Facilitating Interactive Search and Navigation in Videos

Klaus Schoeffmann
Klagenfurt University, Klagenfurt, Austria
ks@itec.uni-klu.ac.at

SUBMITTED to ACM MULTIMEDIA 2010 DEMO PROGRAM
WATCH VIDEO HERE: http://www-itec.uni-klu.ac.at/~klschoef/MM2010/

ABSTRACT
We present a tool that can efficiently facilitate interactive navigation and search in videos. In addition to browsing a video by shots it also allows a user to navigate through a video with extended seeker bars showing time-related content abstractions. Users having a rough knowledge about the content characteristics of scenes to be found can efficiently use these extended seeker bars to quickly find these scenes by interactive navigation. Moreover, users can easily perform similarity queries by utilizing content knowledge that has been gained during the browsing/navigation process. These queries can also be stored and reused for search in other videos having similar/same content. Furthermore, the tool can execute many queries at once and visualize the results as semantic events in a different seeker bar. Our tool provides a real alternative for situations where a user currently needs to employ a common video player for the task of search and navigation.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; D.2.8 [Software Engineering]: Metrics—complexity measures, performance measures

General Terms
Algorithms, Performance, Design, Experimentation

Keywords
Video Navigation, Video Browsing, Video Search, Video Exploration, Video Retrieval

1. INTRODUCTION
During the last decade many tools for video search have been proposed in the literature. These tools may be classified into three categories supporting different search paradigms:

(1) search by textual keyword, (2) search by visual example, and (3) search by concept.

Search by textual keyword allows the user to enter keywords and find video segments matching text which has either been manually annotated or extracted from media by automatic event analysis – like speech recognition or optical character recognition (OCR). This search paradigm has been popular in the early days of video retrieval and is still well accepted by users due to its simple and efficient use. However, automatic event analysis may miss many interesting events in a video, and the results of textual search may degrade in the presence of polysemy. Moreover, users are often not able to formulate a precise textual query according to their concrete imagination.

Search by visual example typically requires a user to give an example image or a visual sketch, which is analyzed by the system to find "similar" video segments according to some similarity measure in a certain feature space. The main problem with this search paradigm is that users rarely have example images of content they want to find. It is also a hard task for users to create a visual sketch of the scene of interest they have in mind.

Search by concept allows a user to select a specific semantic concept (e.g. Aircraft, Beach, Drawing, Outdoor, etc.) and browse through video segments related to this concept. A concept detector associates low-level features like color, texture, shape, and spatiotemporal correlations to concepts by applying machine learning algorithms (see [10]). This search paradigm seems to be convenient for users and has become very popular in recent years. Many researchers have started to build robust concept detectors for several hundreds or even thousands of concepts.

These paradigms are certainly well suited for many video retrieval scenarios, especially for shot-based search in large video archives. However, they are not feasible for efficient browsing and searching in recently recorded video due to the high runtime complexity of content analysis. It is not uncommon that analysis and classification of a one-hour video takes several hours or days, even on recent hardware. For a quick inspection of recently recorded media like hot news recordings, emergency videos, or surveillance videos, users currently employ common video players since they are immediately usable – although they provide only poor VCR-like navigation and no search function at all.

There are two other reasons why users often employ common video players for search and navigation in videos. First, these tools provide an intuitive user interface which is easy
to use for non-experts as well. For instance, the popular YouTube player still uses a simple seeker-bar for navigation. Secondly, for many search tasks user are not able to formulate concrete queries and rather prefer interactive search.

In [9] we have already described a video browsing tool that is designed for efficient navigation and low-complexity search in single videos. In this work we present a demonstration of an extended version of this tool.

The paper is organized as follows: In Section 2 we give a short overview of related work in the field of video browsing applications. Section 5 introduces the interface of our search tool. In Section 3 we describe the interaction model of extended seeker bars and show a few examples while Section 4 describes how users can perform similarity queries. In Section 6 we describe how users can employ content-based filtering in order to focus on shots potentially containing the searched content, according to the knowledge of the user. Finally, Section 7 concludes our paper.

2. RELATED WORK

Many video browsing applications have been proposed in the last years. In this paper we can only summarize a few recent ones, for a comprehensive review the interested reader is referred to [7].

Cheng et al. [3] proposed the SmartPlayer for browsing the content of a video. In addition to manually changing the playback speed it provides an automatic playback speed adaptation according to scene complexity, which is computed through motion complexity analysis for every shot. The 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.

Adams et al. [1] proposed temporal semantic compression for video browsing. They compute a tempo function for every frame and every shot, based on camera motion (e.g. pan and tilt), audio energy, and shot length and use it to improve shot-based navigation. Moreover, their browser enables a user to individually select a “compression rate” in order to shorten (i.e. summarize) the video according to several different compression modes. While linear compression simply speeds up playback, mid-shot constant takes a constant amount from the middle of a shot at constant playback rate. Pace proportional uses variable playback rate based on the frame-level tempo and interesting shots discard shots with low tempo values according to the selected compression rate.

Rooij et al. [4] introduced the concept of video threads, where a video thread is a sequence of feature-based similar shots from several videos in some specific order. A visualization of video threads for the purpose of explorative browsing has been developed as a CrossBrowser and a RotorBrowser.

Goeau et al. [5] proposed the so called Table of Video Contents (TOVC) visualization for browsing story-based video content like news, interviews, or sports summaries.

