [11], ©2001 IEEE

A more general concept of *Interactive Navigation Summaries* (INS) is proposed by Schoeffmann et al. [96]. An INS consists of an *overview* component that contains a zoom-window for a specific time segment of a video, which can be moved and resized by the user. The zoom-window defines the time segment of the video shown in the *detailed* component. Both components act as time-sliders but visualize abstract information about the content in the background. Several examples of content abstractions (see Figure 1.3) for INS have been presented (e.g., *color flow* [97] and *motion flow* [99]).

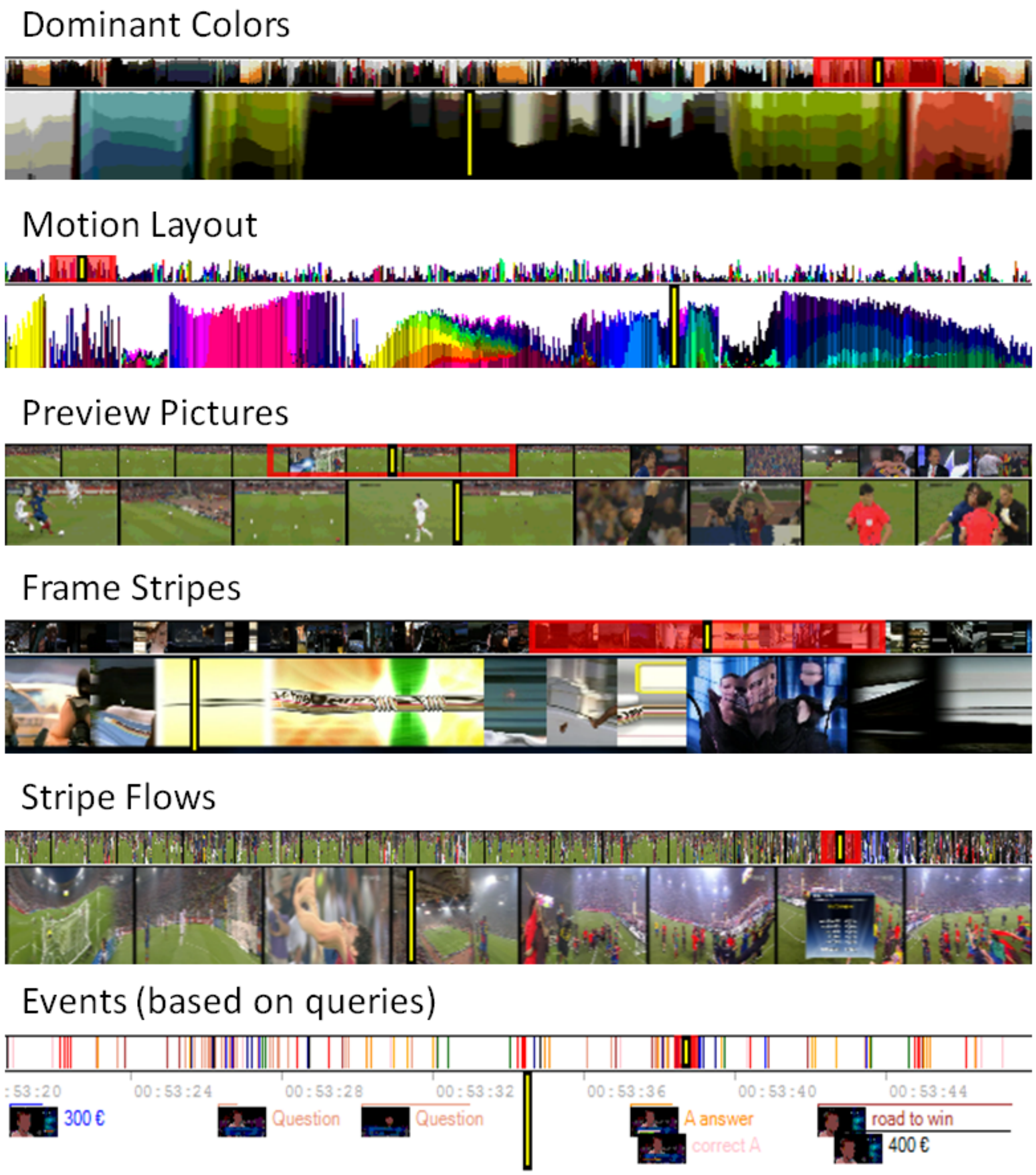
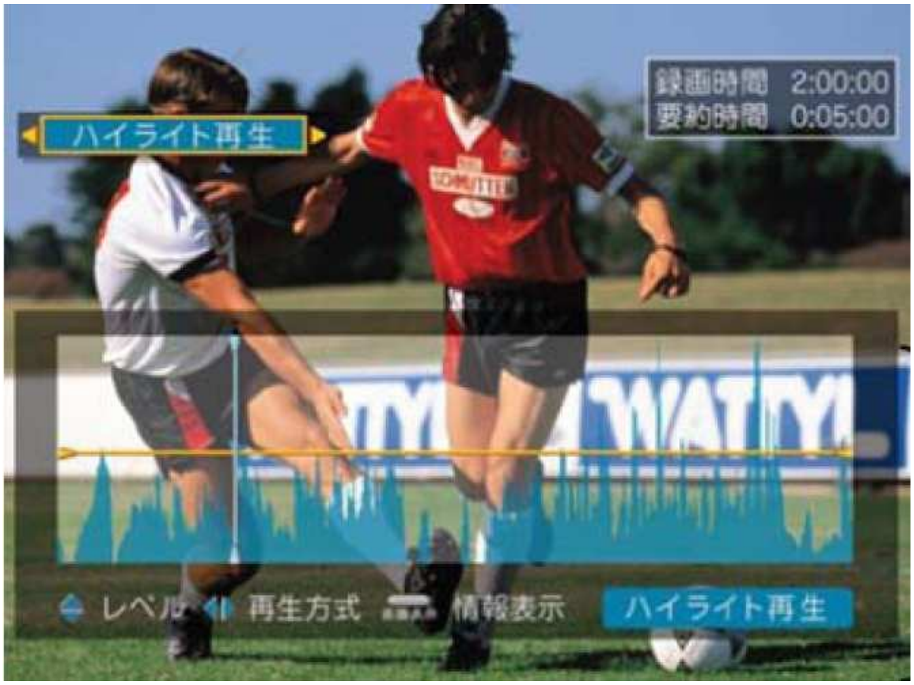


Figure 1.3: Examples of different implementations of *Interactive Navigation Summaries*. (for details see [95])

The fast-forward mode (e.g., 2x, 4x, 8x playback speed) provides not only a simple way of getting a quick overview about the content of a video but also a convenient way to jump to the next segment of interest. In a large user test with 200 users, Crockford et al. [24] found out that when searching for specific content in a video with a VCR-like control set (play, pause, fast-forward, fast-reverse, stop) *speed-switching* is the predominant search technique (most-used and fastest). Speed-switching is defined as a combination of play and fast-forward, where the user switches the playback speed according to the experienced level of interest (for example: play, 2x, 4x, 2x, play). So, when users believe the searched content is hardly to appear in the current segment, the switch to fast-forward and switch back to play when they think the searched scene is just to appear. Therefore, improving fast-forward may have a high significance for the performance of search tasks in videos.

Divakaran et al. [82] propose to use an *adaptive fast playback* feature (see Figure 1.4) that is based on the complexity of scenes in the video. The main idea is to enable quick skimming through a video by an adaptive playback function that automatically plays complex scenes at lower (or normal) speed and less complex scenes at higher speed. In order to determine the complexity of scenes content analysis is used. The same idea was used by Cheng et al. [18] for the *SmartPlayer*. It uses an automatic *playback speed adaptation* function that is based on scene complexity, which is learned through motion analysis. Their player has been designed in accordance with the ”scenic car driving” metaphor, where a driver slows down at interesting areas and speeds up through unexciting areas.

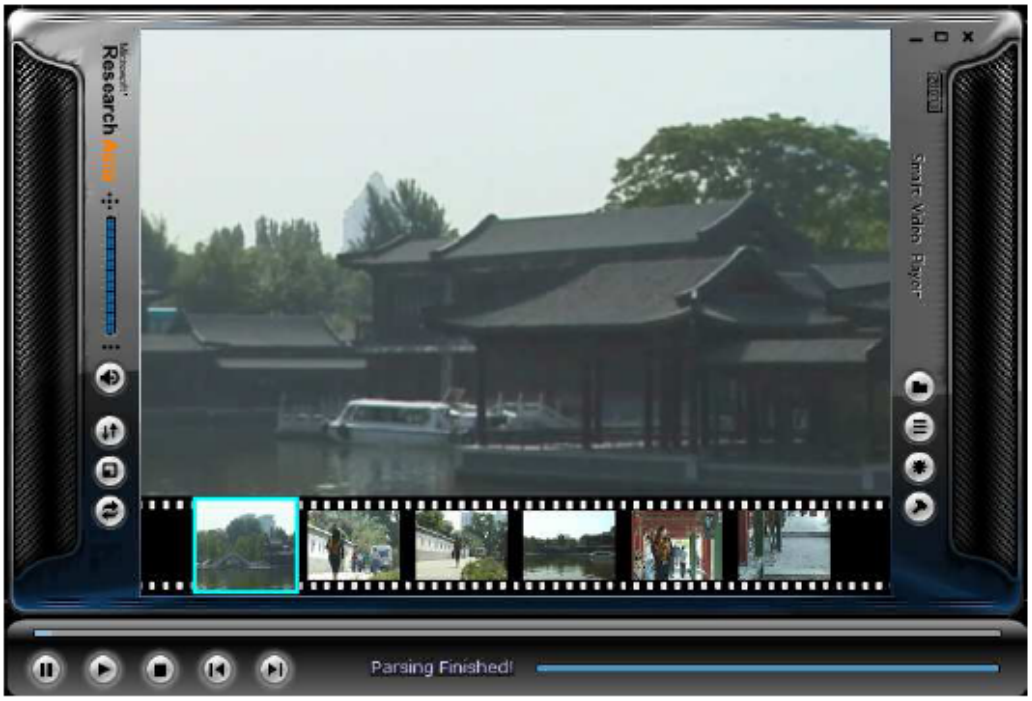


録画時間 2:00:00
要約時間 0:05:00

Figure 1.4: A video browsing enhanced personal video recorder[28], ©2007 IEEE

Pongnumkul et al. [86] also propose a similar feature for dynamic video skimming, which is, however, more sophisticated. They argue that fast-forwarding at very high speed is very hard to perceive by a human because only unrelated frames are shown to the user. So instead of simply speeding up the whole content, they play short clips (one second) at a playback speed of 2x and perform a discrete jump forward to maintain the desired average video speed. In order to perform ”good jumps”, they use a shot-boundary detection method to find out all key frames and generate *key-clips* that are located one second around the key frame. The jump is always performed to the nearest key-clip. Yang et al. [16] propose the *Smart Video*

Player (see Figure 1.5) to facilitate browsing and seeking in videos. It provides a filmstrip view in the bottom part of the screen, which shows key frames of the shots of the video. The user can set the level of detail for that view and, thus, extend or reduce the number of shots/key frames displayed within the filmstrip.



