Locating Information in Video by Browsing and Searching

Andreas Girgensohn[1], Frank Shipman[2], John Adcock[1], Matthew Cooper[1], and Lynn Wilcox[1]

[1] FX Palo Alto Laboratory
3400 Hillview Ave. Bldg. 4
Palo Alto, CA 94304 USA
{andreasg,adcock,cooper,wilcox}@fxpal.com

[2] Department of Computer Science & Center for the Study of Digital Libraries
Texas A&M University
College Station, TX 77843-3112
shipman@cs.tamu.edu

Summary. This chapter describes tools for browsing and searching through video to enable users to quickly locate video passages of interest. Digital video databases containing large numbers of video programs ranging from several minutes to several hours in length are becoming increasingly common. In many cases, it is not sufficient to search for relevant videos, but rather to identify relevant clips, typically less than one minute in length, within the videos. We offer two approaches for finding information in videos. The first approach provides an automatically generated interactive multi-level summary in the form of a hypervideo. When viewing a sequence of short video clips, the user can obtain more detail on the clip being watched. For situations where browsing is impractical, we present a video search system with a flexible user interface that incorporates dynamic visualizations of the underlying multimedia objects. The system employs automatic story segmentation, and displays the results of text and image-based queries in ranked sets of story summaries. Both approaches help users to quickly drill down to potentially relevant video clips and to determine the relevance by visually inspecting the material.

1 Introduction

Locating desired video footage, as with other types of content, can be performed by browsing and searching. However, unlike browsing textual or image content, this task incurs additional time because users must watch a video segment to judge whether or not it is relevant for their task. This chapter describes browsing and searching interfaces for video that are designed to make it easier to navigate to or search for video content of interest.

Video is typically viewed linearly, and navigation tools are limited to fast forward and rewind. As a result, many approaches to summarize videos have

been proposed. One approach is to support skimming via playback of short versions of the videos [4, 13, 22]. Another approach is to support access to video segments via keyframes [27]. Finally, video libraries let users query for video segments with particular metadata, e.g., topic, date, length [14]. We are exploring two alternative approaches. The first approach utilizes interactive video to allow viewers to watch a short summary of the video and select additional detail as desired. Our second approach lets users search through video with text and image-based queries. This type of video search is difficult, because users need visual information such as keyframes or even video playback to judge the relevance of a video clip and text search alone is not sufficient to precisely locate a desired clip within a video program.

1.1 Detail-on-demand Video

We use detail-on-demand video as a representation for an interactive multi-level video summary. Our notion of detail-on-demand video has been influenced by interactive video allowing viewers to make choices during playback impacting the video they subsequently see. An example of interactive video is DVDs that include optional sidetrips that the viewer can choose to take. For example, when playing "The Matrix" DVD with optional sidetrips turned on, the viewer sees a white rabbit icon in the upper left corner of the display that indicates when a link may be taken. These links take the viewer to video segments showing how the scene containing the link was filmed. After the sidetrip finishes playing, the original video continues from where the viewer left off.

Our interactive multi-level video summary takes the form of a hypervideo comprising a set of video summaries of significantly different lengths and navigational links between these summary levels and the original video. Viewers can interactively select the amount of detail they see, access more detailed summaries of parts of the source video in which they are interested, and navigate throughout the entire source video using the summary. This approach explores the potential of browsing via hyperlinks to aid in the location of video content.

1.2 Video Search

For situations where browsing is not practical due to the size or semantic structure of the video collection, we describe a system for searching with text and image-based queries. While searching text documents is a well-studied process, it is less clear how to most effectively perform search in video collections. In text search it is typical for documents to be the unit of retrieval; a search returns a number of relevant documents. The user can then easily skim the documents to find portions of interest. In cases where documents are long, there are techniques to identify and retrieve only the relevant sections [24].

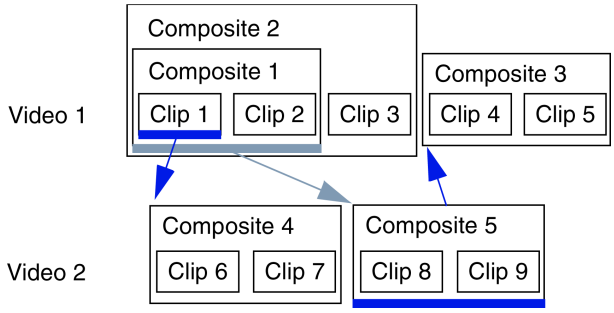

Fig. 1. Hierarchically-organized videos and links.

However, treating entire video documents or programs as units of retrieval will often lead to unsatisfactory results. After retrieving relevant programs, relevant clips, typically less than one minute of length, must be located within their source videos. Even when such videos are broken into sections, or stories of several minutes in length, it is still time-consuming to preview all the video sections to identify the relevant clips.