Hauptmann et al. [6] proposed Rapid Serial Visual Presentation (RSVP) (of keyframes) of query results in order to exploit both maximal use of human perception skills and the system’s ability to learn from human interaction.

Campanella et al. [2] proposed a visualization of MPEG-7 low-level features in a 2D feature space where each axis corresponds to a specific feature, selected by the user. Each shot is represented by a small square showing the dominant color of the shot. The shots are also visualized in a temporal manner by (1) color bars showing the dominant color of the shot and (2) keyframes of a shot.

3. EXTENDED SEEKER BARS

To facilitate interactive navigation in a video the tool provides several different seeker bars that show content abstractions in the background. Each of these seeker bars consists of two components (see Fig. 1):

- The overview diagram shows temporally compressed content information for the whole video.
- The detailed diagram shows content information in a higher level of detail according to a zoom-window defined by the user in the overview diagram. The user can increase or decrease the size of the zoom window in order to adjust the level of detail or the user can move it to another location in order to change the segment for which the visualization is shown.

![Figure 1: Interaction model of extended seeker bars](image)

A user is expected to choose the most appropriate seeker bar for a specific search task. For instance, if the user has some knowledge about the color structure of a searched scene, the Dominant Colors seeker bar can help to find the proper scene. This type of seeker bar visualizes the five dominating colors for every frame in the video in temporal manner.

If the user knows what motion characteristics a searched scene contains, the Motion Layout seeker bar could be more appropriate. It visualizes the temporal flow of several different motion directions with the corresponding motion intensity by colors of the HSV color space. More information about this visualization can be found in [8].

Please note that during the interaction process with the tool a user typically gains knowledge about the content structure of the video being browsed. However, if no knowledge about the searched scene is available, or the user has only a rough imagination in mind (e.g. of a specific person), a seeker bar like Preview Pictures, Frame Stripes or Stripe Flows would be a better choice for starting the search process.

The Preview Pictures seeker bar shows thumbnails of frames linearly sampled in from the entire video according to the selected level of detail. In a similar way, the Frame Stripes seeker bar visualizes only the center column of every frame. The Stripe Flows seeker bar is similar to Frame Stripes but increased the taken column (in a ring-buffer fashion) for every sample. This results in a visualization that is somehow similar to thumbnails but shows camera motion as well.

The tool allows to display several seeker bars at a glance, for which the user can enable a synchronization of the zoom window which ensures that the detailed diagrams of all seeker
bars show the same time position. In Fig. 2 several examples of seeker bars are given.

In the accompanying video of this paper it is shown how these extended seeker bars can be used for more efficient navigation through videos.

4. SIMILARITY SEARCH

Let’s assume the user has already found a desired scene and wants to look for other similar scenes in the current video. For this situation our tool provides two different query features:

1. Select a specific segment in a seeker bar and query the content for other segments containing a similar temporal flow of the underlying content-based feature (e.g. flow of dominant colors or flow of motion characteristics). Figure 3 shows how such a query is selected in the Motion Layout seeker bar. This query would immediately bring out all segments with similar motion flows.

2. Select one or many rectangular regions of interest in the currently displayed frame of the video and query the content for other frames/segments containing a similar color layout in the selected regions. Figure 4 shows how such a query is selected for the right part of the current frame in a video of ‘Who Wants To Be a Millionaire’. This query would immediately bring out all frames/segments in this video where a candidate has won 20000 Euros. Please note that the user can take influence on the internal threshold that is used for similarity search by selection ‘how similar’ the return matches should be.

When the user starts a query the tool searches through the entire content of the video and returns similar matches as a ‘shot-list’ in a new tab page in the right part of the interface (see Fig. 4). This list of results can be used to jump to specific matches. Please note that for every search a new tab page is created, thus, a user can preserve the results of several queries. Moreover, a user can store a query with a semantic name and reuse it later (e.g. for a different recording of the same content type). Stored queries can also be used for the Events seeker bar, which shows the temporal locations of results together with the query name (see Fig. 2).

5. INTERFACE

Fig. 5 shows a screenshot of our tool. It basically consists of three areas. The largest area (top-left) is used to display the video content itself. Moreover, in this area the user can define regions of interest in order to perform similarity queries (not shown in the figure, see Section 4).

At the right part a control area, consisting of several tabs, is shown. Here the user can enable/disable extended seeker bars and see the results of queries.

At the bottom part the extended seeker bars, enabled by the user for the current search/navigation task, are displayed. In this example, two seeker bars have been selected: (1) Frame Stripes and (2) Motion Layout. The user has chosen to lock the zoom windows across all seeker bars. This results in a temporal synchronization of the zoom windows and, thus, allows to see different visualization at the same level of detail. Below the control area (right above the ‘seeker bars’) a preview image is displayed when the mouse moves over any seeker bar in horizontal direction. Furthermore, the Hue circle of the HSV color space is shown as a help to understand the colors in the Motion Layout seeker bar (e.g. blue is down, yellow is up, red is right-up, etc.)

6. SHOT FILTERING

For shot-based navigation an on-the-fly shot-boundary detection feature, which is based on Color Layout, can be used. Similar to the query features all detected shots are displayed
7. CONCLUSIONS

We have presented a novel tool that supports users at the task of interactive search and navigation in videos. For navigation it provides several different content abstractions that can be used as seeker bars. It enables to easily find similar scenes by different types of similarity queries. Moreover, it also provides shot-based navigation whereas shots can be filtered according to content-based constraints selected by the user. The tool can efficiently be used in situations where users currently employ common video players for content-based search.

8. REFERENCES