Parsing Finished!

Figure 1.5: The Smart Video Player [16], ©2008 IEEE

Approaches based on Key Frames

Storyboards A *storyboard* is a grid-like alignment of key frames of shots that is used to browse through the content of a video or through the results of a query in a video retrieval application. As one of the first, Arman et al. [5] proposed to use the concept of *key frames*, which are representative frames of shots, for chronological browsing the content of a video sequence. For every shot a key frame is selected and the storyboard as chronological list of key frames is used to browse through the content of a video. For visualization of the results, they propose to display good results in original size (e.g., 100%), somewhat similar results in smaller size (e.g., 33%) and bad results in an even smaller size (e.g., 5%). A user study of Komlodi et al. [62] in 1998 revealed that simple storyboards are preferred by users over dynamic approaches, even if additional time is required to interact with the user interface (scroll bars) and for eye movements. Dynamic approaches such as slide shows often display the content with a fixed frame rate and do not allow the user to adjust it. Storyboards have been used as basis for many tools, especially in the field of video retrieval. A comprehensive study is out of the scope of this book chapter but can be found in [98]. More recent approaches, such as the *VisionGo* system [77], use keyboard shortcuts to quickly scroll through the list of key frames and/or to provide relevance feedback. Practically all commercial media asset management systems extract key frames for representing media items (many shot-based or at fixed intervals) and use them in storyboards or content time lines. In the latter case, either time line or light table visualizations are used.

Content Hierarchies Jansen et al. [57] propose to use *VideoTrees* for navigation through a video (see Figure 1.6). A VideoTree is a hierarchical tree-like temporal presentation of a video through key frames. The key frames are placed adjacently to their parents and siblings such that no edge lines are required to show the affiliation of a node. With each depth level the *level-of-detail* increases as well until shot-level granularity. For example, a user may navigate from a semantic root segment to one of the subjacent scenes, then to one of the subjacent shot groups, and finally to one of the subjacent shots. 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 show that the VideoTrees can outperform storyboards (a matrix-like alignment of key frames) regarding the search time (1.14 times faster). However, the study also reveales that users find the classical storyboard much easier and clearer.

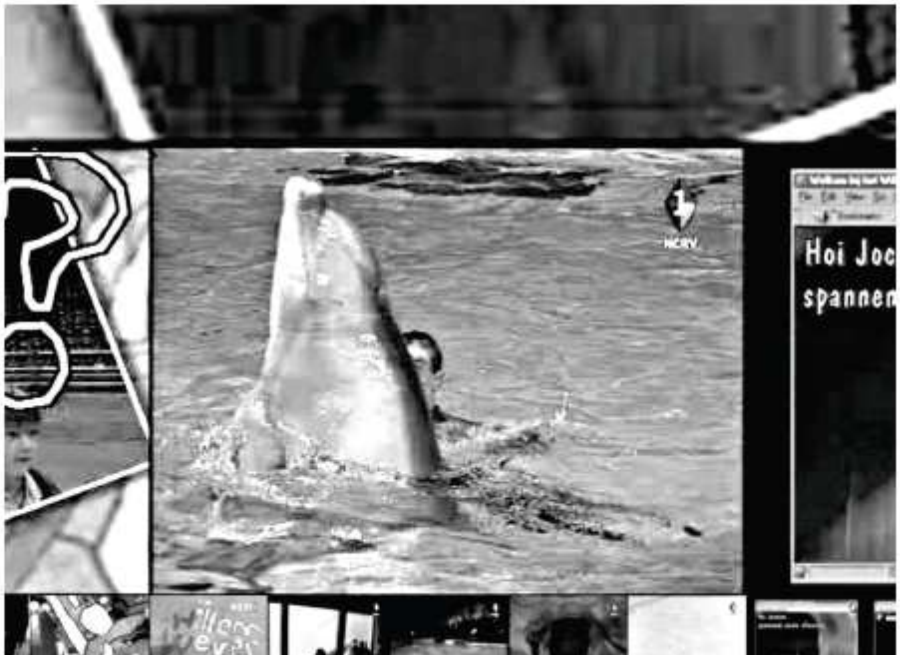


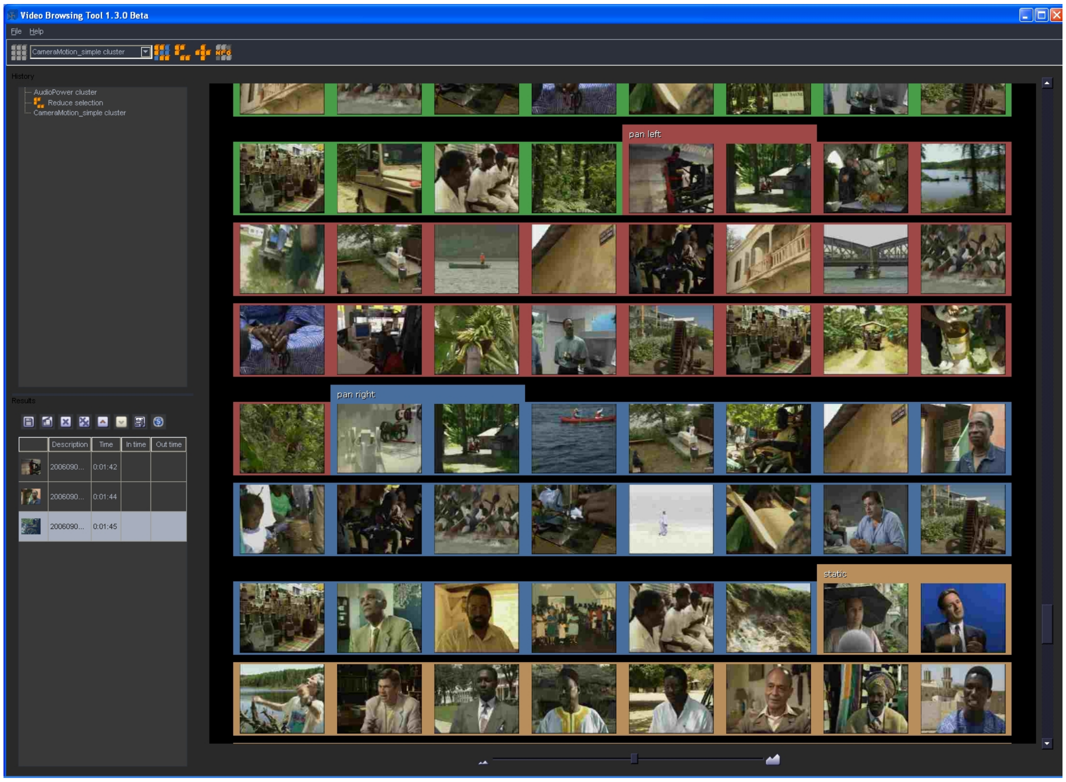
Figure 1.6: Video browsing with the *VideoTree* [57]

Eidenberger [35] proposes a video browsing approach that uses similarity-based clustering. More precisely, a *Self-Organizing Map* (SOM), which is a neural network that uses feed-forward learning, is employed as a similarity-based clustering method. Visually similar segments of a video are grouped together and visualized in hierarchically organized index trees. The clusters are visualized as hexagonally shaped cells showing key frames of shots. The user can interactively select a certain cell and step one layer deeper in the hierarchical tree structure to see more details of the selected shot. Del Fabro et al. [26] propose a hierarchical video browser that enables uniform temporal decomposition of a video by interaction. The browser uses an $m \times n$ storyboard that contains key frames uniformly sampled from the video content. The user is able to specify the size of the storyboard (e.g., select m and n) and go into details of a selected segment by a left-mouse click. For instance, if $m = 2, n = 2$ the four key frames at the top-level are uniformly selected from the whole video (one key frame per quarter) and when the user clicks on the second key frame, for instance, the storyboard is reloaded with new key frames from the second quarter segment of the video. In that way a user can delve into details of a specific segment until frame-level granularity and go back the hierarchy by a right-mouse click. Each segment can be further inspected by an own time-slider and a double click will start playback for the corresponding segment. The browser is called the *Instant Video Browser* as it requires no content analysis and is immediately usable for a newly recorded video.

Bailer et al. [10] argue for an automatic generation of condensed abstracts that shall ease access to videos. Therefore, they introduce a multimedia content abstraction process that aims at providing condensed representation of segments of video documents. The process can be divided into five generic steps, the most important ones being clustering of video content in order to identify similarities, selection of representative clusters and the presentation of these clusters in a graphical user interface (see Figure 1.7). In their paper, they introduce a video browsing tool for content management which is designed based on these steps. The central component of their graphical user interface is a light table where video segment clusters are displayed by representative key frames. The user can manipulate this representation by selecting different features used for clustering (e.g., camera motion, visual activity, faces, etc.). The user can further decide to select a subset of clusters that seem to be relevant and discard the others. This way, the system supports the user in exploring the video content step by step.