Our approach to this problem is to provide flexible user interfaces that present query results in a manner suited for efficient interactive assessment. Our target users are analysts who need both visual and textual information or video producers who want to locate video segments for reuse. While the latter will frequently use libraries that support retrieval with extensive meta-data describing properties such as location, included actors, time of day, lighting conditions, our goal is to enable search within video collections where such meta-data is not available. In this work, we assume that time-aligned text, such as transcripts, automatically recognized speech, or closed captions, is available.

The next section provides a conceptual overview of our approaches for detail-on-demand video and video search and compares these approaches to related work.

2 Overview

We support two approaches for the navigation of video collections, detail-on-demand video and and interactive video search. Both approaches have similarities to other approaches described in the literature but emphasize rich interfaces and intuitive interaction over complexity in analysis and interaction.

2.1 Detail-on-demand Video

Hypervideo allows viewers to navigate within and between videos. Applications of hypervideo include educational environments [10] and narrative storytelling [17]. General hypervideo allows multiple simultaneous link anchors

on the screen, e.g., links from characters on the screen to their biographies. This generally requires anchor tracking — tracking the movement of objects in the video that act as hotspots (source anchors for navigational links) [12, 20]. We have chosen to investigate detail-on-demand video as a simpler form of hypervideo, where at most one link is available at any given time. Such video can be authored in a direct manipulation video editor rather than requiring scripting languages or other tools that are unsuitable for a broad user base. At its simplest, the author selects a segment of the edited video for which a link will be active and the video sequence that will be shown if the viewer follows that link. By removing the need to define and track hot spots in video, the authoring and viewing interfaces can be simplified.

The name detail-on-demand comes from the natural affordances of this form of hypervideo to support gradual access to specific content at finer granularities. With the main video stream presenting the topic or process at a more abstract or coarser-grained level, the viewer can navigate to view clips on the topic of interest. This improvement can save the viewer's time in comparison to current linear access to videos or video summaries. The representation's primary features are navigational links between hierarchical video compositions and link properties defining link labels and return behaviors.

Detail-on-demand video consists of one or more linear video sequences with links between elements of these sequences (see Figure 1). Each video sequence is represented as a hierarchy of video elements. Segments of source video (clips) are grouped together into video composites, which themselves may be part of higher-level video composites. Links may exist between any two elements within these video sequences. The source element defines the source anchor for the link — the period of video playback during which the link is available to the viewer. The destination element defines the video sequence that will be played if the viewer follows the link. The source and destination elements specify both a start and an end time. To keep the video player interface simple, detail-on-demand video allows only one link to be attached to each element. For more information on this representation see [19].

A variety of interfaces for accessing video make use of an explicit or inferred hierarchy for selecting a starting point from which to play the video. These vary from the standard scene selection on DVDs to selection from hierarchically structured keyframes or text outlines in a separate window [16, 15]. Selecting a label or keyframe in a tree view is used to identify a point for playback.

The primary difference between interfaces supporting hierarchical access to video and detail-on-demand video is that the detail-on-demand viewer may request additional detail while watching the video rather than having to use a separate interface such as keyframes or a tree view. Also, the hierarchical representation of a video may not include semantics beyond simple hierarchical composition. The links in hypervideo have labels and a variety of behaviors for when the link's destination anchor finishes playback or when the user interrupts playback. There are two independent link behaviors to define: (1)

what happens when the destination sequence of a video link finishes playing, and (2) what happens when the viewer of the destination sequence ends its presentation before it is finished. The four options are (1) play from where the viewer left the video, (2) play from the end of source anchor sequence, (3) play from beginning of source anchor sequence, and (4) stop playback.

Links in hypervideo serve many of the same purposes they serve in hypertext. They provide opportunities for accessing details, prerequisite knowledge, and alternate views for the current topic. Unlike hypertext, the effectiveness of the hypervideo link is affected by link return behaviors, or what happens after the destination video has been watched or its playback aborted.

2.2 Video Search

Video search is currently of great interest, as evidenced by recently unveiled web-based video search portals by Yahoo [26] and Google [9]. 2004 marked the fourth year of the TRECVID [23] evaluations which draws a wide variety of participants from academia and industry. Some of the more successful ongoing efforts in the interactive search task draw upon expertise in video feature identification and content-based retrieval. The Dublin City University effort [5] includes an image-plus-text search facility and a relevance feedback facility for query refinement. The searcher decides which aspects of video or image similarity to incorporate for each query example. Likewise, the Imperial College interactive search system [11] gives the searcher control over the weighting of various image features for example-based search, provides a relevance feedback system for query refinement, and notably incorporates the NN^k visualization system for browsing for shots "close" to a selected shot. The MediaMill, University of Amsterdam system [21] is founded on a powerful semantic concept detection system allowing searchers to search by concept as well as keyword and example. Likewise the Informedia system from Carnegie Mellon University [3] incorporates their mature technology for image and video feature detection and puts the searcher in control of the relative weighting of these aspects.

