Design of a Casual Video Authoring Interface based on Navigation Behaviour

by

Matthew Fong

B.A.Sc., University of British Columbia, 2011

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES
(Electrical and Computer Engineering)

The University Of British Columbia
(Vancouver)

September 2014

Abstract

We propose the use of a personal video navigation history, which records a user's viewing behaviour, as a basis for casual video editing and sharing. Our novel interaction supports users' navigation of previously-viewed intervals to construct new videos via simple playlists. The intervals in the history can be individually previewed and searched, filtered to identify frequently-viewed sections, and added to a playlist from which they can be refined and re-ordered to create new videos. Interval selection and playlist creation using a history-based interaction is compared to a more conventional filmstrip-based technique. We performed several user studies to evaluate the usability and performance of this method and found significant results indicating improvement in video interval search and selection.

Preface

All the implementation and experiements henceforth were conducted by myself. Concepts and design decisions were discussed among myself, Abir Al Hajri, Gregor Miller and Sidney Fels.

A version of Chapter 4 was published as Fong, M, Al-Hajri, A, Miller, G, Fels, S (2014) at Graphics Interface 2014.

Table of Contents

List of Tables

List of Figures

Acknowledgments

I'd like to thank my supervisor Sidney Fels for overseeing this work, as well as Gregor Miller and Abir Al Hajri, who worked with me on much of this work.

Of course my loving family, my mom, Rosamaria, my dad, Ken, as well as my sister, Heather, who encouraged me throughout my schooling years.

And of course, my loving girlfriend, Crystal Lim, always positive, and always encouraging.

Chapter 1

Introduction

Video, as we know it today, is a widespread and popular medium for news, entertainment, as well as instructional content. It has become popular as a form of entertainment, especially on the Internet. In casual videos, such as the amateur videos found on YouTube™, users watch over 6 billion hours of video everyday, and these hours are not limited to desktop computer users either, as mobile devices account for almost 40% of YouTube's traffic[1]. Casual video sharing and watching video on YouTube and other online video repositories has become commonplace, and watching user-generated-content has become popular. As user-generated-content is generally unfiltered, there is a lack of quality control resulting in longer videos with uninteresting segments. It has been shown, in studies like Vihavainen et al. [40] that content creators are unlikely to edit videos unless given significant incentive. Thus, there is a need for a method in which video watchers can create video to share with others.

The average duration of a video on YouTube is 4.4 minutes [2]. The majority of viewers on YouTube spend less than five minutes watching each video, and the amount of time spent on this activity can vary greatly. The level of commitment to this activity is low, and can be defined as a casual activity. A less casual form of video viewing would be watching feature length movies, where the commitment to

[1] http://www.youtube.com/yt/press/statistics.html

[2] http://www.comscore.com/Insights/Press-Releases/2014/2/comScore-Releases-January-2014-US-Online-Video-Rankings

watching the entire movie is much higher. The idea of casual video viewing can be extended to video creation. Casual video authoring is analogous to searching for video to share with others after having viewed a short video. The current methods for creating and sharing video clips is insufficient for the casual use case.

Video sharing in its current state requires users to search and remember the locations of interesting parts of video. To create video, users need to first find the approprate content. In the context of sharing already-seen video, the video viewing history is the logical place to begin. The use of history and footprints, specifically in Internet browsing sessions, has aided users in finding previously seen information. Online video repositories, for example, YouTube allow users to view their previously seen videos, allowing users to narrow down their search for a specific video. This history mechanism can be extended to support a within-video navigation, allowing users to search for video clips. Searching for video clips within videos can be a difficult task, and becomes more difficult with lengthier videos. The video watching history that we propose allows users to mark and save clips of video simply by re-watching them. These video clips can be filtered so that users can find which video clips they have watched more than once, as well as favourited as a more complete form of organization. Sharing these video clips is then accomplished by dragging clips across to the playlist. This method for authoring is focused towards casual video watchers and the process for creating video is easier than past methods for video authoring.

The processes involved in video production have evolved throughout the years as technologies incrementally improved. The first consumer grade video capture devices were large and difficult to carry around. Today, video cameras are small and ubiquitous enough to be integrated within other hand-held electronic devices, namely mobile phones. The inclusion of video capture in mobile phones has made it extremely convenient for users to record their own video. The basic technique involved in video capture, however, has not changed, whereas editing video has changed significantly. Since the move from analogue to digital formats, manipulating video has become simple enough for amateur video enthusiasts to create their own videos. Now, instead of working with physical filmstrips or video cassettes, editors maneuver virtual video clips on screen. This has made video editing far more appealing, inviting more advanced computer users to attempt to edit video with software such as Apple iMovie, Microsoft Windows Movie Maker and Adobe Premiere. However, as seen in [40], the process is not easy enough to convince users to edit their video. The distribution end of video has also shifted. Without the Internet, users were restricted to handing out physical copies. Under the influence of the Internet, video watchers can share video with each other simply by email or instant message.

Sharing video has become an activity that extends beyond video producers. In the rise of social media, computer users share video that they have watched with others. Unfortunately, many existing video players are oriented toward linear viewing of content and do not provide easy methods to seek around the video. This is an issue when a user wants to search for a specific part of a video to share. On YouTube, for example, the act of sharing a video is accomplished by sharing a URL. The player, however, does not aid in video search, and only provides a linear view on the timeline. Furthermore, as mentioned before, much of the video found on YouTube can be uninteresting, and it is more useful to only share parts of the video, and to do so, users must uncover a hidden context menu to "Copy video URL at current time", like in Figure 1.1. Let us take a look at a typical usage scenario.

Susan is browsing stand up comedy routines on YouTube, and comes across a very funny joke that made her laugh. Susan saves the video into a list of favourites. She does this three times for ten videos. A week later, Susan wants to revisit them and share them with friends. Realizing that her friends may not have the time to watch each entire comedy routine, Susan makes sure to tell them where they should start watching and when they should stop watching. For each video in her favourites list, Susan searches through the video for the three jokes by skipping through the video, finds the beginning and end of the segment and sends it to her friends via email.

This scenario illustrates the idea behind saving and sharing video on YouTube.

1. Watch videos on YouTube

2. Add videos to favourites playlist

3. Memorize temporal locations of interesting parts of video

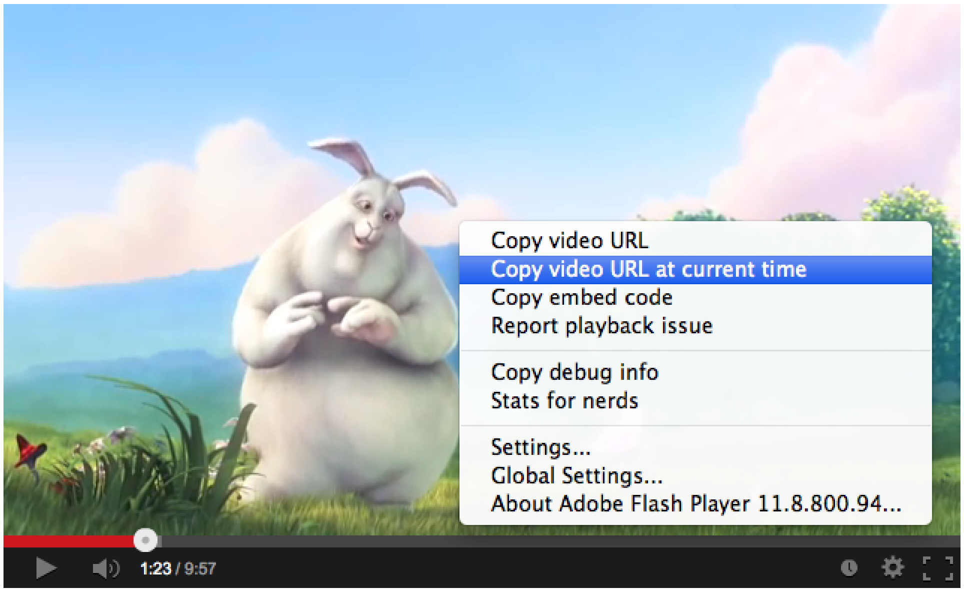


Figure 1.1: Context menu from a YouTube video.

4. Find favourites playlist, after a week, on YouTube

5. Search for temporal locations previously memorized

6. Right mouse click on the video and "Copy video URL at current time"

7. Paste link and send to friend, noting the amount of time to watch

8. Repeat previous 3 steps until finished

YouTube's playlist function is a useful way to save videos. However, sharing becomes difficult if the user wants to share multiple parts of different videos. Susan needed to search for the specific video intervals in each separate video, as well as note down the length of each interval. This requires Susan to rely on her memory both for remembering specifically which video it was in, as well as when the interval happened in that video.

Endel Tulving defines semantic memory as information that have been passed through as knowledge, and episodic memory is defined as information that is defined through personal experiences [39]. We will define our own terms in the video space. Semantic video memory defines the temporal location of a specific space in video, such as its timestamp (Where in time was that joke?). Episodic video memory defines the context of the video (Did this joke happen after that joke?). This search relies on semantic video memory, if Susan is willing to commit to memory the time stamps for those videos. Or, if Susan is seeking around the video looking for the interval, the search relies upon episodic memory, which means that Susan is relying on context. Of course memory is unreliable, and given our scenario, the week between watching the videos and sharing them makes it difficult. Given that remembering sequences of events is easier than remembering time stamps, we would like to build on usage of episodic memory to aid in video search and retrieval for video interval selection and sharing. Introducing a video navigation history would allow users to search for previously seen video intervals. The user will then ask the question "When did I watch that part of the video", which makes searching for video a much more personal experience. Much like a web-history, video-navigation history is displayed temporally, allowing users to see the order in which they watched certain video intervals and allowing users to retrace their steps. This works even if users viewed the video out of order.

Creation of the history can be accomplished in two ways. We allow users to implicitly create a video history by recording when a user rewinds to re-watch an affective [37] interval of video. This takes advantage of human nature and its innate tendency for repetition, as seen in Zipf's Law [44] and Pareto's Principle [26], which we will cover in Chapter 2. We believe that due to this behaviour, allowing users to save video by re-visiting it will encourage such behaviour if it is not already present. For other situations, we allow users to manually select when to record video. This is similar to personal-video-recorders (PVRs).

Using this history, users can then save these intervals to a customized playlist for future manipulation, or leave them in the history to find later. The main idea is that video interval selection becomes much easier, only requiring users to watch the video. If we revisit the scenario, the steps can be reduced significantly.

Susan is browsing stand up comedy routines, and comes across a very funny joke that made her laugh. Susan rewatches the joke, automatically saving it into her history. She does this three times for ten videos. A week later, Susan wants to revisit them and share them with friends. She goes into her history and searches *for the 30 history items (three for each of the ten videos), all next to each other, and adds them all into a playlist.*

1. Watch video
2. Re-watch interesting intervals
3. Add interesting intervals from history into a playlist
4. Find playlist, after a week
5. Share playlist

We believe that by allowing users to save specific intervals in addition to specific videos provides an easier solution to playlist authoring and sharing. It reduces the search space significantly and allows for more flexible manipulation of video over an abstraction of text. The use of history should be simple and easy to understand, as well as useful in providing users with information that can aid in searching and sharing of video intervals. The research, is thus, making sure the authoring using a video navigation history is feasible.

1.1 Research Question

The purpose of this work is to apply the history to a video editing context, meanwhile keeping it simple enough that casual users can utilize it in their every day video viewing and sharing. To direct this work, we propose the question:

Does integrating video history, based on tracking what a person is watching, into a video player provide an effective means for authoring for sharing online video content?

We investigate methods for creating video clips from larger videos, how to integrate the history into the video viewer in a way that does not interfere with other timelines, as well as a method to present it to the user. As the history grows, there also needs to be methods for managing it for search, and we looked at filtering methods. Lastly, we look at a method for collecting video clips in a playlist format as a form of authoring for sharing.

1.2 Contributions

We focus our efforts on answering the research question on the re-visitation behaviour mentioned earlier. We developed a new video viewing and navigation interface designed to integrate both the semantic and episodic memory mechanisms for creating and searching intervals for authoring. We ran two pilot studies and two full studies to test the design and verify its effectiveness. Thus, the main contribution this work makes is to demonstrate that video history based on tracking what and how often a person watches intervals is an effective means for creating and sharing online video content.

Our first contribution was creating and validating the video-navigation-history to facilitate playlist authoring. Through a user study, we demonstrated the effectiveness of history-based navigation to save video intervals for authoring. Results showed that with history, finding previously viewed video clips is faster than using the traditional filmstrip technique to accomplish the same task. Qualitative results revealed that such a tool is also desirable and useful. This design and user study are covered in Chapter 4.

Our second contribution was investigating user preference and watching behaviour a personal-video-recording (PVR) like design to create video intervals in conjunction with the personal navigation history. Through user studies, we determined that this design was preferred for situation where predictable events are in the watched video compared to the history design that was preferred for situation where the viewer doesnt know what is coming up in the video. We illustrate how the two designs work seamlessly together to provide a homogenous history mechanism. The design and user study are covered in Chapter 5.