3D(-like) Visualizations In the last decade most computer systems became equipped with hardware to support 3D graphics. Therefore, video search and navigation tools could also take advantage of 3D graphics: (i) use 3D transformations (e.g., rotation) in order to display more key frames on a 2D screen at-a-glance instead of planar arrangements, (ii) allow users to interactively navigate through a 3D space of content representation and focus on content of interest only. Instead of making all key frames equally visible, 3D graphics would also inherently allow to display more important key frames at larger size and less important key frames in smaller size. With interactive navigation users would be able to change the perspective to their current needs, so that key frames considered as less-important by the system will be larger in size. However, it is not clear yet whether such a 3D representation is beneficial for interactive search tasks. In the literature only a few approaches for video search and navigation can be found that take advantage of real 3D graphics.

In an early work, Manske [69] proposes to use a *Cone-Tree*-like representation of key frame hierarchies. A "side-view" has been chosen as perspective enabling to see many transformed key frames at a glance. A user is able to open and close sub-trees in order to inspect specific segments in more detail and to switch to a front-view perspective based on a selected key frame. Moreover, a feature for an automatic *walk through* the whole video-tree in its temporal sequence is supported. Unfortunately no evaluation has been done, which would assess the performance of such a content representation. The Mitsubishi Electric Research Laboratories (MERL) propose several techniques for improved navigation within a video through content presentation with rather simple 3D transformations. For example, the *Squeeze* layout and the *Fisheye* layout [27] (see Figure 1.8) are proposed for improved fast-forward and rewind with personal digital video recorders. With a user study they show that their approach can


Video Browsing Tool 1.3.0 Beta
File
Help
CameraMotion_simple cluster
History
AudioPower cluster
Reduce selection
CameraMotion_simple cluster
Results
pan left
pan right
static

	Description	Time	In time	Out time
	2006090...	0:01:42		
	2006090...	0:01:44		
	2006090...	0:01:45		

Figure 1.7: Video Browsing Tool as proposed by Bailer et al.[10]

significantly outperform the VCR-like navigation set in accuracy. However, no significant difference was found in the task completion time.

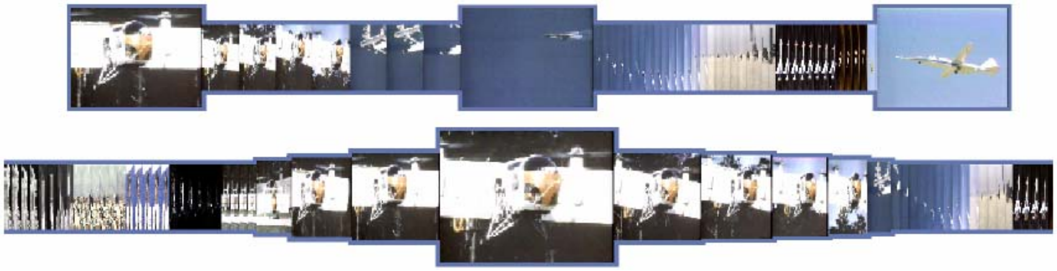


Figure 1.8: The *Squeeze* and *Fisheye* layout for improved Fast-forward and Rewind [27], ©2007 IEEE

De Rooij et al. [25] propose the *CrossBrowser* that uses a 3D-like visualization of key frames. The interface consists of two key frames/shot paths, called *video threads*. The *horizontal* video thread adheres to the temporal sequence of shots in the video, while the *vertical* video thread displays visually (or semantically) similar shots according to the user-selected center-shot of the horizontal thread.

1.2.2 News Content

An important characteristic of news content is the fact that clips are entirely or partially reused in many broadcasts, not only within one organization, but also across organizations, using, e.g., content from distribution networks such as Eurovision. When analyzing complete news broadcasts, the segmentation into the individual news stories is also a relevant topic. In addition, near duplicates exist, as at many events the cameras of different broadcasters record the same scene from different, but nearby positions. Finally, news stories develop over time, thus there is content often showing the same scene at different times, and in some cases the content differs only marginally. Grouping related clips of news content is thus an important tool to organize news content, known as news topic threading.

Browsing

For browsing story-based video content like news, interviews, or sports summaries Goeau et al. [36] propose the *Table of Video Contents* (TOC). Several types of content features are used to decompose the content into semantic units, which are the basis for an improved visualization that facilitates browsing and navigation. Their visualization uses a *Video Backbone* that can effectively show the content structure. For a news video, every news story is shown as a loop of key frames originating from that backbone, while the rather short moderation scenes appear directly along the backbone. Tang et al. [103] propose the *NewsEye*, an application for improved news story browsing. With unsupervised fuzzy k-means clustering the content is first segmented into shots. Then, the shots are grouped together in order to form several different news stories. Their VideoPlayer-like interface contains an own panel showing key frames of all the shots in the current news story as well as the detected caption text based on OCR, which can also be used with a text-based search. Liu et al. [66] propose a news video browsing system called *NewsBR*, which is very similar to the above mentioned NewsEye. The news story segmentation uses a shot detection method, silence clip detection and OCR to detect text of a news topic. Their interface shows a TOC (in combination with a key frame preview) according to the story segmentation which can be used as navigation means. It also provides a keyword-based search on the extracted caption text. Vakkalanka et al. [106] describe a news video indexing, browsing and retrieval system, called *NVIBRS*. Their system performs shot detection and news story segmentation based on localization of anchorperson frames. The interface of their application provides a tree view of all detected news story units in a video and shows key frames of the currently selected story as a navigation means. It also allows a user to perform a textual query by specifying the desired video category as the news content is categorized into a few categories.

News Story Segmentation

In the news video domain, the coherent segments of a broadcast are news stories, commonly defined as segments with a coherent news focus which contain at least two independent declarative clauses. Chaisorn et al. [15] argue that the internal structure of news stories depends on the broadcast station's style. While some stories consist of anchor person shots only, often with a changing background image, other stories can consist of multiple different shots, e.g., other anchor persons, graphics or animations, interview scenes or shots of meetings. News story segmentation is essentially finding the boundaries where one story ends and the other begins.

Various text-based, audiovisual-based and combinations of all features have been studied to segment news videos accordingly. Based on the observation that the structure depends on the broadcast station, some methods use prior knowledges for the segmentation. The detection of the repeated appearance of anchor person shots is exploited by several methods, either by using classifiers for anchor person shots [48, 84] or similarity search [109, 117].

The basic features employed in many of the approaches are visual similarity between shots within a time window and the temporal distance between shots, e.g., [79, 39, 101, 34]. Some approaches use additionally the similarity of faces appearing in the shots [39]. The audio signal is used in many approaches to detect pauses, speaker changes or changes between music and speech. Other audio-based methods are the detection of jingles [90] and the detection of changes in the acoustic environment, such as changes of the signal-to-noise ratio (to discriminate studio from outside shots) [39]. A number of approaches also use text from transcripts or automated speech recognition. Some use the text to find similar word appearances in different shots [48, 39] or watch for trigger phrases that indicate certain types of shots [34].

Some of the reported approaches are based on supervised learning approaches. The approaches discussed in [48] and [101] are based on classifying boundary candidates into story boundaries and non story boundaries using the expectation-maximization algorithm (EM) and support vector machines (SVMs) respectively. The approach in [15] is based on shot classification using a hidden Markov model (HMM). The approach reported in [117] uses a model called shot connectivity graph (SCG). Shots are classified and each node in the graph represents a shot, the edges are transitions from one shot to another. As it is expected that anchor person shots reappear, the task is to search for cycles in the graph. Special types of shots are detected using other features, such as word spotting for detecting sports shots and greenish/bluish color impression to detect weather shots. Detailed surveys of the earlier research on news story segmentation are given by Arlandis et al. [4] and Chua et al. [22].

Recently, research has moved to combining more modalities, applying more powerful language processing, including multilingual approaches (e.g., [92]), and use of contextual knowledge. In [71] anchor shots are detected based on both visual and speaker features. In addition, the fact that most story changes are related to a change of the type of of camera shot is exploited. [74] use Latent Dirichlet Allocation (LDA) on the text transcript in addition to the detection of anchor person shots. Also the approach presented in [68] uses Latent Semantic Indexing (LSI) in addition to cue word spotting, anchor shot detection and audio type detection. The authors of [107] model the social network between the persons appearing in the news, and create a segmentation based on the assumption that persons appearing in the same news story are stronger related than those appearing in different stories. The authors of [88] have conducted a thorough survey of the performance of different types of features for news story segmentation.

News Story Threading

Near-duplicate detection is used for topic tracking of news stories as they develop over time. Some approaches work on a story level and use features such as speech transcripts [49] that are often not available in other content such as rushes. The approach proposed in [53] uses a transcript (obtained e.g., from ASR) to perform news story segmentation and topic threading.

The approach proposed in [33] works on matching sequences of key frames, tolerating gaps and insertions. Starting from single matching key frames, the search for matching sequences is performed temporally around these key frames. The authors of [113] propose a co-clustering approach for near duplicate key frames based on a bipartite graph model including both visual and textual information. In [118] an approach combining textual matching of ASR transcripts, matching of extended face regions for key frames showing faces and affine matching of non-face key frames is proposed. An approach using visual concept detection, together with time constraints is proposed in [60]. An important result of this paper is that the most useful concepts for story tracking are settings, followed by named persons.

In [94] a near duplicate video detection approach based on feature point trajectories is proposed. An inconsistency descriptor is extracted from a spatio-temporal patch around the feature points and from a binary discontinuity sequence is determined by detecting local maxima after Gaussian smoothing of the inconsistency sequence. Efficient matching is performed using the discontinuity sequence representation and histograms representing temporal offsets.

1.2.3 Fictional Content

In film and video production usually large amounts of raw material ("rushes") are shot and only a small fraction of this material is used in the final edited content. The reason for shooting that amount of material is that the same scene is often taken from different camera positions and several alternative takes for each of them are recorded, partly because of mistakes of the actors or technical failures, partly to experiment with different artistic options. The action performed in each of these takes is similar, but not identical, e.g., has omissions and insertions, or object and actor positions and trajectories are slightly different. In addition there are takes that stop earlier (mostly due to mistakes) or start in the middle of the scene ("pickups"). The result of this practice in production is that users dealing with rushes have to handle large amounts of audiovisual material which makes viewing and navigation difficult. In post-production, editors need to view and organize the material in order to select the best takes to be used. The ratio between the playtime of the rushes and that of the edited content is often 30:1 or more.

The task of the editor can be facilitated by identifying near-duplicate video segments in order to group them. One of the most prominent applications for near-duplicate detection is finding illegal copies of video content (cf. [38, 37]). A related problem is the identification of known unwanted content in public access video databases [23]. These applications are based on the following assumptions: (i) the actual content of the videos to be matched is identical, (ii) partial matches need to be identified and (iii) the algorithm needs to be robust against a number of distortions, such as changes of sampling parameters, noise, encoding artifacts, cropping, change of aspect ratio etc. The first assumption is not valid for clustering takes of fictional content, while robustness to distortions is only necessary to a limited degree in this application, as the content to be matched is captured and processed under similar conditions.

The authors of [93] propose an approach for clustering repeated takes into scenes using agglomerative hierarchical clustering of feature vectors. In [78], a near duplicate key frame detection approach based on PCA-SIFT matching between feature points extracted using the Hessian-affine detector is proposed. Using an efficient indexing structure, approximate k-NN matching between PCA-SIFT descriptors is performed, looking for parallel or zoom-like patterns of matching between key frames. The matching sequences are found using transitivity of matches and temporal proximity constraints.

The problem of matching video segments can also be transformed into a problem of matching sequences of feature vectors extracted from the video segments. The feature vectors can contain arbitrary features and can be sampled with different rate from the videos. The task is then to find suitable distance measures between sequences of these feature vectors. Two classes of approaches have been proposed for this problem. One is based on the Dynamic Time Warping (DTW) paradigm [76], which tries to align the samples of the sequences so that the temporal order is kept but the distance is globally minimized. The approach has been applied to detecting repeated takes in rushes video [61]. The authors of [102] propose a method that is conceptually very similar to DTW but includes further strict constraints, e.g., it is assumed that start and end of the two video segments are temporally aligned and only the content in between may vary in timing. The distance measure Nearest Feature Line (NFL) [119] is also conceptually related. It does not align samples of the two sequences but calculates the nearest point as the intersection of a line that is orthogonal to the line between two samples in feature space and passes through a sample of the other sequence. The distances in feature space between the intersection points and the corresponding points in the other feature sequence are summed to yield the total distance of the sequences.

The other class of distance measures is based on the idea of the edit distance between strings, i.e., the cost of inserting, deleting or replacing samples in the sequence. The authors of [2] propose such a measure called vString edit distance. The values of vectors in the feature sequence are mapped to a set of discrete symbols and three new edit operations are introduced: fusion/fission of symbols (in order to deal with speed changes), swapping of symbols or blocks of symbols and insertion/deletion of shot boundaries. The distance is defined as a weighted linear combination of the traditional edit distance (using only equality or inequality of the symbols) and a modified one taking also the difference between the symbol values into account. The drawbacks are that the sequence of feature vectors needs to be mapped to a discrete set of symbols and that operations such as fission/fusion and handling of shot boundaries need to be modeled separately. In [7] a method based on the Longest Common Subsequence (LCSS) model, a variant of the edit distance supporting gaps in the match, has been proposed. The method addresses the problem of clustering repeated takes of a scene, which might have insertions, omissions or different timing. The authors of [12] also propose a method for matching clips in video databases using color similarity and edit distance, avoiding discretization of the feature vectors. A similar approach using the edit distance and different costs for substition, deletion and insertion is proposed in [116]. The method for clustering takes into scenes which is proposed in [32] also uses a variant of the edit distance. Hierarchical clustering of segments with duration one second is performed and yields the cost for matching two segments, which together with a cost for gaps is used to determine the optimal alignment.

1.3 Paradigms for Interactive Search, Browsing and Content Navigation

An important concept in interactive video search is the design of graphical user interfaces that allow the users to both express their information need and to interact with the retrieval results. As we have shown in the previous section, the specific nature of video data requires rather complex graphical user interfaces. In this section, we introduce different interaction patterns that can be observed when interacting with such systems. Understanding usage interactions is of particular interest in the field of adaptive retrieval since users' implicit interactions can be interpreted in order to identify users' interests and to adapt retrieval results accordingly (e.g., [44, 47]). These usage patterns have the potential to significantly improve the performance of content management tools. A large variety of different interface designs exist and thus, the way users interact with these interfaces differs significantly from their textual counterparts. Graphical user interfaces of both textual and multimedia domains are designed to assist users in their information seeking task. Dix et al. [29] argue that user interactions in interactive systems can be represented as a series of low-level events, e.g. key presses or mouse clicks. These events are the most basic interactions that users can perform during their interaction. The interfaces that have been surveyed in [98] provide various low-level events that users can trigger while interacting with given documents. Any action that users perform during their information seeking activity, further referred to as their search session, consists of a series of these events. A further analysis reveals the following six events:

- *Previewing:* Hovering the mouse over a key frame. This can result in a tool tip showing neighbored key frames and additional text or in highlighting the query terms in the text associated with the key frame. This low-level event indicates further interest in a key frame as the user receives additional information about the result.

- *Clicking result:* Click, e.g., on a key frame, to trigger playback of a video shot or to perform further actions. This event indicates the users' interest in the video shot which is represented by the key frame.

- *Sliding:* Using the sliding bar to navigate through a video. This event indicates further interest in the video. Users appear to slide through a video when the initial shot is not exactly what they were searching for but when they believe that the rest of the video might contain other relevant shots. Hence, the initial shot might not be an exact match of the users' need but raises hope to find something of relevance in the same video.

- *Exploring:* Looking at metadata (date of event, capture location, production notes,...). By performing this event, users show a higher interest in the current shot, as they want to get additional information. This information can help them to judge about the relevance of the shot. A user for example might search for a specific news event such as a certain speech of the president of the United States. In such cases, the direct correlation between capture date and event date can help to identify relevant shots of the speech itself, comments on expectations from just before and reactions of politicians and commenters shortly afterwards.

- *Browsing:* Browsing through a video by clicking on its neighbored key frames. Similar to using the sliding bar to navigate through a video, this feedback indicates users' interest in this shot. Unlike using the sliding bar, browsing indicates that users suspect a relevant shot in the neighborhood of the current shot.

- *Viewing:* Viewing a video. The playing duration of a video might indicate users' interest in the content of the video.

Bezold [13] describes such event series as probabilistic finite-state automata. Considering that each low-level event combination within a user's interaction sequence depends on the preceding event, we argue that user interactions can be simplified in a Markov Chain [73]. Markov Chains consist of states and transitions between these states. A state change is triggered by a certain event with a certain probability. In the remainder of this section, we introduce example Markov Chains that represent possible user action sequences consisting of low-level events when users interact with a given document using an interactive video retrieval system. Note that due to simplicity reasons, the scenarios cover some possible user interaction, not necessarily a user interaction including all features an interface can provide.

User Action Sequence S_1 (Preview-Click'n'View)

Sequence S_1 combines three different low-level events, encompassing all interfaces that provide the minimal functionalities of previewing, clicking on a key frame in the result set and viewing the video shot. Due to these functionalities, we refer to this sequence as "Preview-Click'n'View". Example interfaces that allow this event combination have been presented in [20, 45]. Possible low-level event combinations are visualized in Figure 1.9.

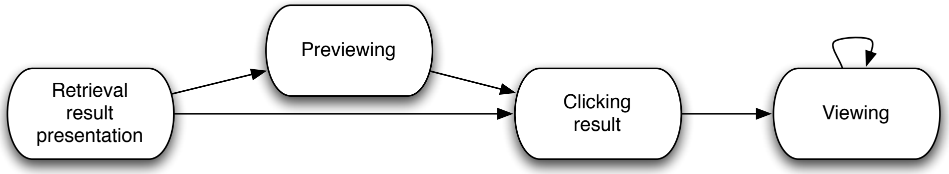


Figure 1.9: Possible event combinations on a given document in Sequence S_1

Given a displayed document, denoted "Retrieval result presentation" in above figure, this sequence models users (i) hovering the mouse over listed key frames to get some additional information of the shot, e.g., in a tool tip (previewing). Further, the users may (ii) click on the key frame (clicking result) to (iii) start playing a video (viewing).

User Action Sequence S_2 (Click'n'Browse)

Sequence S_2, referred to as "Click'n'Browse", combines two low-level events that can be given when interacting with a document: clicking a result on a result list to display the video and its key frames, followed by browsing these key frames. An example interface supporting this sequence is introduced by Heesch et al. [41]. In this interface, information is presented on different panels. Retrieval results are represented by key frames. Clicking on one key frame in a result panel will set focus on that key frame and update all other panels. One panel contains the neighbored key frames in a fish eye presentation. In this panel, a user can browse through the results. Possible low-level event combinations are visualized in Figure 1.10.

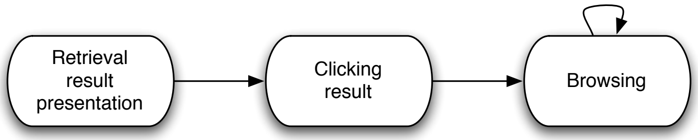


Figure 1.10: Possible event combinations on a given document in Sequence S_2

In this sequence, users can (i) click on a key frame in the result list (clicking result) and (ii) browse through its presented neighbored frames (browsing).