1.3 Publications

We have been able to publish portions of this work in several conferences. A paper we published at the Interact Conference on Human-Computer Interaction (2013) [2] provides an overview and evaluation of the usage of video viewing history for video navigation and video interval search. These tasks, particularly in long videos, can be very daunting for users, and allowing users search a subset of the longer video helps immensely. Searching the history becomes more difficult as the number of videos being watched becomes larger. A paper we published at the ACM CHI Conference on Human Factors in Computing Systems (2014) [3] provides two different visualizations to allow users to organize and sort their history. We compared the two visualizations for finding video intervals against the conventional YouTube style navigation, and found that they offered improved performance.

At the Graphics Interface Conference (2014), we published two papers. One of the papers [1] developed a variation of traditional video timelines (introduced as the 'Filmstrip' in subsequent chapters) called the View Count Record (VCR). This allowed users to quickly find the most viewed parts of a video. This included self views as well as crowd sourced views. The other paper [14] is the work covered in Chapter 4.

Chapter 2

Related Work

In this chapter, we first introduce some of the work that has already been completed in the realm of video. Our approach to using video navigation behaviour for casual video authoring and sharing relies on video browsing, video authoring, and user-action-history. The overview of video browsing looks at methods for navigating video, for example, the timeline, or the VCR controls found in conventional television players. We take design cues from these works to integrate into our own methods for video navigation. Video authoring encompasses the use of traditional video editors found in commercial applications, as well as several works by other researchers for creating video. Lastly, user-action-history shows us how we can use users' navigation behaviour as a technique for creating interesting video.

2.1 Video Browsing

A popular video player can be found in YouTube's embedded video player, pictured in Figure 2.1. In this player, there are two basic parts: the main video view and the navigation timeline. The navigation timeline allows the user to skip through the video non-linearly. In order to provide users with a preview of the video, YouTube employs work by Girgensohn et al. [15]. As the cursor moves over the timeline, the thumbnail changes its contents corresponding to the position of the mouse. This preview thumbnail provides the user with a good indication of where they can seek without actually having to seek. In our interface, we applied a flexible thumbnail

Figure 2.1: YouTube's video player shows the main video, and a timeline that includes a preview for users to know where they are seeking.

visualization of each history segment: initially, each segment is visualized using a thumbnail that indicates the beginning of that segment which then changes while the user navigates this segment.

One of the primary uses of a video player is to allow users to navigate the content. Navigating a video space, specifically for searching, can be mentally demanding and time consuming, especially in longer videos. There have been many interaction techniques proposed in the literature to alleviate this problem in order to quickly navigate and search video content. The standard navigation tools used in most video systems are the VCR-like controls. These controls provide the user with the ability to move video time against its natural progression (forwards or backwards), which in turn allows them to search faster than by just watching at regular speed. As this method only allows users to go sequentially throughout the video, it is unsuitable for the quick video search needed.

An improvement along the VCR controls is the chapter based system found in interactive DVDs. This menu-like system allows users to skip ahead to time points predetermined by the video creator. In most cases, chapters are represented by

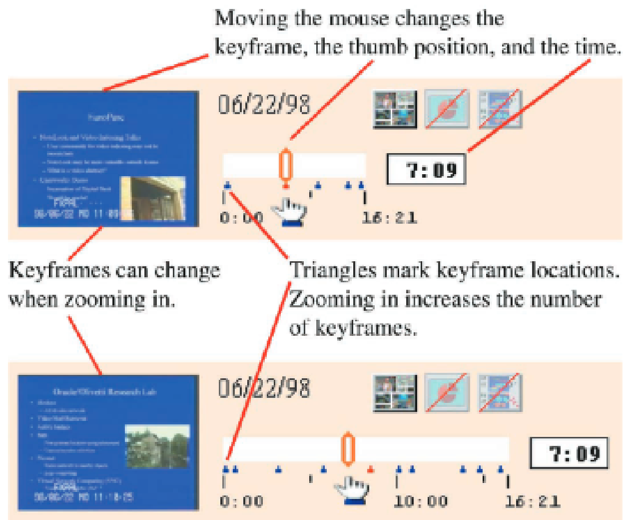


Figure 2.2: Dynamic thumbnail, by Girgensohn et al.

thumbnails representing the most important scene of the chapter. These thumbnails or previews, provide the users with a better understanding of the contents of the video, allowing users to preview the video. This system is used in conjunction with the VCR controls, and helps reduce the amount of time that a user has to search linearly. Due to its reliance on the VCR controls however, it still suffers the same problems. Furthermore, the chapter locations are placed at the discretion of the video creator, which may or may not be useful for the user.

An improvement to the chaptering system was developed by Li et al. [31]. They allow users to create their own annotations in the video, thus creating a customized table of contents. This allows users to bookmark interesting parts of the video and create a chapter system that makes sense for themselves. Self-created bookmarks inherently make more sense to the user and the temporal location of such bookmarks is easier to remember, making non-linear search easier. We make use of manual annotations in video in our interface due to this. The mechanism proposed in this work, however, intrudes on the video viewing experience and requires users to label and write down descriptions for each of the time. Furthermore, there is no visual representation of each of the bookmarks, and requires the user to seek to each bookmark to figure out the actual contents, should they be mislabeled.

An improvement to this would be to give thumbnail previews for each bookmark, like in the DVD chapter system described above.

Since video is a visual medium, providing visual previews is a very important part in enhancing video navigation. In the video systems developed by Christel et al. [9], and Drucker et al. [13], the use of thumbnail previews is heavily used to aid the user in search tasks. The system by Christel et al. uses thumbnails to display a storyboard of the video to aid in a search task. These thumbnails are keyframes extracted from the video automatically using text metadata already present in the video to separate clips from one another. This system relies heavily upon metadata to already be embedded within the video, and makes it an unlikely tool for searching in newly captured, or home made video. Drucker et al., in their SmartSkip video player, allows users to skip along regular 30 second intervals, which are previewed to the user via thumbnails along the timeline. This extends the functionality of the skip-ahead button, and allows users to look at where they are seeking. We adopt this interface element design, allowing users to preview the sections of the video without needing to interact with it.

Traditional timelines, such as the one shown in Figure 2.2 have inherent problems as the length of video becomes larger. Representation of each particular time-step becomes too small to see and use. Work by Hürst et al. [24, 25] introduces the ZoomSlider, which shows only part of the timeline, and allows users to shift the slider across the screen to seek across the video. The seeker bar is zoomable, and the granularity of seeking is dependent on the vertical mouse position, allowing for higher accuracy in seeking for longer videos. Commercially, this multi-level seeking functionality can be found in the Apple iOS default video player. The zooming functionality, however, is undesirable as it hides portions of the timelines, impeding access to the entire video. The ZoomSlider can be seen in Figure 2.3.

On the idea of showing more information of the video, Barbieri et al. [5], introduced the ColorBrowser, which extracted the dominant colour in each frame and presented it to the user in the form of a timeline. This can be seen in Figure 2.4. This is useful for showing scene changes in the video, and may be useful to the user in terms of search if the user remembers a scene to be predominantly a specific colour. In the context of a short video, however, this is not as useful.

Another interesting method of manipulating the timeline was introduced by

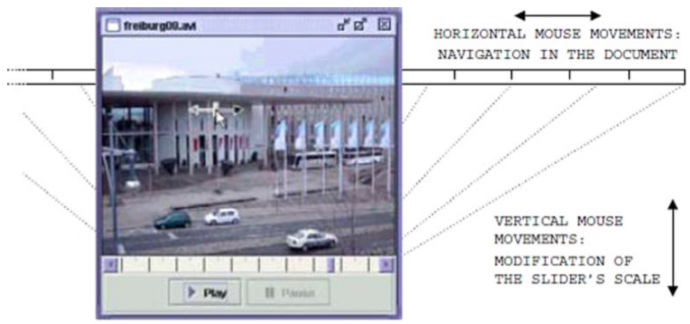


Figure 2.3: Dynamic thumbnail, by Hürst et al.

Figure 2.4: ColorBrowser by Barbieri et al.

Kimber et al. [28]. They developed a system that allowed users to use direct mouse manipulation (i.e., dragging) to manipulate objects directly within the scene along their natural flow. They achieved this by preprocessing video using computer vision techniques to enable object tracking. Using background/foreground segmentation they were able to extract objects, and allow them to be draggable. For example, users would be able to drag a moving car across the screen to control the forward and backward flow of time in the video. In an example application, they showed objects with motion trails shown to indicate the direction of movement to allowed for dragging. Dragivic et al. [12], Karrer et al. [27] used optical flow to accomplish the same task. This can be seen in Figure 2.5. Goldman et al. [17] set upon to improve upon the systems shown in the previous works, by providing support for partial occlusion in objects, motion grouping, and long-range accuracy of object tracking.

Divakaran et al. [11] proposed a method for quick video browsing by dynamically adjusting the playback rate for video. They used the compression information found in the video codec to evaluate motion. In places with low motion, playback rate was increased. In places with high motion, playback rate was decreased. Further work by Peker and Divakaran [35] introduced the same adaptive playback rate

Figure 2.5: Trailblazer by Kimber et al.

and included spatial-temporal complexity to the evaluation of scenes. High complexity resulted in lower playback rates, and lower complexity resulted in higher playback rates. Cheng et al. [8] extended upon the work by Peker and Divakaran, introducing SmartPlayer. They expand the previous works to include scene complexity, predefined scenes of interest, as well as user's preferences with respect to playback speeds. In addition, it will also learn the user's preferred event types and the preferred playback speeds specific to the event type through manual intervention from the user.

2.2 Using Video Watching History

The relatively high interest in the social web, sharing and the use of videos online has motivated researchers to investigate the use of video navigation history. Users leave footprints during the video-browsing process, and this can add value to the content for both analytical purposes, as well as future viewers. Syeda et al. [38] use data analysis to model video browsing behaviour. By collecting navigational behaviour, they can then deduce which parts of a video are more interesting. For example, they can look at how far a user watches a video and then stops watching, or if a user skims parts of a video first to find interesting parts. These portions of video are referred to as 'touched' video. They then use this data to assemble short previews by selecting clips based on video visit count. Using the model of video browsing behaviour, it is possible for the system to create previews on the video tailored to each particular user, whether they are "searching for something", or "found something", or "curious" etc. This authoring of videos based on user-navigation-history indicating user interest is a concept we adopted in allowing users to find video intervals that interest them.

Yu et al. [43] expanded on the work by Syeda et al., and introduces the notion of "ShotRank", which is a measure of the subjective interestingness and importance of a video shot. The ShotRank of a video's shot, or scene, is thus the probability that a viewer would visit a shot during browsing. ShotRank is computed based on link analysis between scenes in a video, as well as some low level feature detection in different scenes. The link analysis is generated using an "Interest-guided Walk", which takes a viewer's browsing behaviour to create a map linking different scenes of a video together. Furthermore, using the feature detection, ShotRank can then deduce links from scene to scene based on a variety of factors: they look similar, they sound similar, they are temporally sequential, etc. Using ShotRank, Yu et al. created a system that allowed users to skim through a video through a "chapter" based system determined by the scenes that ShotRank determined to be interesting. Unfortunately, ShotRank requires a priming phase, requiring multiple users to go through the video itself to create the data necessary to link scenes together. Mertens et al. [33] created a similar system for web lectures. It kept track of user footprints across the entire lecture and displayed it to users to show areas of interest. They further allow users to create bookmarks, display it to the users, as shown in Figure 2.6.

Video navigation history can play an important role in user-based information retrieval from videos on the web. Shamma et al. [37] and Yew et al. [42] have proposed a shift from content-based techniques to user-based analysis because it provides a more promising basis for indexing media content in ways that satisfy user needs. Leftheriotis et al. [30] developed a system called VideoSkip that records user video browsing actions (play, pause, skip). Using this data, they can then gen-

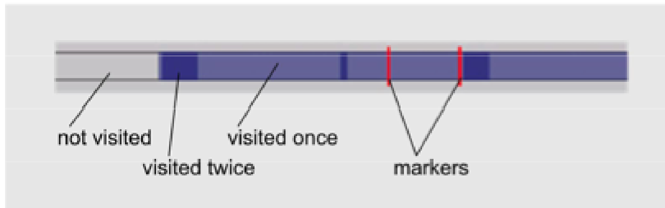


Figure 2.6: User footprints bar by Mertens et al.

erate thumbnails to represent each video based on the frame that has been mostly viewed by users. Gkonela et al. [16] also indicates that this simple user heuristic while navigating videos can be as effectively used to detect video-events as when content-based techniques are applied. Providing users with explicit access to their video navigation history gives users the flexibility to see what interests them, making it easier to select video intervals in short authoring tasks.

2.3 Video Authoring

In order to have video to watch, someone needs to create the video and make it suitable for general consumption. Video authoring is a difficult task and requires users to perform many search tasks to look for video intervals they want to share. They then have to ensure that each interval is correctly cut, as well as arranged in the proper order. The difficulty of video authoring can then be separated by the amount of interaction required to create the video. The most popular video authoring methods require users to perform every task in the process, such as importing videos, cutting them into usable clips, arranging them clips in a timeline, and selecting video transitions. Performing these tasks professionally requires skill and insight that the average consumer may not have. There are several tools for the job: software like Adobe Premiere [1] is difficult to learn, but allows for professional, full-featured video and audio mixing; whereas Apple's iMovie[2] and Microsoft's Windows Movie Maker[3] provide a simpler interface but have limited functionality.

[1] http://www.adobe.com/products/premiere.html
[2] http://www.apple.com/ilife/imovie
[3] http://windows.microsoft.com/en-ca/windows-live/movie-maker

As these tools are offered to users free of charge, they are an attractive solution to those seeking a video authoring tool. However, these types of tools may be excessive and require too much effort for a novice computer user who mainly just wants to share short video clips with their friends, as seen in studies by Vihavainen et al. [40], where users are not inclined to launch a video editor.

On the opposite end of the spectrum is the fully-automatic approach. These types of systems take only video as input from the user, making video creation extremely easy for users. There are commercial applications that implement this approach, such as Muvee autoProducer[4], Magisto[5]. The manual tool Pinncale Studio[6] also includes a "smart movie" function to generate video summaries from existing videos. Christel et al. [10] created a system that analyzes a video streams to produce a professional-quality summary video. To do this, they first need to identify "important" audio and video info. For video, they first decompose the video into different shots, and then going through each shot, they attempt to identify important objects, such as faces and text, and identify movement. For the audio portion, they run it under speech recognition, and align it to the video's transcript. They then analyze the video transcript and score the phrases based on frequency. Using all this information, they can extract the appropriate clips and assemble them into a video summary. This process requires a lot of computation to be performed to produce the video, with much of the information being generally unavailable in regular video.

Pfieffer et al. [36] also developed a similar system. They define scene cuts through the use of video and audio analysis. To then extract interesting scenes, they evaluate each based upon contrast, motion, colour composition, and dialog. Ranking upon each of these criteria allows the system to pick and choose a number of scenes, specified by the users, as well as the length of the scenes to be included in the resulting "trailer" video, also specified by the user. Using these systems to create video removes a lot of the freedom in creating video, and may not be suitable for just quick and easy sharing of video. Furthermore, make it very difficult to quickly create shareable video clips. Automatic tools, while not satisfactory

[4]http://www.muvee.com/
[5]http://www.magisto.com/
[6]http://www.pinnaclesys.com/publicsite/us/home/

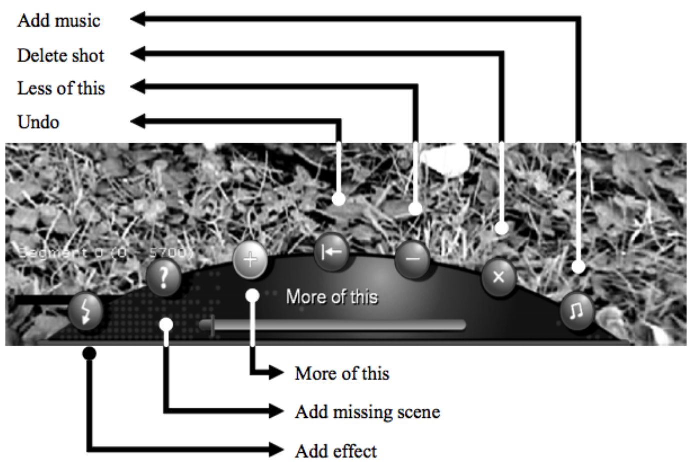


Figure 2.7: Edit While Watching by Weda et al.

yet, provide techniques that can be integrated alongside the history approach to provide further insight on the use of filters to help users find video intervals they find affective so that they can author and share videos.

As automatic solutions do not suit the criteria, we take a look at semi-automatic video creation tools. Weda et al. [41] produced a system that features “Edit While Watching”, providing users with an easy home video editing solution in the living room. Using a remote control to interact with their television, the user is then allowed to perform several video editing related tasks, such as adding music, deleting shots, adding scenes, adding effects, and altering suggested scenes.

They extract low-level features from raw video data such as camera motion, contrast and luminosity, and using this data, they can split video data into shots of about 1 to 10 seconds. These segments are then assigned scores based on visual quality from the features. The interface for these options can be seen in Figure 2.7. The purpose of this system then, is to provide users with a video editing system that is able to give users some control over the video editing process, but hide the details and reduce the skill required to take on full video editing programs.

Another content-aware system, LazyMedia, by Hua et al. [22], uses media content analysis and composition templates to allow users to create videos. These composition templates guide the user as to what kind of shot needs to go into which spot to create a visually pleasing video. The media content analysis features many video content filters, such as motion, colour, face detection, scene grouping, and attention detection for video. For audio, they only analyze music, for which onset, beat and tempo detection are supported.

Cattelan et al. [7] developed a Watch and Comment (WaC) system that would allow novices users to created interactive video just by watching video. They do this by associating user navigation commands with video editing commands and tagging users' comments with the video. As a user is watching the video, they can "edit" the video as they are watching it, inserting one of three commands that the interactive video produced will perform: seek, skip, or loop. The interactive video, or the resulting video will then follow a script based and perform almost like an edited video.

These works provide us with techniques in which we can help users author video intervals that can be used for sharing. These semi-automatic systems show that automatic analysis of video can provide satisfactory aid. With that in mind, highlighting affective video intervals using video-navigation history is a possibility that has yet to be explored.

2.4 User Action History

Humans are repetitive in nature and many of the tasks that we perform are likely to be performed again in the future. As seen in work by Greenberg et al.[18], people reuse commands in command line interface frequently and they developed a system to facilitate repeated usage. Further examples of repetition in human nature can be seen in natural language, as shown by Zipf et al.[44], as well as in economics, by the Pareto's Principle [26]. Zipf's law states that in natural language, the frequency of any word is inversely proportional to its rank in a frequency table. Naturally, some words occur more frequently than others, illustrating repetition in natural language. Pareto's Principle states that 80% of a business's revenues come from 20% of the business's clients. In other words, business's thrive on repeat customers. We investigate into the use of user histories and footprints for inspiration on how to use implicit tagging in video viewing for video authoring purposes.

We can start by looking into the undo mechanism found in almost every com-

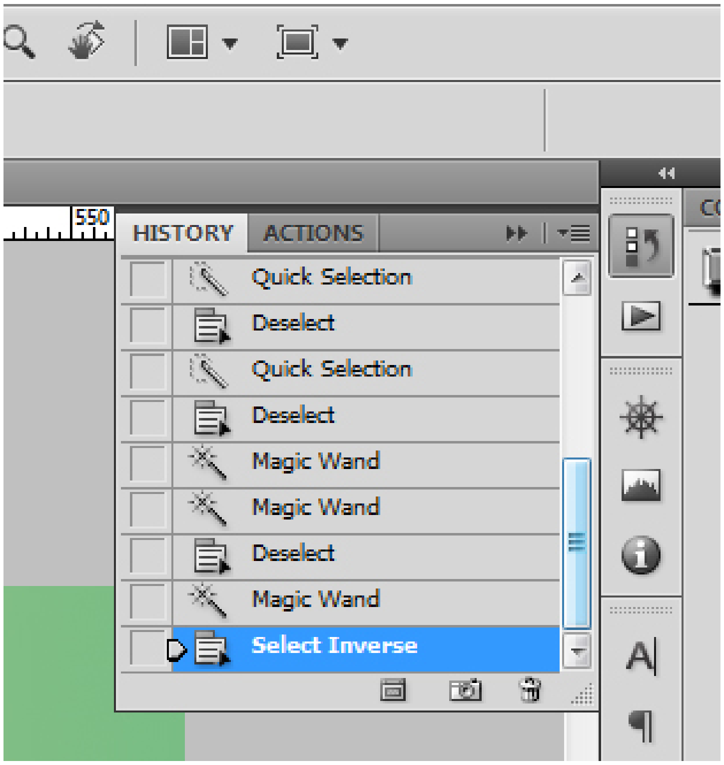


Figure 2.8: History found in Adobe Photoshop. The most recent action is on the bottom.

puter program. Many implementations allow users to explore the historical changes made to a document. A good example of this can be found in Adobe Photoshop's history, which can be seen in Figure 2.8. It is arranged in chronological order and allows users to view changes to their document every step of the way. Unfortunately, the visualization of the undoable actions does not provide much useful information about how the document has changed. It does, however, provide users with cues to aid in episodic memory retrieval, allowing users to look back and recognize the workflow in case they need to leave the task for a period of time.

Looking further back, Kurlander and Feiner [29] developed a graphical representation for user interaction history and macro creation. Users can use history to easily create macros. By giving users a graphical representation, it also allows users to easily see the macro created and easily deduce the procedure of the macro.

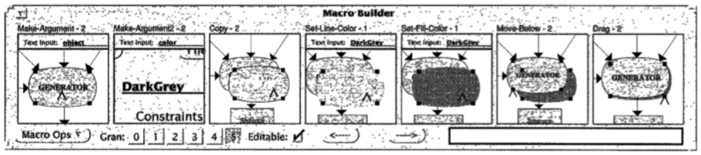


Figure 2.9: Chimera by Kurlander and Feiner. Shown is a macro to draw a diagram.

The system they created, Chimera, allows users to manipulate each stage of the macro, as well as pick and choose specific operations to create new macros. In Figure 2.9, we see a macro being created to draw one of the diagrams included in the original paper. They use a comic strip metaphor to indicate sequence.

To see better visualizations of history, we can look toward Grossman et al. [19], who introduced Chronicle, a tool to support graphical document workflow exploration and playback. It allows users to explore document revisions, called chronicles, graphically, through a timeline, as well as video representation. Chronicles employs screen video capture to show the user the evolution of the document, interaction recording, and some application specific logic to detect changes to the document. The main interface can be seen in Figure 2.10. The main chronicle window (Figure 2.10a) shows the step by step changes made to the document. These changes include visual changes, as well as layer information in the image that may not be visible. The timeline (Figure 2.11) shows a detailed view of the actions taken by the user throughout the document's history. Finally, the playback window (Figure 2.10c) shows the entire process as captured, in video form. Chronicle also includes data probes to allow users to filter out specific actions taken by the user. Nakamura and Igarashi [34] created a similar system that aims to be application independent. It monitors GUI events and uses screen snapshots to give visualization to the user, instead of long videos. They use a comic-strip metaphor, and using annotations and still screen snapshots, they are able to convey user action history. An example of this is shown in Figure 2.12. As we can see, the annotations on top of the GUI make the actions fairly clear and easy to understand.

Researchers interested in studying history of users may find work by Heer et al. [21] interesting. Their work brings forth new visualizations for user action history.

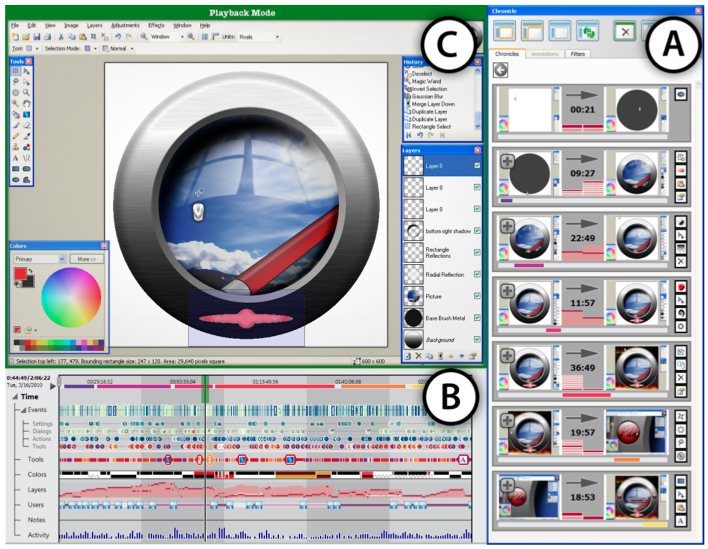


Figure 2.10: Chronicle, by Grossman et al. a) main Chronicle window, b) timeline, c) application/playback window

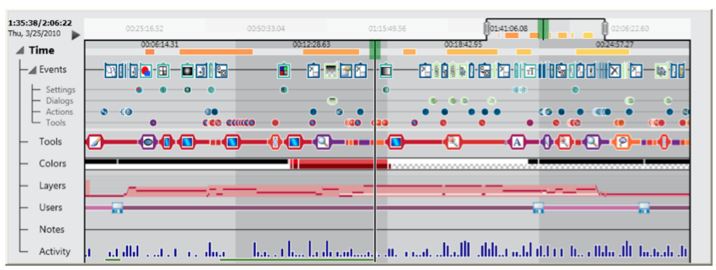


Figure 2.11: Chronicle Timeline, by Grossman et al.

They created a database visualization system, Tableau, which allows for recording and visualization of interaction histories, and supports data analysis as well as mechanisms for presenting, managing and exporting histories. Guimbretiere et al. [20] developed ExperiScope, which is an analytical tool for visualizing user actions in an experimentation setting. It allows interaction designers to identify key patterns in subject interaction, and allows easy analysis of these interactions.