User Action Sequence S_3 (Click-View-Explore'n'Slide)

The third sequence S_3 covers an event combination which can be achieved when interacting with a document using the text-only video retrieval system provided by Browne et al. [14]. Their web interface ranks retrieved results in a list of relevant video programs. Each row displays the most relevant key frame, surrounded by its two neighbored key frames. Below the shots, the text associated with the result is presented. The query terms which are associated with the key frame are highlighted when the user moves the mouse over the key frame. When clicking on a key frame, the represented video shot can be played. Different from S_1 and S_2, this sequence considers two additional low-level events: highlighting metadata (exploring) and using a sliding bar (sliding). We refer to this scenario as "Click-View-Explore'n'Slide". Possible low-level event combinations are visualized in Figure 1.11.

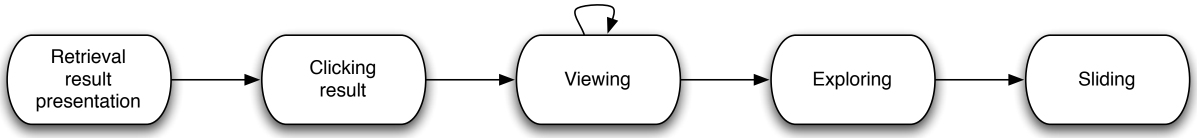


Figure 1.11: Possible event combinations on a given document in Sequence S_3

In this sequence, users can (i) click on a key frame (clicking result) to trigger (ii) video playback (viewing). They can (iii) highlight associated query terms (exploring) and (iv) navigate through the video using a sliding bar (sliding).

User Action Sequence S_4 (Preview-Click-View'n'Browse)

This sequence models the users' interaction on a given document using the system provided by Hopfgartner et al. [45]. In their interface, retrieved video shots, represented by a key frame, are listed in a result panel. Hovering the mouse over a key frame will highlight a tool tip showing its neighbored key frames and the associated text (previewing). When clicking on a key frame, the corresponding video is played (viewing). The video which is currently played is surrounded by its neighbored key frames. Users can click on them and browse through the current video (browsing). We refer to this sequence as "Preview-Click-View'n'Browse". Possible low-level event combinations are visualized in Figure 1.12.

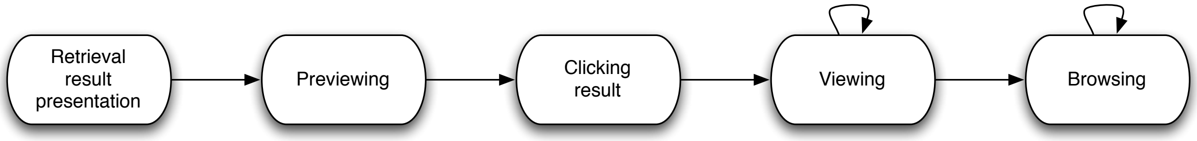


Figure 1.12: Possible event combinations on a given document in Sequence S_4

In this sequence, users can (i) highlight additional information in moving the mouse over a retrieved key frame to get some additional information of the shot (neighbored key frames and text from the speech recognition software) (previewing), (ii) click on a key frame of a result list (clicking result) and (iii) play a video (viewing). Also, they can (iv) browse through the video to find new results in the same video (browsing).

User Action Sequence S_5 (Click'n'More)

Sequence S_5 is the most complex of all introduced sequences. It is based on the retrieval interface by Christel and Conescu [20]. In this interface, retrieved results are represented by key frames and presented in a list. Clicking on one key frame, the user can choose to explicitly mark a shot as relevant (providing explicit relevance feedback), to play the video (viewing) or to display additional information (exploring). Further, it allows to browse displayed key frames (browsing) and slide through the video (sliding). We refer to this sequence as "Click'n'More". Possible low-level event combinations are visualized in Figure 1.13.

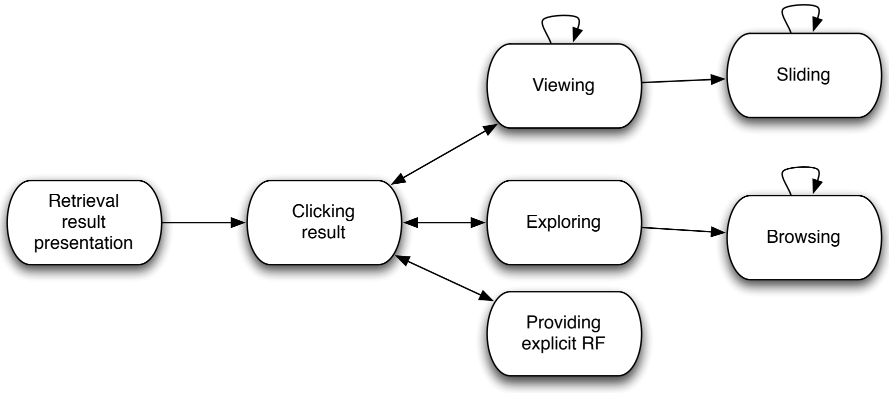


Figure 1.13: Possible event combinations on a given document in Sequence S_5

In this sequence, users can (i) click on a key frame in the result list (clicking result) and (ii) play a video (viewing). They can also (iii) use the sliding bar (sliding). Users may (iv) browse through the video to find new results in the same video (browsing). Moreover, they can (v) show additional video information and sort results by date and broadcasting station (exploring). Besides, they can explicitly judge the relevance of a video shot (providing explicit relevance feedback).

The introduced user interaction scenarios illustrate that the design of graphical video retrieval interfaces directly influence user behavior patterns. Even though users might follow the same aim, i.e., finding documents of interest, the interface design forces them to interact differently. Sequence S_2, for example, shows that users can interact with video results without viewing the actual video. In sequence S_3, however, viewing a video is essential while interacting with the results. Consequently, Hopfgartner and Jose [42] argue that the interpretation and importance of the implicit indicators of relevance depend on the interface context. By applying the above introduced scenarios, they simulate users providing implicit relevance feedback while performing a retrieval task over multiple iterations. They conclude that these features can be seen as implicit indicator of relevance and thus, retrieval results can be adapted to the user's current interest. Further applications of this implicit relevance feedback in the video domain have been presented in [43, 46].

1.4 Evaluation and Benchmarking

This section reviews the evaluation methods for browsing and interactive search, near duplicate detection as well as for event detection and linking video content by occurrence of objects, persons and places.

1.4.1 Browsing and Interactive Search

As pointed out in [111] evaluation of browsing and exploratory search tools is still an open issue. In the information retrieval and multimedia information retrieval community evaluations following the Cranfield paradigm, which is also used by the TREC (text) and TRECVID (video) [100] retrieval benchmarks, has been widely adopted. This type of evaluation is system or component centric and answers a well defined information need, i.e., question answering or fact finding. In browsing or exploratory search the user's information need may not be well-defined [112]. User centric evaluation methods such as surveys take the context of the user's task when using a system into account. The issue of limited correspondence between these evaluation methods has been discussed for information retrieval systems [105].

Most of the literature on evaluation of exploratory search deals with text documents. In the multimedia domain evaluation approaches for summarization and skimming systems often deal only with single multimedia documents, rather than with collections. The following classes of evaluation approaches have been proposed (of course combinations of the methods from different classes are sometimes used).

Survey The users are asked about their experience with the tool, their satisfaction with the results and the relevance of certain features of the tool (e.g., [89]). This type of evaluation does not require any ground truth or specific preparation of a data set.

Analysis of system logs This approach uses either server-side logs [1] or specific client applications that log user actions [56]. The main advantage is that evaluation does not interfere with the user's work with the system and that the approach can be used for long-term studies. However, comparison across different types of tasks and systems might be difficult.

Question answering Users are asked fact finding questions about the content in order to evaluate whether they have found the correct segment of content or were able to extract information from the collection of multimedia documents (e.g., [59]). The questions can be open or in the form of a multiple choice test (quiz). The correctness of answers to open questions needs to be checked by a human, while multiple choice tests can be very efficiently evaluated once the ground truth for a specific data set has been created.

Indirect evaluation The user performs a task using the tool or system. Based on the success of this task the effectiveness of the tool can be measured. The task can for example be a content retrieval task [9] or gathering information from a meeting archive browser [63]. Once ground truth for theses tasks has been created the answers can be checked automatically.

In [8] two TRECVID style fact-finding approaches, a retrieval and a question answering task, and a user survey were applied to the evaluation of a video browsing tool for use in movie and TV post-production. The retrieval task correlates better with the user experience according to the survey than the question answering tasks. As retrieving relevant content is also closer to the real-world application of the tool than finding facts about the content, it seems to be the more appropriate evaluation method in this case, although it is a costly method due to the efforts for data set and ground truth preparation. The authors conclude that surveys are rather suitable for comparing the general usability of tools for certain applications than for getting information about strengths and weaknesses of specific functionalities of a tool.

In 2005 and 2006 the TRECVID video retrieval benchmarking initiative [100] organized a pilot task for exploring rushes material. Some participants used interactive search and browsing approaches. For the pilot task no specific evaluation scheme was defined, and most participants evaluated specific feature detectors rather than the overall systems. In general the results show that the recall scores are lower than precision in such an application. The difficulty in evaluating such a task led TRECVID to move to video skimming of rushes video from 2007 onwards.

1.4.2 Near Duplicate Detection

Different measures for evaluating near duplicate video detection are found in literature. A recent study [6] found the following measures are applied for evaluating near duplicate video detection: three variants of precision/recall based measures, a measure based on normalized mutual information, and three measures coming from classic copy detection, i.e., two measures used in the MUSCLE VCD benchmark [64] and the one used in the TRECVID benchmark [100] content-based copy detection task. The measure differs in several dimensions, such as providing frame or segment precision, requiring the same segmentation for ground truth and results and support for evaluation of clustering per scene/topic. There are further methods used for evaluating near duplicate detection that require additional metadata (e.g., topic labels), such as the measure for evaluating news story threading in [33]. However, as they are not generally applicable this additional metadata are not available in all tasks that are relevant for near duplicate detection.

In the TRECVID 2007 and 2008 rushes summarization experiments, the redundancy of the summaries was judged by the evaluators, which is an indirect measure of how well removal of near duplicates works in the summary generation process. The evaluators used a five point scale. The results are in the range 2 to 4, with both mean and median 3.33 [81]. Other results for near duplicate detection are difficult to compare due to different task context, data sets and measures. On the TRECVID data set, equal precision/recall of around 0.7 [7] or precision of up to 0.85 at recall of 0.3 [32] are reported.

In [6] the correlations of the different measures on real algorithm results as well as on simulated results have been analyzed, as well as their correlation with human judgment of redundancy from the TRECVID rushes experiments. The results show that depending on the type of differences (segments added, dropped, shifted) between ground truth and results the correlation between the measures can in some cases be quite low, so that results obtained from different measures cannot be compared as it is sometimes found in literature. Shifting segment boundaries has the strongest impact on the correlation of the measures, and does not only affect frame based measures, but also shot based ones that assume alignment between result and truth segments. In general there is no grouping into frame and segment based measures. However, the different variants of precision/recall type measures have higher correlations among them than others. For obtaining comparable evaluation results, choosing one of the measures from the well correlated group seems to be advantageous. The aligned cluster and frame based precision/recall measures both do not require the same segmentation on ground truth and result set and support clustering. These are useful properties for many practical problems. The correlation of the measures with human perception is generally weak, but rank correlation is slightly higher. The frame based precision/recall measure performs best with a correlation coefficient of 0.37. Human perception seems to take a broader range of factors into account than covered by near duplicate detection methods.

1.4.3 Event Detection and Instance Linking

Related emerging topics are event detection in video and linking of video segments based on recurring objects, persons or places. In 2010, TRECVID included new tasks to address these emerging research topics. The multimedia event detection task aims at finding complex events in video. A normalized detection cost measure has been used, and the results were quite good, i.e., most runs were clearly below the baseline cost of reporting no events (and thus no false positives). However, some groups tried dynamic features for representing the events, that did not outperform static features. The other new task targets detecting the occurrence of the same object instance, person or location in a moderately large database. The instances differ strongly in terms of size, pose, lighting, etc. The results show that automatic systems perform very poorly, with top mean average precision around 0.030, and top median average precision around 0.005. However, the experiment also shows that interactive systems can reach a mean average precision of around 0.5 on the same data. This clearly underlines the potential of interactive search and browsing approaches.

A somewhat related task, i.e., determining the location of images, was part of the MediaEval 2010 [70] benchmark. In the top runs, nearly half of the images were placed in a one km radius of the target, however, most approaches relied only or mainly on tags provided with the images.

1.5 Conclusion and Outlook

In this chapter we have reviewed tools for browsing, navigating and organizing audiovisual content in TV post-production. We find a broad range of tools for this purpose, ranging from simple or extended video players, which are already widely used today, to much more advanced approaches, that are based on various automatic analysis methods and provide sophisticated user interfaces. Such tools are currently still research prototypes, partly because of the fact that the underlying content analysis methods are not robust enough for production use. However, such tools have the potential to change workflows and enhance productivity in TV post-production.

The graphical user interfaces of the tools directly influence user behavior patterns, as is illustrated by the user action sequences we have modeled based on representative video retrieval interfaces that have been surveyed in [98]. Each sequence models different user interaction scenarios where users trigger different events while interacting with a given document. Even though users might follow the same aim, i.e., finding documents of interests, the interface design forces them to interact differently. Sequence S_2 (Click'n'Browse), for example, shows that users can interact with video results *without* viewing the actual video. In sequence S_3 (Click-View-Explore'n'Slide), however, viewing a video is essential while interacting with the results.

The review of benchmarking methods and results highlights two important facts. First, fully automatic content based methods for organizing and cross-linking are limited in terms of their performance, and providing additional metadata to support the process is too costly. Benchmarking results show that finding reoccurring objects, persons and locations under general conditions, as well as identifying complex dynamic events in video are hard problems. Both are emerging research topics, and while we can expect significant progress in coming years, the results are still not sufficient to be directly used in production. However, benchmarking results also show that having the "human in the loop" can boost the results of such tasks. This means that interactive browsing and navigation tools, combining content-based methods with intelligent user interfaces, can already provide solutions to practical problems in TV post-production workflows.

The second conclusion from the state of the art in benchmarking is that evaluating interactive search and browsing tools still poses open research questions. The focus of evaluation is on automatic tasks, which is clearly a simpler problem, and tries to transfer these methods also to interactive tools. However, this neglects many important aspects of browsing and navigation tools.

Finally, there are new trends in TV production that also need to be reflected by future content management tools. One is 3D TV, which at present means stereoscopic, but that might change to a larger number of views or actual 3D information with future display technologies. The consequence for post-production is that even more content has to be handled, and that further properties of the content (e.g., the depth of objects) are relevant for selection, and also need to be visualized appropriately. Another trend is the integration of broadcast content with web and social media content, for example in Hybrid TV, i.e., delivering internet content together with the main audiovisual stream, or "Second Screen" approaches, i.e., delivering personalized content related to the broadcast to devices such as smart phones or tablets. While production tools for interactive TV, which have been thoroughly studied in the recent years, provide some relevant solutions, there are new challenges in terms of content management. For example, not all information is local at the production site, but might be accessed from anywhere on the internet, and the content to be delivered is potentially dynamically changing over time.

Acknowledgment

The first author's work has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 215475, "2020 3D Media – Spatial Sound and Vision" (http://www.20203dmedia.eu/). The third author was supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

References

[1] S. Fissaha Adafre and M. de Rijke. Exploratory search in Wikipedia. In *SIGIR Workshop on Evaluating Exploratory Search Systems*, 2006.

[2] Donald A. Adjeroh, M. C. Lee, and Irwin King. A distance measure for video sequences. *Comput. Vis. Image Underst.*, 75(1-2):25–45, 1999.

[3] MAGIX AG. Movie Edit Pro. `http://www.magix.com/us/movie-edit-pro/`, 2011. [Online; accessed March 24, 2011].

[4] Joaquim Arlandis, Paul Over, and Wessel Kraaij. Boundary error analysis and categorization in the trecvid news story segmentation task. In *CIVR'05: Proceedings of the 4th International Conference on Image and Video Retrieval, Singapore, July 20-22, 2005*, pages 103–112, 2005.

[5] F. Arman, R. Depommier, A. Hsu, and M.Y. Chiu. Content-based browsing of video sequences. *Proceedings of the Second ACM International Conference on Multimedia*, pages 97–103, 1994.

[6] Werner Bailer. Evaluating detection of near duplicate video segments. In *ACM International Conference on Image and Video Retrieval*, pages 197–204, Xian, CN, Jul. 2010.

[7] Werner Bailer, Felix Lee, and Georg Thallinger. Detecting and clustering multiple takes of one scene. In *Proceedings of 14th Multimedia Modeling Conference*, pages 80–89, 2008.

[8] Werner Bailer and Herwig Rehatschek. Comparing fact finding tasks and user survey for evaluating a video browsing tool. In *Proceedings of ACM Multimedia*, Beijing, CN, Oct. 2009.

[9] Werner Bailer, Christian Schober, and Georg Thallinger. Video content browsing based on iterative feature clustering for rushes exploitation. In *Proc. TRECVID Workshop*, pages 230–239, Gaithersburg, MD, USA, Nov. 2006.

[10] Werner Bailer, Wolfgang Weiss, Gert Kienast, Georg Thallinger, and Werner Haas. A video browsing tool for content management in post-production. *International Journal of Digital Multimedia Broadcasting*, Mar. 2010.

[11] 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.

[12] Marco Bertini, Alberto Del Bimbo, and Walter Nunziati. Video clip matching using MPEG-7 descriptors and edit distance. In *CIVR*, pages 133–142, 2006.

[13] Matthias Bezold. Describing user interactions in adaptive interactive systems. In *UMAP'09: Proceedings of the 17th International Conference on User Modeling, Adaptation, and Personalization, formerly UM and AH, Trento, Italy*, pages 150–161, 2009.

[14] Paul Browne, Csaba Czirjek, Georgina Gaughan, Cathal Gurrin, Gareth Jones, Hyowon Lee Sean Marlow, Kieran Mc Donald, Noel Murphy, Noel O'Connor, Neil O'Hare, Alan F. Smeaton, and Jiamin Ye. Dublin City University Video Track Experiments for TREC 2003. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD, USA, 2003.

[15] Lekha Chaisorn and Tat-Seng Chua. The segmentation and classification of story boundaries in news video. In *VDB'02: Proceedings of the IFIP TC2/WG2.6 Sixth Working Conference on Visual Database Systems, Brisbane, Australia*, pages 95–109, 2002.

[16] Linjun Chang, Yichen Yang, and Xian-Sheng Hua. Smart video player. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1605–1606, 23 2008-April 26 2008.

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

[18] Kai-Yin Cheng, Sheng-Jie Luo, Bing-Yu Chen, and Hao-Hua Chu. Smartplayer: user-centric video fast-forwarding. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 789–798, New York, NY, USA, 2009. ACM.

[19] Patrick Chiu, Andreas Girgensohn, Wolfgang Polak, Eleanor Rieffel, and Lynn Wilcox. A genetic algorithm for video segmentation and summarization. In *Proc. IEEE International Conference on Multimedia and Expo*, volume III, pages 1329–1332, New York, NY, USA, 2000.

[20] Michael G. Christel and Ronald M. Conescu. Addressing the challenge of visual information access from digital image and video libraries. In *JCDL'05: Proceedings of the ACM/IEEE Joint Conference on Digital Libraries, Denver, CA, USA, June 7-11, 2005*, pages 69–78, 2005.

[21] Michael G. Christel, Alexander G. Hauptmann, Adrienne S. Warmack, and Scott A. Crosby. Adjustable Filmstrips and Skims as Abstractions for a Digital Video Library. In *Proc. of the IEEE Forum on Research and Technology Advances in Digital Libraries*, pages 98–104, Baltimore, MD, USA, 1999.