Another example of history would be the one found in every web browser. Navigating back to previously viewed web pages is a useful feature and allows

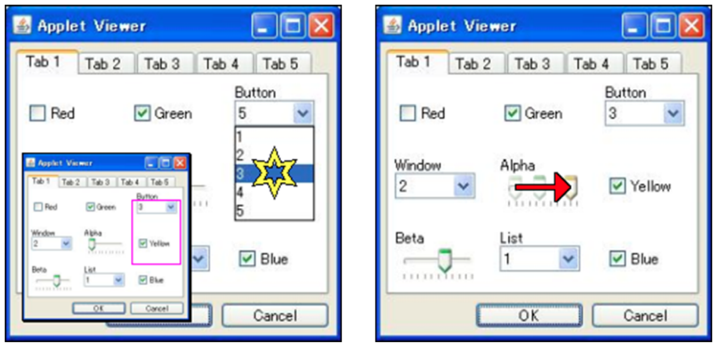


Figure 2.12: Annotated history, by Nakamura and Igarashi

users to not only view their progression, but allows users to re-find information far easier than retracing their steps manually. In fact, web browser history is a popular domain for research, and many researchers have created many different tools to related to keeping, recording, and reusing history. Anupam et al. [4] developed WebVCR, a system allowing users to set a recording mode, record their interactions with a web page, and play them back for later use. The interface itself is simple and included the conventional VCR playback controls mentioned earlier, as seen in Figure 2.13. The purpose was to allow users to navigate through webpages using macro-like behaviour where conventional bookmarks failed, such as session sensitive pages. Hupp and Miller [23] extended the idea, and allowed users to not only developed smart bookmarks, an extension to WebVCR, can be shared with other people, as well as edited after creation. Furthermore, unlike WebVCR, the system allows users to retroactively create macros without explicitly pressing record. As a passive macro creation tool, this allows users to concentrate on the task at hand. Li et al. [32] continue to extend this idea with ActionShot. Contrary to a traditional web browser history, ActionShot records history at a more detailed level, down to the specific interactions made on a web page, such as textual entry, ticking a checkbox, or clicking a button. This then allows users to create macros like those found in Adobe Photoshop or Microsoft Word. A screenshot of the history view can be seen in Figure 2.14. The system implemented also allowed users to share their executables with other people.

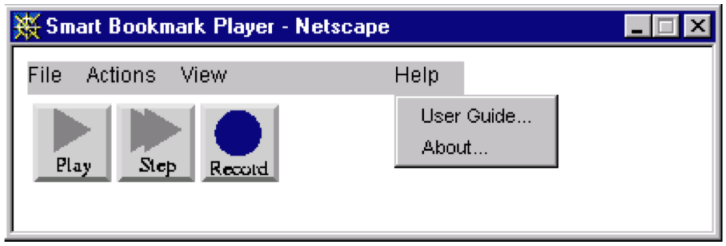


Figure 2.13: WebVCR by Anupam et al.

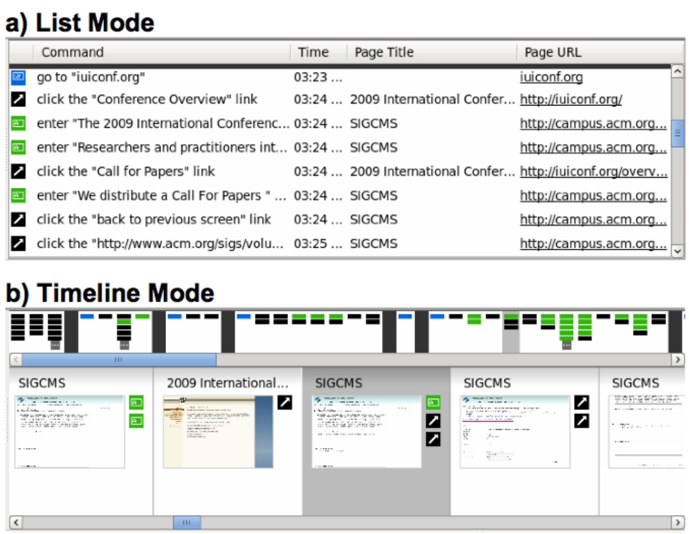


Figure 2.14: History Views of ActionShot by Li et al.

In systems such as Facebook[7] (Figure 2.15) and Twitter[8] (Figure 2.16, events are shown in a vertical fashion. This is in stark contrast to the layout found for timelines. These conventions provide semantic cues to the user to indicate the difference between a user action or event history and the video timelines. We take these cues into our own work, placing our history views into a vertical orientation, and our video timelines in a horizontal orientation.

History can also be used to create tutorials. Like the previous methods, this employs recording to playback actions that tutorial creators have performed. Bergman

[7]http://www.facebook.com
[8]http://www.twitter.com

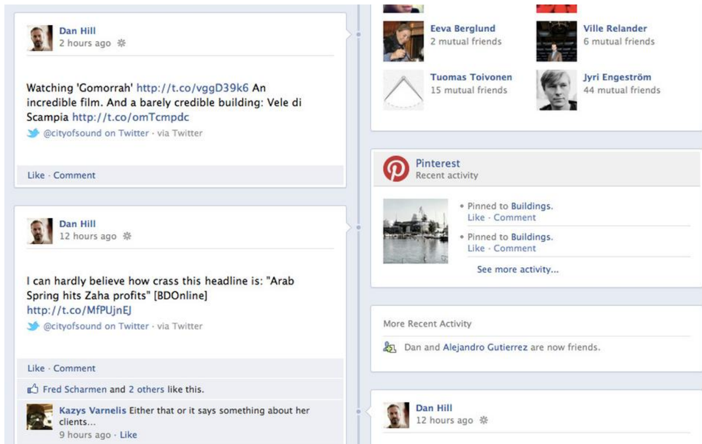


Figure 2.15: Facebook's vertical event timeline

et al. [6] such a system. Their solution, "follow-me documentation wizards", continually show users the current position in the procedure, highlight the relevant application controls, and like the previous systems described, can automate portions of the procedure automatically. Their recording system turned the user's actions into a script language that they can then edit retroactively to change actions, or to add annotations.

All of the systems mentioned above are based upon workflow and tutorial generation. They all use some method of capturing user interaction history, and all use that information for repeating boring tasks, or for teaching and so do not directly apply to video.

2.5 Summary and Influence on Design

In this chapter, we covered video browsing, video authoring, and user action history. In the video browsing section, we take inspiration from video preview thumbnails, as well as timelines introduced to allow for non-linear search through a video. Under video authoring, we gain insight on the different types of interfaces seen in popular video editors, and the use of automatic methods for generating video intervals. Further, we investigated the use of user action history in tutorial creation and sharing, which has direct implications for video authoring and sharing.

Figure 2.16: Twitter's vertical event timeline

Chapter 3

General Interface Design for History Based Editing

Based on the functionality found in professional video editors, we developed an interface that supplies users with four basic functions: watching video, searching video, selecting video clips, and authoring new video. The important components in our design are: a video player, a user-history-timeline, and video clip list to collect video clips. We describe the complete video interface that integrates our components in features from typical commercial based ones for our test. We rely on some of the design decisions found in Apple's iMovie, as that is a current video editor focused on beginning video editors and provides a basis for comparison in our user tests.

3.1 Overview

A screenshot of the interface can be seen in Figure 3.1. In the middle, the large picture is the main video view, similar to most video players and video editing software. Clicking on the video view allows the user to toggle playing and pausing of the video. As a large target, this makes it easy for the user to stop the video. This same functionality can also be found in YouTube's player as well as various video viewing applications found in the mobile space.

The rest of the interface, such as the history (yellow, Section 3.5, filmstrip

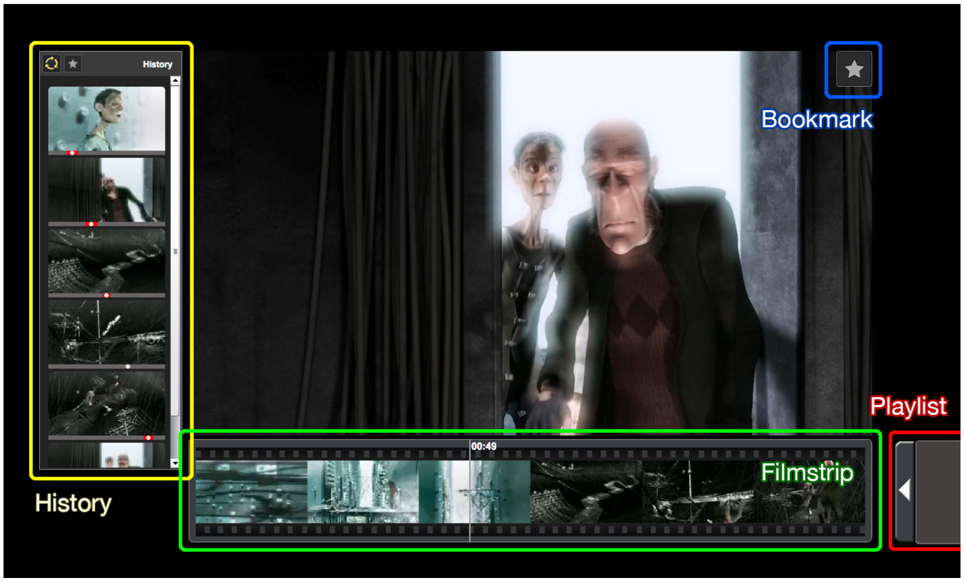


Figure 3.1: Overview of the interface. Here, we see the history (yellow), and the filmstrip (green), and a bookmarking button (blue)

(green, Section 3.4), and the playlist (not shown, Section 3.6) will be described in detail below. We first describe the general interface design decisions.

3.2 Design Language

There are three design decisions that we made to account for consistency throughout the entire interface.

1. The design of the interface revolves around the concept of video intervals, each of which is represented by a small thumbnail preview, similar to those found in [15]. Manipulation of video can be achieved by dragging thumbnails around, from one widget to another. For example, dragging a thumbnail over to the main video view, will cause the viewer to start playing at the interval specified by the interval, and pausing when the interval has ended. Dragging a thumbnail to a playlist will add it to the playlist.

2. The width of the thumbnail generally represents the entire length of the video interval that it is showing. The actual temporal location of the represented

interval is communicated to the user by a small, red, horizontal bar along the bottom of each thumbnail acting as a miniature timeline. The spacial representation of video length versus width is maintained throughout the entire interface and can be seen with the filmstrip and the width of the main player, show in Figure 3.1.

3. The arrangement of the two different timelines is also taken into account. We wanted to separate video timelines and user-history timelines so that users would not become confused with the two. Since video timelines are conventionally horizontal, (and this guideline is also carried out in the miniature timeline in all thumbnails), we decided to lay out user-histories in a vertical fashion, as mentioned earlier in Chapter 2.4. This provides a mental separation of the two concepts and prevents confusion.

These guidelines provide the application with consistency so that the interface can be more intuitive and easy to use. In the next sections, we describe the various interface components.

3.3 Thumbnail

Thumbnails are used to represent video intervals. By placing the cursor over the thumbnail, a miniature timeline appears. This can be seen in Figure 3.2. The bottom grey and red line represents the timeline of the entire video, with the red section showing the thumbnail's video interval. In accordance The popup thumbnail, is also in red, is a zoomed in version of the bottom timeline. Using this, the user can then utilize the entire width of the thumbnail to scan across the video interval. It only partially obscures the thumbnail and is semi-transparent, allowing the user to continually see the preview while scanning.

Interacting with these thumbnails can be achieved by dragging and dropping them around various elements of the interface. Clicking on the button will allow the user to play the video interval in the main player, and highlight the corresponding video interval in the filmstrip in blue. The same action can also be performed by clicking and dragging the thumbnail over to the main player. The button allows the user to insert the video interval into the playlist. To indicate successful

Figure 3.2: Video Interval Thumbnail. The red bar on the bottom represents the video interval. The timeline pops up allowing the user to search through the interval.

insertion, a ghost thumbnail is animated to fly over from the history to the playlist. The ★ button allows users to manually mark an interval as a favourite. By making users move thumbnails and giving video intervals a concrete representation that users can see, we can ensure that users are kept in the know about everything happening on the screen.

3.4 Filmstrip

The filmstrip is the interface element bound in green in Figure 3.1. The filmstrip allows users to seek to any part of the video. Unlike timelines that only feature a playhead and a timestamp, the filmstrip provides users with a preview to allow better accuracy in seeking, rather than only relying on timestamps to judge the location of the potential seek. In the first version of this interface, both the filmstrip and the traditional (non-previewable) timeline are given to the user.

The filmstrip is a set of the thumbnails described in Section 3.3, placed along a horizontal axis, respective of the design guidelines. There are six thumbnails and each thumbnail represents $\frac{1}{6}$ of the entire video. We decided to use six in this version, as the resulting height was large enough to allow users to see the preview while not interfering with the main viewer. The highlight in blue indicates a currently playing interval invoked by the user. The red highlight indicates a video interval being selected by the user.