[22] Tat-Seng Chua, Shih-Fu Chang, Lekha Chaisorn, and Winston H. Hsu. Story boundary detection in large broadcast news video archives: techniques, experience and trends. In *ACM MM'04: Proceedings of the 12th ACM International Conference on Multimedia, October 10-16, 2004, New York, NY, USA*, pages 656–659, 2004.

[23] Michele Covell, Shumeet Baluja, and Michael Fink. Advertisement detection and replacement using acoustic and visual repetition. In *IEEE Workshop on Multimedia Signal Processing*, pages 461–466, Oct. 2006.

[24] Chris Crockford and Harry Agius. An empirical investigation into user navigation of digital video using the VCR-like control set. *International Journal of Human-Computer Studies*, 64(4):340 – 355, 2006.

[25] O. de Rooij, C.G.M. 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.

[26] M. del Fabro, K. Schoeffmann, and L. Boeszoermenyi. Instant Video Browsing: A Tool for Fast Non-sequential Hierarchical Video Browsing. *HCI in Work and Learning, Life and Leisure*, pages 443–446, 2010.

[27] A. Divakaran, C. Forlines, T. Lanning, S. Shipman, and K. Wittenburg. Augmenting fast-forward and rewind for personal digital video recorders. In *IEEE International Conference on Consumer Electronics (ICCE), Digest of Technical Papers*, pages 43–44. Citeseer, 2005.

[28] 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.

[29] Alan Dix, Janet Finlay, and Russell Beale. Analysis of user behaviour as time series. In *HCI'92: Proceedings of the Conference on People and computers VII*, pages 429–444, New York, NY, USA, 1993. Cambridge University Press.

[30] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 237–246, New York, NY, USA, 2008. ACM.

[31] Pierre Dragicevic, Gonzalo Ramos, Jacobo Bibliowitcz, Derek Nowrouzezahrai, Ravin Balakrishnan, and Karan Singh. Video browsing by direct manipulation. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 237–246, New York, NY, USA, 2008. ACM.

[32] Emilie Dumont and Bernard Mérialdo. Rushes video parsing using video sequence alignment. In *CBMI 2009, 7th International Workshop on Content-Based Multimedia*

Indexing, June 3-5, 2009, Chania, Crete Island, Greece, Jun. 2009.

[33] Pinar Duygulu, Jia-Yu Pan, and David A. Forsyth. Towards auto-documentary: tracking the evolution of news stories. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 820–827, New York, NY, USA, 2004. ACM Press.

[34] David Eichmann and Dong-Jun Park. Boundary and feature extraction at the university of iowa. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD, USA, 2004.

[35] H. Eidenberger. A video browsing application based on visual MPEG-7 descriptors and self-organising maps. *International Journal of Fuzzy Systems*, 6(3):125–138, 2004.

[36] H. Goeau, J. Thievre, ML 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.

[37] A. Hampapur and R. M. Bolle. Comparison of distance measures for video copy detection. In *IEEE International Conference on Multimedia and Expo*, pages 737–740, Aug. 2001.

[38] A. Hampapur, K. Hyun, and R. M. Bolle. Comparison of sequence matching techniques for video copy detection. In M. M. Yeung, C.-S. Li, and R. W. Lienhart, editors, *Storage and Retrieval for Media Databases 2002*, volume 4676 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 194–201, Dec. 2001.

[39] Alexander G. Hauptmann and Michael J. Witbrock. Story segmentation and detection of commercials in broadcast news video. In *Proceedings of the Advances in Digital Libraries Conference*, 1998.

[40] M.A. Hearst, ST Dumais, E. Osman, J. Platt, and B. Scholkopf. Support vector machines. *Intelligent Systems and their Applications, IEEE*, 13(4):18–28, 1998.

[41] Daniel Heesch, Peter Howarth, J. Magalhães, Alexander May, Marcus Pickering, Alexei Yavlinski, and Stefan Rüger. Video Retrieval using Search and Browsing. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD, USA, 2004.

[42] Frank Hopfgartner and Joemon M. Jose. Evaluating the implicit feedback models for adaptive video retrieval. In *ACM MIR '07 - Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval, Augsburg, Germany*, pages 323–331. ACM Press, 09 2007.

[43] Frank Hopfgartner and Joemon M. Jose. Semantic user modelling for personal news video retrieval. In *Proceedings of Multimedia Modeling Conference*, pages 336–346, 2010.

[44] Frank Hopfgartner and Joemon M. Jose. Semantic user profiling techniques for personalised multimedia recommendation. *Multimedia Syst.*, 16(4-5):255–274, 2010.

[45] Frank Hopfgartner, Jana Urban, Robert Villa, and Joemon M. Jose. Simulated Testing of an Adaptive Multimedia Information Retrieval System. In *CBMI'07: Proceedings of the Fifth International Workshop on Content-Based Multimedia Indexing, Bordeaux, France*, pages 328–335. IEEE, 06 2007.

[46] Frank Hopfgartner, Thierry Urruty, Pablo Bermejo Lopez, Robert Villa, and Joemon M. Jose. Simulated evaluation of faceted browsing based on feature selection. *Multimedia Tools Appl.*, 47(3):631–662, 2010.

[47] Frank Hopfgartner, David Vallet, Martin Halvey, and Joemon M. Jose. Search trails using user feedback to improve video search. In *ACM Multimedia*, pages 339–348, 2008.

[48] W. Hsu, S.-F. Chang, C.-W. Huang, L. Kennedy, C.-Y. Lin, and G. Iyengar. Discovery and fusion of salient multimodal features toward news story segmentation. In *Proc. Storage and Retrieval Methods and Applications for Multimedia*, pages 244–258, 2004.

[49] Winston Hsu and Shih-Fu Chang. Topic tracking across broadcast news videos with visual duplicates and semantic concepts. In *International Conference on Image Pro-*

cessing (ICIP), Oct. 2006.

[50] W. Hürst, G. Goetz, and M. Welte. Interactive video browsing on mobile devices. In *Proceedings of the 15th international conference on Multimedia*, pages 247–256. ACM, 2007.

[51] W. Hürst and P. Jarvers. Interactive, dynamic video browsing with the zoomslider interface. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, page 4. IEEE, 2005.

[52] Wolfgang Hürst, Georg Götz, and Philipp Jarvers. Advanced user interfaces for dynamic video browsing. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA '04, pages 742–743, New York, NY, USA, 2004. ACM.

[53] Ichiro Ide, Hiroshi Mo, Norio Katayama, and Shin'ichi Satoh. Topic threading for structuring a large-scale news video archive. In *International Conference on Image and Video Retrieval*, pages 123–131, Jul. 2004.

[54] Apple Inc. Final Cut Studio. `http://www.apple.com/at/finalcutstudio/`, 2011. [Online; accessed March 24, 2011].

[55] Avid Technology Inc. Avid DS System. `http://www.avid.com/US/products/dssystem`, 2011. [Online; accessed March 24, 2011].

[56] B. J. Jansen, R. Ramadoss, M. Zhang, and N. Zang. Wrapper: An application for evaluating exploratory searching outside of the lab. In *SIGIR Workshop on Evaluating Exploratory Search Systems*, 2006.

[57] 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.

[58] Y.G. Jiang, J. Yang, C.W. Ngo, and A.G. Hauptmann. Representations of keypoint-based semantic concept detection: A comprehensive study. *Multimedia, IEEE Transactions on*, 12(1):42–53, 2010.

[59] V.B. Jijkoun and M. de Rijke. A pilot for evaluating exploratory question answering. In *SIGIR Workshop on Evaluating Exploratory Search Systems*, 2006.

[60] John R. Kender and Milind R. Naphade. Visual concepts for news story tracking: Analyzing and exploiting the nist trecvid video annotation experiment. In *CVPR (1)*, pages 1174–1181, 2005.

[61] Jim Kleban, Anindya Sarkar, Emily Moxley, Stephen Mangiat, Swapna Joshi, Thomas Kuo, and B. S. Manjunath. Feature fusion and redundancy pruning for rush video summarization. In *TVS '07: Proceedings of the international workshop on TRECVID video summarization*, pages 84–88, New York, NY, USA, 2007. ACM.

[62] A. Komlodi and G. Marchionini. Key frame preview techniques for video browsing. *Proceedings of the 3rd ACM Conference on Digital Libraries*, pages 118 – 125, 1998.

[63] W. Kraaij and W. Post. Task based evaluation of exploratory search systems. In *SIGIR Workshop on Evaluating Exploratory Search Systems*, 2006.

[64] J. Law-To, A. Joly, and N. Boujemaa. Muscle-VCD-2007: a live benchmark for video copy detection, 2007. http://www-rocq.inria.fr/imedia/civr-bench/.

[65] R. Lienhart, S. Pfeiffer, and W. Effelsberg. Video Abstracting. *Communications of the ACM*, 40(12):54–63, December 1997.

[66] J. Liu, Y. He, and M. Peng. NewsBR: a content-based news video browsing and retrieval system. In *Computer and Information Technology, 2004. CIT'04. The Fourth International Conference on*, pages 857–862, 2004.

[67] D.G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[68] Chengyuan Ma, Byungki Byun, Ilseo Kim, and Chin-Hui Lee. A detection-based approach to broadcast news video story segmentation. In *International Conference on Acoustics, Speech and Signal Processing*, pages 1957 –1960, 2009.

[69] K. Manske. Video browsing using 3D video content trees. In *Proceedings of the 1998*

workshop on New paradigms in information visualization and manipulation, pages 20–24. ACM, 1998.

[70] Mediaeval benchmarking initiative. `http://www.multimediaeval.org`. [Online; accessed Dec. 15, 2010].

[71] A. Messina, R. Borgotallo, G. Dimino, D. Airola Gnota, and L. Boch. Ants: A complete system for automatic news programme annotation based on multimodal analysis. In *Workshop on Image Analysis for Multimedia Interactive Services*, pages 219 –222, 2008.

[72] A. Messina and M. Montagnuolo. Fuzzy mining of multimedia genre applied to television archives. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 117–120. IEEE, 2008.

[73] Sean P. Meyn and Richard L. Tweedie. *Markov Chains and Stochastic Stability (Communications and Control Engineering)*. Springer Verlag, 1996.

[74] Hemant Misra, Frank Hopfgartner, Anuj Goyal, P. Punitha, and Joemon Jose. Tv news story segmentation based on semantic coherence and content similarity. In *Advances in Multimedia Modeling*, pages 347–357. Springer Berlin / Heidelberg, 2010.

[75] 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.

[76] C. S. Myers and L. R. Rabiner. A comparative study of several dynamic time-warping algorithms for connected word recognition. *The Bell System Technical Journal*, 60(7):1389–1409, Sept. 1981.

[77] Shi-Yong Neo, Huanbo Luan, Yantao Zheng, Hai-Kiat Goh, and Tat-Seng Chua. Visiongo: bridging users and multimedia video retrieval. In *CIVR '08: Proceedings of the 2008 international conference on Content-based image and video retrieval*, pages 559–560, New York, NY, USA, 2008. ACM.

[78] Chong-Wah Ngo, Wan-Lei Zhao, and Yu-Gang Jiang. Fast tracking of near-duplicate keyframes in broadcast domain with transitivity propagation. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 845–854, New York, NY, USA, 2006. ACM.

[79] Noel E. O'Connor, Csaba Czirjek, Seán Deasy, Noel Murphy, Seán Marlow, and Alan F. Smeaton. News story segmentation in the fischlar video indexing system. In *Proc. International Conference on Image Processing*, pages 418–421, 2001.

[80] JungHwan Oh and K.A. Hua. An efficient technique for summarizing videos using visual contents. In *IEEE International Conference on Multimedia and Expo (ICME)*, pages 1167–1170, New York, NY, USA, 2000.

[81] Paul Over, Alan F. Smeaton, and George Awad. The TRECVID 2008 BBC rushes summarization evaluation. In *Proceedings of the 2nd ACM TRECVid Video Summarization Workshop*, TVS '08, pages 1–20, 2008.

[82] KA Peker and A. Divakaran. Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure. In *Multimedia and Expo, 2004. ICME'04. 2004 IEEE International Conference on*, volume 3, 2004.

[83] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg. Abstracting Digital Movies Automatically. *Journal of Visual Communication and Image Representation*, 7(4):345–353, December 1996.

[84] Marcus J. Pickering, Lawrence W. C. Wong, and Stefan M. Rüger. Anses: Summarisation of news video. In *Proc. of International Conference on Image and Video Retrieval*, pages 425–434, 2003.

[85] D. Ponceleon. Hierachical brushing in a collection of video data. In *HICSS '01: Proceedings of the 34th Annual Hawaii International Conference on System Sciences (HICSS-34)*, volume 4, page 4045, Washington, DC, USA, 2001. IEEE Computer Society.

[86] S. Pongnumkul, J. Wang, G. Ramos, and M. Cohen. Content-aware dynamic timeline for video browsing. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, pages 139–142. ACM, 2010.

[87] Suporn Pongnumkul, Jue Wang, Gonzalo Ramos, and Michael Cohen. Content-aware dynamic timeline for video browsing. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 139–142, New York, NY, USA, 2010. ACM.

[88] Gert-Jan Poulisse and Marie-Francine Moens. Multimodal news story segmentation. In *Proceedings of the First International Conference on Intelligent Human Computer Interaction*, pages 95–101. 2009.

[89] Yan Qu and George W. Furnas. Model-driven formative evaluation of exploratory search: A study under a sensemaking framework. *Inf. Process. Manage.*, 44(2):534–555, 2008.

[90] G. M. Quénot, D. Mararu, S. Ayache, M. Charhad, , and L. Besacier. Clips-lis-lsr-labri experiments at trecvid 2004. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD, USA, 2004.

[91] H. Rehatschek, W. Bailer, H. Neuschmied, S. Ober, and H. Bischof. A tool supporting annotation and analysis of videos. *S. Knauss/A.D. Ornella(eds.), Reconfigurations. Interdisciplinary Perspectives on Religion in a Post-Secular Society*, pages 253–268, 2007.

[92] Andrew Rosenberg and Julia Hirschberg. Story segmentation of brodcast news in english, mandarin and arabic. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 125–128, 2006.

[93] E. Rossi, S. Benini, R. Leonardi, B. Mansencal, and J. Benois-Pineau. Clustering of scene repeats for essential rushes preview. *Image Analysis for Multimedia Interactive Services, International Workshop on*, 0:234–237, 2009.

[94] Shin'ichi Satoh, Masao Takimoto, and Jun Adachi. Scene duplicate detection from videos based on trajectories of feature points. In *MIR '07: Proceedings of the international workshop on multimedia information retrieval*, pages 237–244, New York, NY, USA, 2007. ACM.

[95] Klaus Schoeffmann. Facilitating interactive search and navigation in videos. In *Proceedings of the ACM International Conference on Multimedia*, Firenze, Italy, October 2010.

[96] Klaus Schoeffmann and Laszlo Boeszoermenyi. Video browsing using interactive navigation summaries. In *Proceedings of the 7th International Workshop on Content-Based Multimedia Indexing*, Chania, Crete, June 2009. IEEE.

[97] Klaus Schoeffmann and Laszlo Boeszoermenyi. Enhancing seeker-bars of video players with dominant color rivers. In Yi-Ping Phoebe Chen, Zili Zhang, Susanne Boll, Qi Tian, and Lei Zhang, editors, *Advances in Multimedia Modeling*, page , Chongqing, China, January 2010. Springer.

[98] Klaus Schoeffmann, Frank Hopfgartner, Oge Marques, Laszlo Boeszoermenyi, and Joemon M. Jose. Video browsing interfaces and applications: a review. *SPIE Reviews*, 1(1):018004–1–018004–35, 2010.

[99] Klaus Schoeffmann, Mario Taschwer, and Laszlo Boeszoermenyi. Video browsing using motion visualization. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, New York, USA, July 2009. IEEE.

[100] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and TRECVid. In *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, pages 321–330, New York, NY, USA, 2006. ACM Press.

[101] Masaru Sugano, Keiichiro Hoashi, Kazunori Matsumoto, and Yasuyuki Nakajima. Shot boundary determination on mpeg compressed domain and story segmentation experiments for trecvid 2004. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD,

USA, 2004.

[102] Yap-Peng Tan, Sanjeev R. Kulkarni, and Peter J. Ramadge. A framework for measuring video similarity and its application to video query by example. In *Proceedings of International Conference on Image Processing*, volume 2, pages 106–110, Kobe, JP, Oct. 1999.

[103] X. Tang, X. Gao, and C.Y. Wong. NewsEye: a news video browsing and retrieval system. In *Intelligent Multimedia, Video and Speech Processing, 2001. Proceedings of 2001 International Symposium on*, pages 150–153, 2001.

[104] Ba Tu Truong and Svetha Venkatesh. Video abstraction: A systematic review and classification. *ACM Trans. Multimedia Comput. Commun. Appl.*, 3(1):3, 2007.

[105] Andrew H. Turpin and William Hersh. Why batch and user evaluations do not give the same results. In *Proc. ACM SIGIR*, 2001.

[106] S. Vakkalanka, S. Palanivel, and B. Yegnanarayana. NVIBRS-news video indexing, browsing and retrieval system. In *Intelligent Sensing and Information Processing, 2005. Proceedings of 2005 International Conference on*, pages 181–186, 2005.

[107] Alessandro Vinciarelli and Sarah Favre. Broadcast news story segmentation using social network analysis and hidden markov models. In *Proceedings of the 15th international conference on Multimedia*, pages 261–264, 2007.

[108] Autonomy virage. `http://www.virage.com/`. [Online; accessed Mar. 21, 2011].

[109] T. Volkmer, S. M. M. Tahahoghi, and H. E. Williams. Rmit university at trecvid 2004. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD, USA, 2004.

[110] J. Wang, C. Xu, and E. Chng. Automatic sports video genre classification using pseudo-2D-HMM. *Pattern Recognition*, 4:778–781, 2006.

[111] Ryen W. White, Bill Kules, Steven M. Drucker, and m.c. schraefel. Supporting exploratory search. *Commun. ACM*, 49(4):36–39, 2006.

[112] Ryen W. White, Gary Marchionini, and Gheorghe Muresan. Editorial: Evaluating exploratory search systems. *Inf. Process. Manage.*, 44(2):433–436, 2008.

[113] Xiao Wu, Chong-Wah Ngo, and Qing Li. Threading and autodocumenting news videos: a promising solution to rapidly browse news topics. *Signal Processing Magazine, IEEE*, 23(2):59 –68, Mar. 2006.

[114] Ziyou Xiong, Regunathan Radhakrishnan, Ajay Divakaran, Yong Rui, and Thomas S. Huang. *A Unified Framework for Video Summarization, Browsing and Retrieval: with Applications to Consumer and Surveillance Video.* Academic Press, 2005.

[115] J. Yang, Y.G. Jiang, A.G. Hauptmann, and C.W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pages 197–206. ACM, 2007.

[116] Mei-Chen Yeh and Kwnag-Ting Cheng. Video copy detection by fast sequence matching. In *ACM International Conference on Image and Video Retrieval*, July 2009.

[117] Yun Zhai, Xiaochun Cao, Yunjun Zhang, Omar Javed, Alper Yilmaz, Fahd Rafi, Saad Ali, Orkun Alatas, Saad M. Khan, and Mubarak Shah. University of central florida at trecvid 2004. In *Proceedings of TRECVID Workshop*, Gaithersburg, MD, USA, 2004.

[118] Yun Zhai and Mubarak Shah. Tracking news stories across different sources. In *ACM Multimedia*, pages 2–10, 2005.

[119] Li Zhao, Wei Qi, Stan Z. Li, Shi-Qiang Yang, and H. J. Zhang. Key-frame extraction and shot retrieval using nearest feature line (NFL). In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, pages 217–220, New York, NY, USA, 2000. ACM.