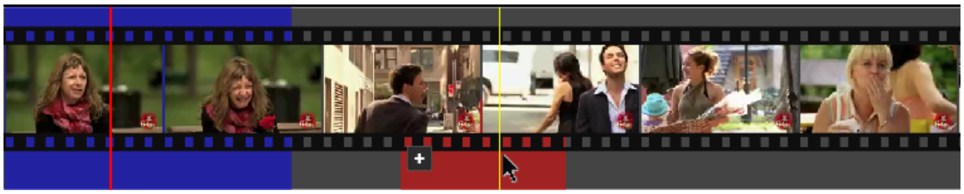


Figure 3.3: Overview of filmstrip. It is used for seeking and searching through a video.

There are two methods to seek the video. The first method is the traditional method and involves clicking on the timeline slightly below the main player. The second method involves the filmstrip.

3.5 Navigation History View

The history view is the interface element bound in yellow in Figure 3.1. The history is a mechanism that records a user's video watching behaviour, and represents it in a method that a user would be able to understand and navigate. Its purpose is to allow users to see which parts of a video they have watched most, go back to review them, and create video clips by rewatching the video.

The history, shown in Figure 3.4, allows users to revisit old video intervals that were previously viewed. These intervals are represented by thumbnails found in a scrollable field. This size was chosen to make the thumbnails relatively the same size as those found in the filmstrip. They can also be previewed within the thumbnails by clicking on them. Like the filmstrip, each thumbnail can be dragged to the main player to be previewed. As noted in our design language in Section 3.2, a vertical layout is used to here to help the user differentiate between the video timeline, and user-action time. The order of the thumbnails is arranged with the bottom being the most recently watched history item. The reverse order was found to be confusing in our own tests, as moving all the thumbnails down caused too much motion and distracted from the video viewing experience.

The history itself starts off with one interval, and this interval extends itself to the main player's time until a navigational action, such as seeking, is performed with the filmstrip or the timeline. Upon such an action, the current interval ends

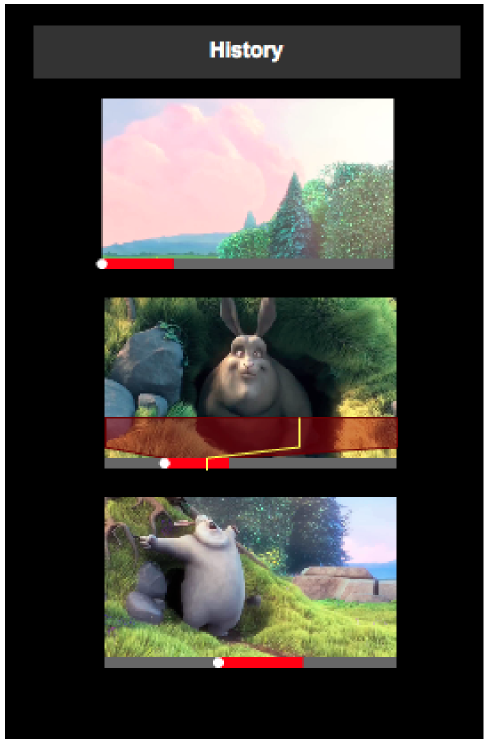


Figure 3.4: Overview of history. History items are created as the user watches and navigates around the video. Each history item has a start and end point that can be explored by placing the cursor over it.

and a new one begins. Visually, a new thumbnail is added to the history.

In initial pilot studies, users found it difficult to create history items in exactly the way they wanted to. They were confused about how to stop the currently created history item from being extended, and seeking was not sufficient in aiding with that. In an effort to provide the user with a better solution, we came up with the system shown in Figure 3.5. Across the top of the history, there is a drop-down selection box that allows users to filter their history for segments of video that they have seen multiple times. Upon selecting and filtering out clips that they have seen twice or more, for example, the history will find the intersections of all history items for which the view count is higher than one. The envisioned method for using this to create clips is:

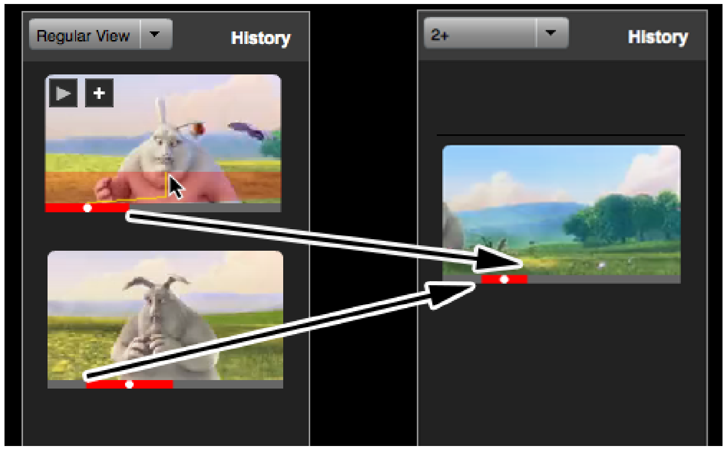
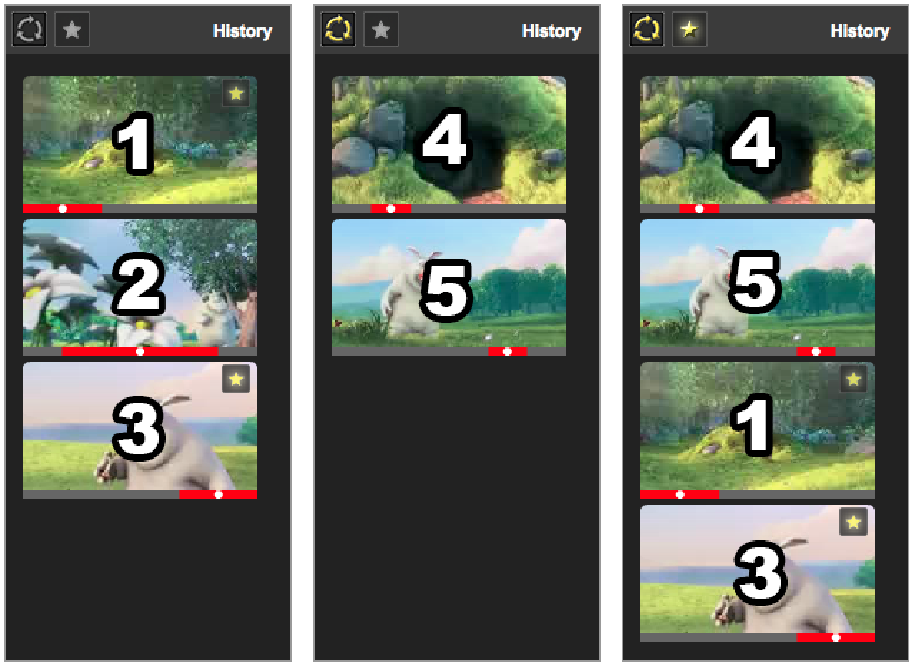


Figure 3.5: Re-watch filter in history.

1. Watch the video clip
2. Come across an interesting moment
3. Finish watching the interesting moment
4. Seek back to where the interesting moment starts
5. Finish watching video clip
6. Find intersection of history items with a view count >1

As such, setting the start time is achieved in step 4, and setting the end time is achieved in step 3. This removes the need to predict if future clips coming up are going to be of interest.

Manual creation of history items was also included, and we decided to add another filter to allow users to search for these specially annotated history items. These intervals can be created by toggling the "bookmarks" button seen in Figure 3.1 (blue). Clicking on the button will cause the history to start recording while the video is playing, and clicking it again will stop the recording, much like the interaction found in modern personal-video-recorders. Combined with the re-wawtched filter, these allow users to see what clips they know to be of interest, as well as clips they may not realize they found interesting. In Figure 3.6a, there are



(a) No filters **(b)** Re-watched filter **(c)** All filters

Figure 3.6: History Filtering

three clips that the user watched. Clips 1 and 3 are favourited. If we turn on the re-watched filter, as in Figure 3.6b, we can see the intersections of all the clips, created as new video clips 4 and 5. If we turn on both the re-watch filter and the favourites filter, we see that the favourites, clips 1 and 3 reappear, as in Figure 3.6c.

3.6 Playlist

We can bring out a playlist by clicking on the arrow button on the far right in Figure 3.1 (red) which is shown in Figure 3.7. The playlist functions as a location for users to store video clips. Using this users can create their own videos filled with clips that were of interest. This functions as a simple video editor, allowing users to playback, lengthen, and shorten clips.

The playlist, shown in Figure 3.7, contains several thumbnails. Each playlist item is represented by a grey box with a thumbnail showing the clip's contents.

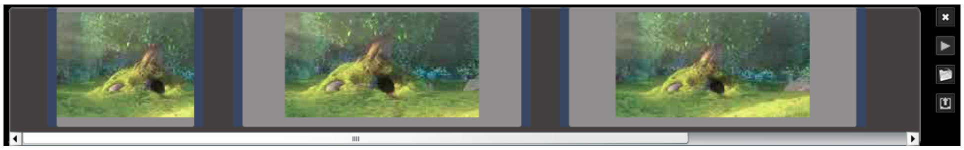


Figure 3.7: Overview of the playlist

Like the previous thumbnails, this thumbnail seeks to represent the location of the cursor over the length of the playlist item. On the left and right edges of each playlist item is a draggable region, allowing users to set the clip's starting and ending points from the video. This larger region makes modifying the clip easier than the last version. Upon dragging either edge, the thumbnail within the grey box will change to reflect the edge that the user has dragged, providing a preview for the change.

With this interaction, the temporal length of the video clip correlates with the width of the playlist item represented on the screen. Unfortunately, due to the variable width, sometimes the preview thumbnail is not all visible, which makes previewing the clip difficult. Our solution was to have a popup thumbnail preview, as shown in Figure 3.8. This thumbnail pops up whenever the cursor is over a playlist item whose width is less than the original preview thumbnail. It also appears every time one of the edges are dragged to provide the user with a larger, easier to see thumbnail.

There are also some buttons to the right of the playlist. In order from top to bottom, these were: a close playlist button, a play button, an open existing playlist button and an export playlist button. Button 1 hides the playlist back into its original position. Button 2 plays the contents of the playlist in the video viewer. For the third and fourth button, we implemented an option to save and open playlists. Playlists are saved as simple XML files that are portable and easy to read. The last button is used for the evaluation and was for the user to indicate that they were finished performing the task.

The interface should be able to support users in watching, saving and sharing video intervals, and do it in a more seamless manner than what is provided in separate video viewing, editing, and sharing software suites. We carry out preliminary evaluations to evaluate the effectiveness of not only the interface, but more im-

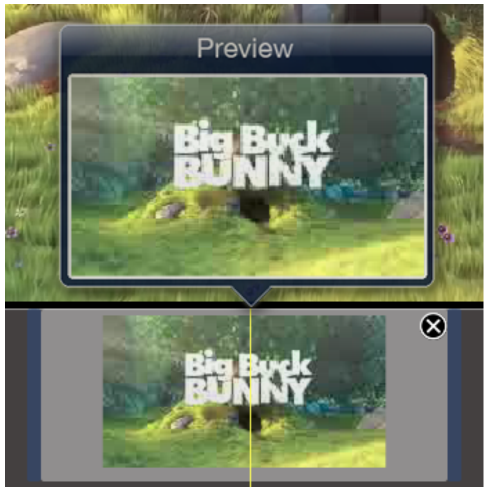


Figure 3.8: Preview thumbnail in the Playlis

portantly, the interaction of re-watching video to create video intervals for further use. Throughout the next three chapters, we will be describing the testing, and the iterative evolution of the interface.

Chapter 4

Comparison of History and Filmstrip Selection

In this chapter, we discuss the first experiment to evaluate our interface. This evaluation was performed following pilot studies to polish and receive initial feedback from users about the use of the History for a video authoring task. Here, the pilot studies helped us make the decision for horizontal and vertical timelines. In previous iterations, our playlist was oriented vertically. Placing the playlist in a horizontal layout also made altering video intervals more accessible. In previous iterations of the History, there was no method for filtering or otherwise managing the History and pilot studies showed that there needed to be a better method for creating video intervals, and so we introduced the re-watched filters.

4.1 Experiment

A user study was carried out to evaluate the design and performance of our interface and the usefulness of using personal video navigation histories for creating playlists. We developed an evaluation protocol to ensure users have sufficient viewing history while keeping the experiment short and maintaining a bias free evaluation. Likewise, for comparison to current practice, we needed to ensure that our interface mimicked currently adopted approaches as well as logical extensions to them to provide a fair comparison to using viewing history. Using our protocol, we investigated whether visualizing and using a video navigation history would make creating and inserting clips to playlists more efficient. We conducted a comparative user study, comparing the performance of participants using a personal navigation history against the state-of-the-art navigation method (Filmstrip) to find clips to be added to a playlist.

4.1.1 Hypothesis

We hypothesize that participants save video intervals to the playlist in less time using the History method than the Filmstrip method, and prefer the History method.

4.1.2 Apparatus

The experimental interface was implemented using Adobe Flash CS6 and ActionScript 3. The study was performed on a 15" MacBook Pro, with a screen resolution of 1680x1050 pixels. Participants used a regular Microsoft mouse to manipulate the contents of the screen. The application window was set to a size of 1280x720 pixels.

4.1.3 Participants

Eighteen volunteers, 13 male and 5 female, participated in the experiment. They were monetarily compensated for their time. Participants ranged in ages from 19 to 50. Each participant worked on the task individually. All participants were experienced computer users and watched videos at least 3-5 times a week. 11 rarely created videos, and 7 have never created videos.

4.1.4 Design and Procedure

There were two methods and six videos experimented by each subject. Since participants will be selecting video intervals and relying on their memory, we ensure that they watch each video only once to reduce the learning effect on the video. We separated participants into two groups. One group used the first set of videos with the History, and the second with the Filmstrip, while the other group reversed the sets. This allowed participants to use both methods and only view each video once, and the design is shown in Figure 4.1. Each set of videos consists of 3 two-minute

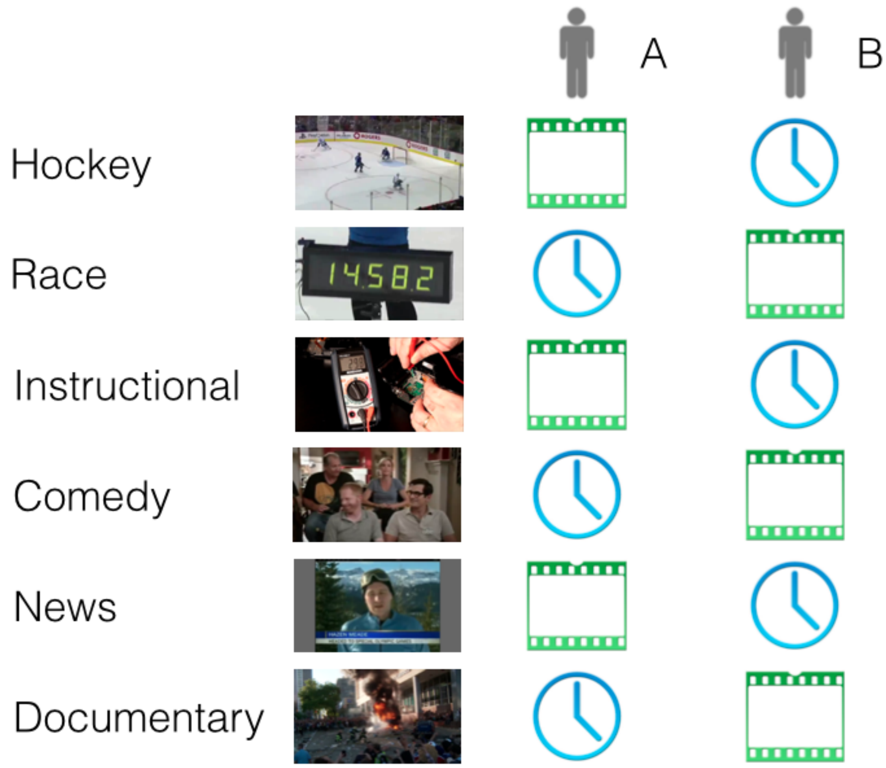


Figure 4.1: Experimental Design: History and Filmstrip

videos. The content of these videos varied from sports, news segments and comedy shorts.

Each participant was asked to watch each video and to re-watch parts of the video that follow two themes specified by the experimenter. When the participant finished viewing the video, they were asked to find 2 to 3 specified clips, which belong to one theme, from within the viewed video and add them to the playlist. Using History, participants were asked to locate the clips in the history using the view count filter and add them to the playlist. To restrict the participants to use only History, adding items from the Filmstrip was disabled. Using the Filmstrip, participants were asked to use the selection tool by clicking and dragging along the Filmstrip to select a portion of video to add to the playlist. To restrict the participants to use only Filmstrip, History was hidden from the interface as soon as the viewing phase had ended. Incorrect selections would be indicated to the participant and they would be asked to add the correct one.

The participants were asked to complete the task as quickly as possible. For each task, the time taken to perform the clip selection (i.e. adding them to the playlist) was recorded. The timing begun when the user clicked the start button, and stopped when the stop button was clicked. We checked whether the correct clips were chosen to ensure a proper evaluation among all participants and if not these clips were considered as error.

The experiment proceeded as follows:

1. The evaluation started with a familiarity phase for each technique. Participants were shown how to use each of the interface elements, and most importantly, how to add clips to the playlist from the history and the Filmstrip. They were also shown how to create a useful history through video review, and how to access created clips with the view count filter.

2. After the training phase, a trial started by asking the participant to watch a video and re-watch clips that fit in the two themes provided by the experimenter (e.g. shots of fire fighters and police officers).

3. The participants were asked to add clips to the playlist using the available component for the tested method.

4. Once completed, participants were advanced to the next video and using the alternate method. Upon the completion of the six videos, participants were given time to experiment with both techniques at the same time on a blooper reel. This allowed us to explore which method participants will prefer when they are give the option of using both.

5. They were then given a questionnaire to fill out asking about their reactions to the interface.

The total time taken for the experiment was approximately one hour.

4.2 Results and Discussion

We would first like to note that, as explained in the design, this was a within participant study that was conducted with between elements for the method and videos. Using this design, we were able to get more informed data from the participants who would be able to give a more accurate opinion of both techniques. The analysis was done between so that each participant was exposed to each video only once with a single method.

A two-way ANOVA analysis test was carried out to explore the effect of method and videos on the time needed to create a playlist. The analysis showed significant main effects of method ($F(1, 96) = 66.69$, $p < 0.0001$) and of video ($F(5, 96) = 5.86$, $p < 0.0001$). However, these main effects are qualified by a significant interaction effect ($F(5, 96) = 2.37$, $p = 0.045$). Simple main effects of the video at method show that for each video, the time needed to find clips and insert them into the playlist was significantly faster using History over the Filmstrip. Participants took at most two-thirds the average time using History than using Filmstrip as shown in Figure 4.2.

The evaluation required participants to create video intervals containing the interval of interest. As the target audience is casual video watchers and authors, the accuracy of was not important. The qualifier for a correct trial was for either a portion or the entire interval was inserted into the playlist. As a result, the error rate in this experiment was zero.

Of the six videos, we found that History worked better for some videos than others, and the same effect can be observed in the Filmstrip. These traits can generally be attributed to the content of the videos, which is why the analysis is separated for each video.

1. The first video consisted of a short hockey video of a power-play. The themes for the video were two goals, and two segments where the defensive team had possession of the hockey puck. The participants were asked to find the two goals. In this video, it was difficult to see what was going on within the thumbnails. This was particularly difficult with the Filmstrip because many participants forgot when the first goal happened, and were forced to look through the entire video in the Filmstrip, whereas History reduced the

search space required, down to four short clips.

2. The second video consisted of six hockey players, three wearing blue jerseys, and three wearing white jerseys, skating around a rink and competing for the fastest time. The theme in this video is the hockey players turning the last corner. The participants were asked to find the white jerseys to insert into the playlist. Again, like the previous video, participants had to search through the entire Filmstrip to find the correct clips, while History only required participants to look through six different clips, all of which were easily distinguishable by the screenshot within the thumbnail. Furthermore, because one of the players skates in the opposite direction of the other five, using the Filmstrip became slightly more difficult because the participant had to distinguish the different turns within the small thumbnail.

3. The third video was a short comedy segment featuring the cast from the sitcom Modern Family, where a little girl runs around, makes smart remarks, and pulls pranks on various people. The themes to this video were the reactions to her remarks, as well as the result of the two pranks she pulled. The participants were asked to find the clips with the two pranks. This video had a slight advantage given to the Filmstrip because the scene of one of the pranks was visible in the Filmstrip, and the second prank was right at the end of the video, making both mental and physical retrieval relatively easy. The data reflects this condition as this was the fastest video for Filmstrip.

4. The fourth video was a comedic instructional video of someone taking apart a camera. Along the way, the instructor would make mistakes that were very obvious, and he would disconnect things within the camera. The chosen theme for this video were the three mistakes made throughout the video. Like the second video, the clips were fairly spread out along the video, however they were somewhat hard to see within the Filmstrip, which made it slightly more difficult using that method. Additionally, some participants forgot the location of the clips and were forced to look through the entire movie, like the first video.

5. The fifth video was a news segment on an Olympic hopeful wanting to enter

the snowshoeing competition. The theme of the video was the six talking heads within the video. Of those, the participant was asked to find two specific ones. This ended up being very easy for History, as it was not visually tasking to find clips of a specific person. Again however, one of the clips appeared in the Filmstrip, and the second clip was right after.

6. The last video was of the 2011 Stanley Cup riot in Vancouver, and the themes of the video were of shots of fire fighters and police officers. The clips to be selected were the fire fighters. These clips were mostly within the middle 30 seconds of the video. For the Filmstrip, some participants started searching from the beginning of the video, as they did not remember when the first clip appeared. They were, however, very close together within the video and allowed for quick selection once they found the first clip. They were visually distinctive from the police clips, and were also easy to find within History.

These videos were chosen to represent real world video, and sometimes the video interval of interest occurs near the beginning or end of the video. As a result, finding these intervals using the Filmstrip is much easier in these cases and explains the variation in the times taken by Filmstrip selection. Choosing events that were more spread throughout the video would have minimized this effect. In Figure 4.2, we see that the variation in History is minimal, and History provides a consistent time expected to search for video intervals.

Watching the participants use each of the methods allowed us insight into the weaknesses of each technique. The Filmstrip provided a linear view of the entire video, and provided no clue as to the location of specific video clips that needed to be found. As such, participants would often forget where the clips were and needed to search through the entire video. The time taken was further aggravated when the video clips were spread evenly throughout the video. It should be noted, our implementation of the Filmstrip did not feature a zooming function. This would, however, likely result in longer times taken for the clip selection task because of the need to scroll horizontally to find clips, and would also hide clips from the user, making retrieval slightly more difficult. But if we were to extend this interface to work for many more videos, and especially those longer than a couple of minutes, zooming becomes a necessity to maintain accuracy in clip selection.

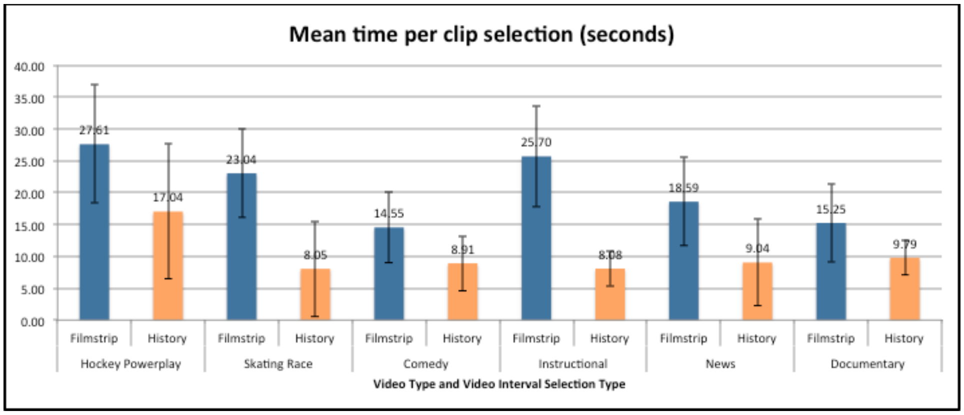


Figure 4.2: Descriptive Statistics for History vs Filmstrip Interval selection. Mean times are shown for each video and method. Numbers are in seconds.

The questionnaire results shown in Table 4.1 demonstrate the positive reaction of participants to our history-based authoring interface. The overall scores for the History method were all above 5 (on a Likert scale of 1 to 7). The Filmstrip method scored well but participants did not find it as easy to find video clips; this is supported by the quantitative data.

Observation of users in the free play section of the study showed a strongly positive reaction to using History. Eleven users created a history the way we intended, creating a proper history, and using the view count filter as intended. This is shown by the questionnaire that not only is it useful, but a usable history can be easily created. Three participants created a history but did not use the filter to add clips, but instead created a playlist with whatever happened to be in the history. It is likely that these participants forgot to use the filter and did not realize it. Two of the participants used a combination of the two techniques. For example, they used History as they were taught, and after using History, use the Filmstrip to find additional clips that they did not think to re-watch at the time. Finally, two participants were unreceptive to creating and using the history, and only used the Filmstrip to create the playlist.They stated that they did not think that the viewing pattern required by the interface fit with their own viewing behaviour, and that doing so would disrupt their video viewing experience. While these participants did

Question	Mean	σ
Overall usefulness	5.77	1.06
Overall ease of use	5.61	1.29
Overall reaction	5.47	1.19
Learning to use the interface	5.50	1.42
History is useful	6.11	1.23
Creating usable history is easy	5.56	1.29
History filtering is useful	5.89	1.07
History filtering is intuitive	5.28	1.56
Finding history video clips is easy	5.44	1.89
Finding Filmstrip video clips is easy	4.50	1.89
Inserting history video clips is easy	6.44	0.70
Inserting Filmstrip video clips is easy	5.61	1.50

Table 4.1: The aggregated results of our questionnaire, with the mean score (Likert Scale, 1 to 7) and standard deviation (σ). Participants found the interface with the history to be useful for the creation of new videos. Their overall reaction to our novel video authoring mechanism was highly positive.

not use History, it still recorded the video they watched. In this case, the history contained a single item for the entire video. If we extend the system to multiple videos, it would be easy to see which videos they liked, if the video was viewed more than once, and when they watched them. While only two participants used both techniques to create their own video in the free play, it is worth noting History works well in conjunction with the Filmstrip. Further integration between the two, such as placing markers in the Filmstrip indicating history intervals may improve performance even more.

The experiment showed that users were able to make use of History and its filtering mechanism fairly effectively, and well enough that it was significantly faster than using the Filmstrip. Furthermore, it also revealed that the type of video used also affects the performance of either History method. These were illustrated by the interaction effect found between video type and selection method. Allowing users both methods of selection showed that the two methods can also work in conjunction with each other.

4.3 Summary

This chapter introduced methods for browsing through history, which helped users make accurate interval selection. The filtering method allowed users to more easily set end times for the selected intervals, reducing the need to adjust the intervals in the playlist. We performed an evaluation of the interface, and found that users were able to use History effectively, and were able to select video intervals significantly faster over the traditional Filmstrip method.

Chapter 5

Comparison of Implicit History and Explicit Favourites

In the last evaluation, it became easier to see that the use of history to find specific video clips was much easier and quicker to use than the filmstrip. It was noted by the users, however, that having to re-watch the video clips to input them into the history was slow, and sometimes unnecessary. They would then like to manually create video clips without needing to re-watch them. This approach is similar to work by Li et al. [31]. In this work, we categorize the events in our videos into two categories: predictable and unpredictable. Predictable events have a lead-up and are easy for users to foresee, whereas unpredictable events have no lead-up.

5.1 Experiment

A user study was carried out to evaluate the design and performance of our interface, the usefulness of using personal video histories and Favourites for video interval creation and retrieval. We allowed users to freely watch videos and create a history of video intervals that they would share with others. We conducted a qualitative user study to evaluate the use of History and Favourites in video interval search. Following our scenarios, we give our participants videos to watch categorized between the presence of predictable and unpredictable events.

5.1.1 Hypothesis

We hypothesize that participants use and prefer the History method for unpredictable events, and use and prefer the Favourites method for predictable events.

5.1.2 Apparatus

The experimental interface was implemented using Adobe Flash CS6 and ActionScript 3. The study was performed on a 15" MacBook Pro, with a screen resolution of 1680x1050 pixels. Participants used a regular Microsoft mouse to manipulate the contents of the screen. The application window was set to a size of 1280x720 pixels.

5.1.3 Participants

Eleven volunteers, 7 male and 4 female, participated in the experiment. They were monetarily compensated for their time. Participants ranged in ages from 19 to 50. Each participant worked on the task individually. All participants were experienced computer users and watched videos at least 3-5 times a week.

5.1.4 Design and Procedure

There were two methods and eight videos experimented by each subject. Each participant was exposed to each video only once. The two methods for video clip creation were tested: the History, and Favourites. We performed a study where each subject used each method on separate videos. Each video was 2-3 minutes long. The experiment proceeded as follows:

1. The participants were given a training session. Participants were shown both methods for creating video intervals for saving, and they were given time to try it for themselves.

2. The participants were given two mock trials (but were not told that they were mock) to further familiarize themselves with the tools. For the first mock trial, they were given a predictable video and the re-watching feature to use. The option to view and create Favourites was disabled for this trial. They were given 1.5 times the length of the video to watch it and create a history

by re-watching segments. During this time, they were asked to create at least three different history items.

3. After viewing, the participants were asked to locate two of the created intervals that the experimenter named.

4. The participants were then asked to fill out part of a questionnaire, which asked about the effectiveness for the method applied for that particular video.

5. The participants would repeat the task with Favourites enabled and the re-watching feature disabled with an unpredictable video.

6. The participants then repeated the task, twice with re-watching with both a predictable and unpredictable video, and then twice with Favourites, again with a predictable and unpredictable video. The opposite feature is always disabled.

7. After each trial, they were again given the questionnaire to fill out detailing their thoughts on using the method for creating video intervals.

8. They repeated the task with both methods enabled. They were given free reign as to which method they would like to use, and observations were taken to review the methodology in which participants employed with each method and when they used them.

9. They were then given a questionnaire detailing their preferences for each technique, and which videos they thought each technique was effective for producing a retrievable video clips.

The entire experiment took approximately one hour.

5.1.5 Videos Content

In total, the participants were given nine videos to watch, with one training video. There were two types of videos, which were categorized as having events that were predictable and unpredictable. Predictable videos can be described as videos that build-up to a particular moment as the climax of the particular segment. Unpredictable videos have minimal build-up (at most 1 second) and are easily missed upon the first pass of the video. Of the eight trial videos users were given to watch, four were predictable, and four were unpredictable. The contents of the four predictable videos are as follows:

1. A hockey shootout video. Clips were shown with players preparing to shoot, skating up to the goaltender and shooting the puck. Multiple replays from different camera angles were shown after each shot.

2. Another hockey shootout video.

3. Three clips of guests getting scared by a man on the afternoon talk-show, Ellen Degeneres. A man in a costume is shown sneaking up behind the guest and ultimately screaming at them while they are being distracted by Ellen.

4. Three clips of scientific experiments. All the clips include an explanation as lead-up to the climax. The first clip involved dropping rubidium in a bathtub, resulting in an explosion. The second clip involves a man walking on a non-Newtonian fluid. The third clip shows a rocket being fired on a along track towards a car, ultimately destroying it.

The contents of the four unpredictable videos are as follows:

1. Three different clips from separate videos. The first clip shows two skateboarders attempting tricks. A horn sounds and the camera pans towards an elderly lady walking across the street. A car revs its engines and the lady hits the car and the airbag pops open. The second clip is from a dash camera on a car driving along a road. From the left appears a car that is spinning and passes the car, and goes back into lane safely straightening out. The third clip is of a Ferrari Enzo driving and crashing into a cement block.

2. Three different clips from separate videos. The first clip shows an excerpt from a documentary about whales. The video shows closeups of smaller fish swimming in groups and a giant whale coming out and filling the screen, shown opening its mouth and eating all the fish. The second clip shows a dance competition, where a teenager comes out and performs some seemingly underwhelming dance moves. He then breaks out and performs spins

that show an enormous amount of skill. The third clip shows a flooded street. From the left, a car with an open trunk drives through the flood, and behind it is a man on surfboard being towed through the water.

3. Three different clips from separate videos. The first clip shows a stream of water. A hand then appears and turns on a switch that produces a 24Hz sine wave tone, augmenting the water flow in an interesting way. The second clip shows a stereo system playing a song on top of a desk. Soon after, the song produces multiple loud bass notes, shaking the desk uncontrollably. The third clip shows a washing machine in a yard running normally. Soon after, a person throws a brick into the washing machine, and it shakes uncontrollable until the drum is disconnected from the rest of the machine.

4. Three different clips from separate videos. The first clip is another dash-cam video. This time the car is following a large truck. It proceeds to turn left, and from behind the truck appears a smaller truck, causing a crash. The driver of the second truck pops out the front and is seen walking away from the accident, harm free. The second clip shows people fishing off a dock. As they try to use the net to pick up a fish, a shark appears and eats the fish. The third clip shows a girl gloating to the camera about selling a large amount of cookies. Through the middle of her gloating, she slips and hits her head on the table in front of her.

5.2 Results and Discussion

The results in this experiment were almost entirely derived from the questionnaire. The structure of the questionnaire consisted of eight sections, one for each video. Each section consisted of three questions:

1. It was easy to predict when something was going to happen

2. I could easily create the video clips I wanted

3. I could easily find the video clips I wanted

The aim of the Question 1 was to ensure that our own thoughts of the predictability of video was in-line with reality. For what we selected as predictable videos, participants rated the videos on average, from 5.8 to 6.5 on a scale from 1 to 7 where 1 was strongly disagree and 7 was strongly agree. For what we selected as unpredictable videos, participants rated them from 3.8 to 4.5 (neutral score). Using these metrics, we then asked users about which method they preferred with respect to the two different types of videos, as shown in Table 5.1. The results agree with our hypothesis that History is preferred for unpredictable videos and Favourites for predictable videos. The graph shows three people preferred History for both. One reason was that they did not mind having to re-watch segments of video, because it was an easier method of saving clips as we only tested for video clips under 10 seconds. The other reason was related to the ease of use and number of clicks required to create video intervals. For the participant who said Favourites is better for unpredictable, the reason was that creating video intervals was imprecise using the History method. Observation into the method in which they employed revealed that particular participant would watch an event, and seek back, and start Favourites precisely where they wanted.

Question	No.
re-watching is better for predictable	3
re-watching is better for unpredictable	10
bookmarking is better for predictable	8
bookmarking is better for unpredictable	1

Table 5.1: Questionnaire results for History vs Favourites Clip Selection. When questioned, most users stated that the use of re-watching was more suited towards videos that were unpredictable, and bookmarking was better for videos that were more predictable.

These results lead us to the observable usage patterns of Favourites creation. These can be shown in Table 5.2. The results here show that at least half of the participants rewind and re-watch parts of video they liked in order to create video intervals, regardless of the video's predictability. For unpredictable videos, this means that most of the participants (9 of 11) manually produced the same result as the re-watching history. For the two participants who used Favourites without rewinding in the unpredictable videos, they were quick enough on the button that they could create video intervals during the event. It should be noted that the intervals created would be lower quality (they would give no context, and be very difficult to understand). For predictable videos, the results are split between the two types of users. Half the users created what is essentially an identical re-watching history, and half created no re-watching history at all. While these videos were predictable, and it was obvious something was going to happen, some participants did not create Favourites as they were watching the video and preferred to give each event a first pass before going back to Favourite them.

Use of Favourites	Type of video	
	Unpredictable	Predictable
rewind and then Favourite	9	6
Favourite straight through	2	5

Table 5.2: Results for use of the Favourites feature. When given the ability to use Favourites and a predictable video, the participants were split among re-watching interesting intervals to Favourite and Favourite while the video was playing. For unpredictable video, participants generally missed the interesting intervals and were forced to rewind and Favourite the interval on the second pass.

In the last two trials in the experiment, participants were given both methods to use freely when creating video intervals for both predictable and unpredictable videos. Results of the observation of usage patterns is shown in Table 5.3. Given the results, we can see that most of the participants used the History exclusively and did not bother with the Favourites technique at all. When asked about this, the participants stated that the re-watching method simple enough that taking a little bit more time to create video intervals was not bothersome. It would be interesting to investigate if this usage pattern holds true for videos with longer events. Another participant stated that using the Favourites feature required too many clicks to create intervals. Thus, contrary to our hypothesis, we have found that participants would rather use the re-watching method for creating short video intervals.

When asked about possible improvements to the interface, one participant suggested that ability to be able to seek while holding down the Favourites button. This

Method use when given both	No.
used re-watching history exclusively	6
used Favourites exclusively	3
used both where appropriate	2

Table 5.3: Results for use of History and Favourites when given both methods. When given both methods participants were more likely to use the-HHistory exclusively for both videos regardless of the type of video they were watching. Others used Favourites exclusively, and only two participants used the intended technique (History for unpredictable, Favourites for predictable) for the videos.

would extend the ability for users to create a Favourite even when the event had already passed, without having to use the filmstrip to seek back. Another participant pointed out that the creation of the re-watched intervals was slightly confusing. With the re-watching, it was clear where the start point of the video interval is, but not clear when the video clip would end. Additionally, it was stated by one participant that the notification that Favourites was active (the star button glows), was not easy enough to see. It was suggested that an outline be drawn around the edge of the main player to more visually signify Favourites.

Given the feedback about the interface, it seems that the automatic creation of video intervals through re-watching is a preferred method of interaction. However, it lacks the preciseness of the Favourites feature. It is worth further investigation to improve it in this respect. Possible fixes include Favouriting re-watched clips and allowing editing of Favourited clips, or being able to collect clips into a playlist for further editing.

The experiment showed that participants agreed with our hypothesis in using Favourites for predictable events, and History for unpredictable events. However, we found that in practice, participants more often used History for their video interval creation. This reaffirmed the utility of the re-watching history, and shows that Favourites are not suitable for replacing the History, but instead, complements History and filmstrip selection methods.

5.3 Summary

This chapter introduced an alternative method for creating history by allowing users to manually create a history without having to re-watch video intervals. We predicted that this would allow users to preemptively save video intervals that fore-saw to be interesting. We performed an evaluation to verify this, and found that while users vocally agreed with our hypothesis, they performed differently. Instead, users found it more convenient to re-watch intervals.

Chapter 6

Discussion and Conclusions

In this chapter, we present a discussion of the results of our experiments to evaluate a novel interface for video authoring using both a passive video viewing history and an active video viewing history. We also discuss the significance and contributions of the thesis, as well as a proposal for future research.

6.1 Discussion

In this thesis, we presented an answer to our research question:

Does integrating video history, based on tracking what a person is watching, into a video player provide an effective means for authoring for sharing online video content?

We have attempted to create an interface that is tailored to the way the human memory system works. Current video players and authoring tools provide some aid in helping users find video in the form of video previews and annotation, but none allow users to search for video intervals based on watching behaviour. In our interface, we adopt video previews, as well as annotation, but we present it in a form that users can easily interact with to search and insert into playlists for authoring and sharing.

The results from our experiments indicate that users are able to effectively and quickly author videos based on what they watched. In our first evaluation, we made a comparison between the use of history and state-of-the-art selection technique, filmstrip, for video clip selection and playlist authoring. For all the videos tested, we found that the use of history was significantly faster in aiding users to search for specific video clips for previously seen video. This can be explained by looking at the amount of video the user has to search through. In the filmstrip, the user must manually reduce the search space by remembering specifically where the clips were in time in relation to the video. With the history, we can dramatically reduce the search space and allow users to more easily find the clips they want. In this evaluation, we found that creating a history can be inconvenient because of the need to re-watch video. For clips that do not need re-watching, this can feel cumbersome and slow. The evaluation itself was completed in a controlled environment and users were coached in advance on how to use the interface. In order to make the results more generalizable, a longitudinal study would be more favourable. The evaluation also showed that the type of video reflected differences in the performance of the history.

In our second evaluation, we looked to remedy the situation and introduced a method to create history items without having to re-watch any video. The introduction of a favourites button allowed users to manually create history items. In our experiment, we looked qualitatively at using both the passively created history and the actively created history. In the videos we chose, we specifically looked for events that differed in their predictability of appearance. This experiment attempted to improve on the experimental design from the first evaluation, providing users with more videos that had affective intervals that were more evenly distributed through the video. We hypothesized that predictable events would be easier to capture using the new favourites, and that unpredictable events would be easier to capture by re-watching. These hypotheses proved to be true from a theoretical standpoint (participants indicated that this is the method for which they would use the system), but in practice, participants were more likely to solely use the history for their interval authoring tasks. The favourites system is thus, more complementary to the history than it is a replacement.

Supplying users with their video navigation history, along with manual methods to create it, as well as proper methods for filtering and history management helps users in video search for authoring tasks. They also work well with more traditional point-and-drag methods of video interval selection like filmstrip, and are ideally used in conjunction with each other to create a simple and pleasurable experience for video authoring and sharing.

6.2 Future Work

In this thesis, we developed an interface for authoring video using history that was used for testing with a controlled set of videos and limited number of participants. The next step in evaluating this interface is to expand on that, and build a full fledged application with access to a larger library of videos, for example, those on YouTube, and to gather more data from a wider audience. To do this, we have started developing an application with greater functionality for the Apple iPad. This will allow us to more easily deploy the application through the Apple App Store allowing us to reach more users.

Throughout the work, we have come up with some design guidelines for video browsing and history tracking applications, outlined in Chapter3.2. Future work in this area would be to formalize these guidelines, and investigate the implications of each of these design decisions with respect to memory, as well as repetition in human behaviour. This can then be extended into studies on different methods in using the history, for different types of videos.

The contributions in this work have shown that giving users access to their video navigation history is beneficial for simple video authoring. Providing proper methods of filtering and managing this history is an essential part of video interval selection using this concept.

Bibliography

[1] A. Al Hajri, M. Fong, G. Miller, and S. Fels. Fast forward with your vcr: Visualizing single-video viewing statistics for navigation and sharing. In *Proc. Graphics Interface*, May 2014. → pages 8

[2] A. Al Hajri, G. Miller, S. Fels, and M. Fong. Video navigation with a personal viewing history. In *Human-Computer Interaction – INTERACT 2013*, volume 8119 of *LNCS*, pages 352–369. Springer, 2013. → pages 7

[3] A. Al Hajri, G. Miller, M. Fong, and S. Fels. Visualization of personal history for video navigation. In *Proceedings of the ACM CHI Conference on Human Factors on Computing Systems*, CHI'14, New York City, New York, U.S.A., April 2014. ACM. → pages 8

[4] V. Anupam, J. Freire, B. Kumar, and D. Lieuwen. Automating web navigation with the webvcr. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 503–517, Amsterdam, The Netherlands, The Netherlands, 2000. North-Holland Publishing Co. → pages 23

[5] M. Barbieri, G. Mekenkamp, M. Ceccarelli, and J. Nesvadba. The color browser: a content driven linear video browsing tool. In *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pages 627–630, 2001. → pages 12

[6] L. Bergman, V. Castelli, T. Lau, and D. Oblinger. Docwizards: a system for authoring follow-me documentation wizards. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, UIST '05, pages 191–200, New York, NY, USA, 2005. ACM. → pages 25

[7] R. G. Cattelan, C. Teixeira, R. Goularte, and M. D. G. C. Pimentel. Watch-and-comment as a paradigm toward ubiquitous interactive video

editing. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(4):28:1–28:24, Nov. 2008. → pages 19

[8] K.-Y. Cheng, S.-J. Luo, B.-Y. Chen, and H.-H. Chu. Smartplayer: user-centric video fast-forwarding. In *Proceedings of the 27th international conference on Human factors in computing systems*, CHI ’09, pages 789–798, New York, NY, USA, 2009. ACM. → pages 14

[9] M. Christel and N. Moraveji. Finding the right shots: assessing usability and performance of a digital video library interface. In *Proceedings of the 12th annual ACM international conference on Multimedia*, MULTIMEDIA ’04, pages 732–739, New York, NY, USA, 2004. ACM. → pages 12

[10] M. G. Christel, M. A. Smith, C. R. Taylor, and D. B. Winkler. Evolving video skims into useful multimedia abstractions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI ’98, pages 171–178, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co. → pages 17

[11] A. Divakaran, K. Peker, R. Radhakrishnan, Z. Xiong, and R. Cabasson. Video summarization using mpeg-7 motion activity and audio descriptors. In A. Rosenfeld, D. Doermann, and D. DeMenthon, editors, *Video Mining*, volume 6 of *The Springer International Series in Video Computing*, pages 91–121. Springer US, 2003. → pages 13

[12] P. Dragicevic, G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh. Video browsing by direct manipulation. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI ’08, pages 237–246, New York, NY, USA, 2008. ACM. → pages 13

[13] S. M. Drucker, A. Glatzer, S. De Mar, and C. Wong. Smartskip: consumer level browsing and skipping of digital video content. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Changing our world, changing ourselves*, CHI ’02, pages 219–226, New York, NY, USA, 2002. ACM. → pages 12

[14] M. Fong, A. Al Hajri, G. Miller, and S. Fels. Casual authoring using a video navigation history. In *Proc. Graphics Interface*, May 2014. → pages 8

[15] A. Girgensohn, J. Boreczky, and L. Wilcox. Keyframe-based user interfaces for digital video. *Computer*, 34(9):61–67, 2001. → pages 9, 28

[16] C. Gkonela and K. Chorianopoulos. Videoskip: event detection in social web videos with an implicit user heuristic. *Multimedia Tools and Applications*, pages 1–14, 2012. 10.1007/s11042-012-1016-1. → pages 16

[17] D. B. Goldman, C. Gonterman, B. Curless, D. Salesin, and S. M. Seitz. Video object annotation, navigation, and composition. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST '08, pages 3–12, New York, NY, USA, 2008. ACM. → pages 13

[18] S. Greenberg and I. H. Witten. Supporting command reuse: Empirical foundations and principles. *International Journal of Man-Machine Studies*, 39:353–390, 1993. → pages 19

[19] T. Grossman, J. Matejka, and G. Fitzmaurice. Chronicle: capture, exploration, and playback of document workflow histories. In *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, UIST '10, pages 143–152, New York, NY, USA, 2010. ACM. → pages 21

[20] F. Guimbretiére, M. Dixon, and K. Hinckley. Experiscope: an analysis tool for interaction data. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '07, pages 1333–1342, New York, NY, USA, 2007. ACM. → pages 22

[21] J. Heer, J. Mackinlay, C. Stolte, and M. Agrawala. Graphical histories for visualization: Supporting analysis, communication, and evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1189–1196, Nov. 2008. → pages 21

[22] X.-S. Hua and S. Li. Interactive video authoring and sharing based on two-layer templates. In *Proceedings of the 1st ACM international workshop on Human-centered multimedia*, HCM '06, pages 65–74, New York, NY, USA, 2006. ACM. → pages 18

[23] D. Hupp and R. C. Miller. Smart bookmarks: automatic retroactive macro recording on the web. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, UIST '07, pages 81–90, New York, NY, USA, 2007. ACM. → pages 23

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

[25] W. Hurst and P. Jarvers. Interactive, dynamic video browsing with the zoomslider interface. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 4 pp.–, 2005. → pages 12

[26] J. Juran and A. Godfrey. *Juran's Quality Handbook*. Juran's quality handbook, 5e. McGraw Hill, 1999. → pages 5, 19

[27] T. Karrer, M. Weiss, E. Lee, and J. Borchers. Dragon: a direct manipulation interface for frame-accurate in-scene video navigation. In *Proceedings of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 247–250, New York, NY, USA, 2008. ACM. → pages 13

[28] D. Kimber, T. Dunnigan, A. Girgensohn, F. Shipman, T. Turner, and T. Yang. Trailblazing: Video playback control by direct object manipulation. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1015 –1018, july 2007. → pages 13

[29] D. Kurlander and S. Feiner. A history-based macro by example system. In *Proceedings of the 5th annual ACM symposium on User interface software and technology*, UIST '92, pages 99–106, New York, NY, USA, 1992. ACM. → pages 20

[30] I. Leftheriotis, C. Gkonela, and K. Chorianopoulos. Efficient video indexing on the web: A system that leverages user interactions with a video player. In *Proceedings of the 2nd International Conference on User-Centric Media (UCMEDIA)*, 2012. → pages 15

[31] F. C. Li, A. Gupta, E. Sanocki, L.-w. He, and Y. Rui. Browsing digital video. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, CHI '00, pages 169–176, New York, NY, USA, 2000. ACM. → pages 11, 47

[32] I. Li, J. Nichols, T. Lau, C. Drews, and A. Cypher. Here's what i did: sharing and reusing web activity with actionshot. In *Proceedings of the 28th international conference on Human factors in computing systems*, CHI '10, pages 723–732, New York, NY, USA, 2010. ACM. → pages 23

[33] R. Mertens, R. Farzan, and P. Brusilovsky. Social navigation in web lectures. In *Proceedings of the seventeenth conference on Hypertext and hypermedia*, HYPERTEXT '06, pages 41–44, New York, NY, USA, 2006. ACM. → pages 15

[34] T. Nakamura and T. Igarashi. An application-independent system for visualizing user operation history. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, UIST ’08, pages 23–32, New York, NY, USA, 2008. ACM. → pages 21

[35] K. A. Peker and A. Divakaran. Adaptive fast playback-based video skimming using a compressed-domain visual complexity measure. In *In IEEE International Conference on Multimedia and Expo*, pages 2055–2058, 2004. → pages 13

[36] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg. Abstracting digital movies automatically. Technical report, 1996. → pages 17

[37] D. A. Shamma, R. Shaw, P. L. Shafton, and Y. Liu. Watch what i watch: using community activity to understand content. In *Proceedings of the international workshop on Workshop on multimedia information retrieval*, MIR ’07, pages 275–284, New York, NY, USA, 2007. ACM. → pages 5, 15

[38] T. Syeda-Mahmood and D. Ponceleon. Learning video browsing behavior and its application in the generation of video previews. In *Proceedings of the ninth ACM international conference on Multimedia*, MULTIMEDIA ’01, pages 119–128, New York, NY, USA, 2001. ACM. → pages 14

[39] E. Tulving. Episodic and semantic memory 1. *Organization of Memory. London: Academic*, 381:e402, 1972. → pages 4

[40] S. Vihavainen, S. Mate, L. Seppälä, F. Cricri, and I. D. Curcio. We want more: human-computer collaboration in mobile social video remixing of music concerts. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI ’11, pages 287–296, New York, NY, USA, 2011. ACM. → pages 1, 3, 17

[41] H. Weda and M. Campanella. Use study on a home video editing system. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI...but not as we know it - Volume 2*, BCS-HCI ’07, pages 123–126, Swinton, UK, UK, 2007. British Computer Society. → pages 18

[42] J. Yew, D. A. Shamma, and E. F. Churchill. Knowing funny: genre perception and categorization in social video sharing. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI ’11, pages 297–306, New York, NY, USA, 2011. ACM. → pages 15

[43] B. Yu, W.-Y. Ma, K. Nahrstedt, and H.-J. Zhang. Video summarization based on user log enhanced link analysis. In *Proceedings of the eleventh ACM international conference on Multimedia*, MULTIMEDIA '03, pages 382–391, New York, NY, USA, 2003. ACM. → pages 15

[44] G. Zipf. Human behaviour and the principle of least-effort. Addison-Wesley, Cambridge, MA, 1949. → pages 5, 19