Our effort is distinguished from others primarily by the simplicity of our search and relevance feedback controls in favor of an emphasis on rich interfaces and intuitive paths for exploration from search results. Our scenario is not so much one of query followed by refinement as it is query followed by exploration.

Whether explicitly stated or not, a goal in all of these systems is a positive user experience. That is, an informative and highly responsive interface cannot be taken for granted when handling thousands of keyframe images and tens of gigabytes of digital video.

The next two sections present two systems for browsing and searching video data. Section 3 presents a system that summarizes longer videos at multiple granularities. The summaries are combined to form detail-on-demand video enabling users to explore video and locate specific portions of interest.

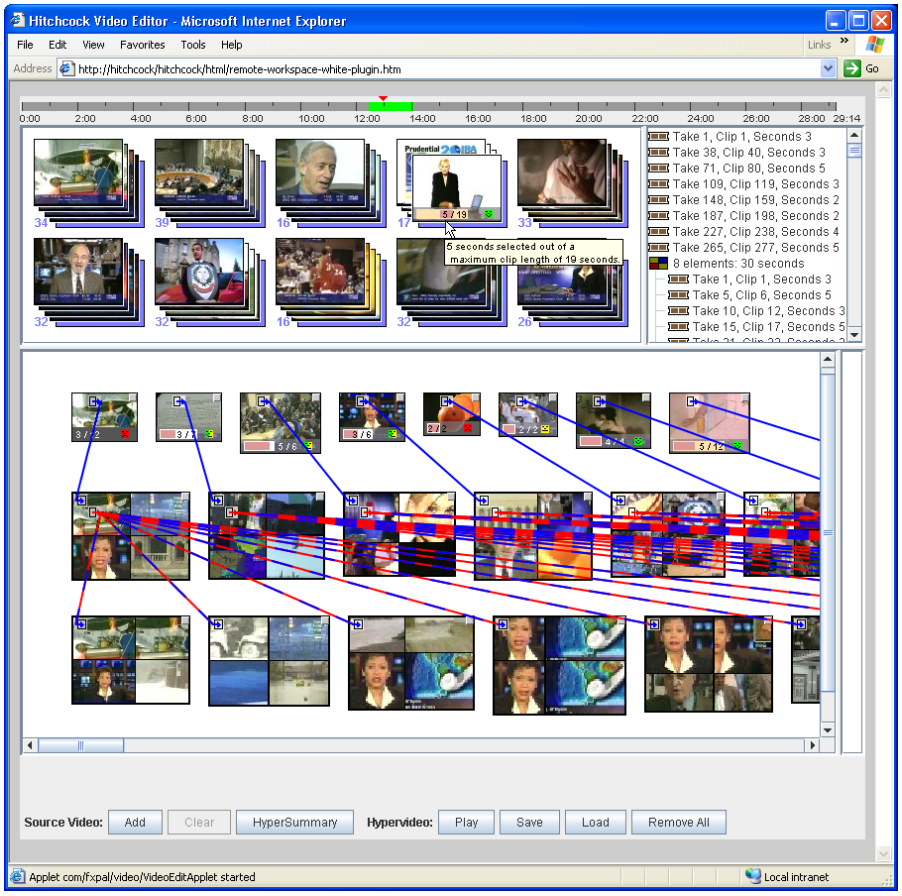

Fig. 2. An automatically generated hypervideo summary in Hyper-Hitchcock.

Section 4 presents our system for searching a video repository using text, image, and video queries. The repository's content is automatically organized at the program, story (i.e. topic), and shot levels enabling a powerful interface. We compare both approaches to related work. We summarize the paper in Section 5.

3 Hypervideo Summarizations

For many types of video, simply viewing the content from beginning to end is not desired. For example, when viewing an instructional video, you may want to skip much of the material and view only a topic of particular interest. Similarly, when viewing home video you may want to watch only particular

Fig. 3. Hypervideo player interface.

sections. Detail-on-demand videos can be structured as an interactive summary providing access into longer linear videos.

We created the Hyper-Hitchcock system as a means to author detail-on-demand videos. The Hyper-Hitchcock authoring interface (see Figure 2) allows the rapid construction of hierarchies of video clips via direct manipulation in a two-dimensional workspace. Navigational links can be created between any two elements in the workspace. A more detailed description of the Hyper-Hitchcock authoring interface can be found in [19].

Human authoring of such summaries is very time consuming and not cost effective if the summary will only be used a few times. We developed an algorithm for the automatic generation of hypervideo summaries composed of short clips from the original video. The interactive video summaries include linear summaries of different lengths and links in between these summaries and the entire source video.

The following sections describe the user interface for viewing hypervideo summaries, different clip selection techniques that are suitable for different types of video (e.g., home video or produced training video), and methods for placing links and setting link behaviors that improve interaction with the hypervideo summary.

The generation of the multi-level video summary includes three basic decisions: (1) how many levels to generate and of what lengths, (2) which clips from the source video to show in each level of the summary, and (3) what links to generate between the levels of the summary and the behaviors of these links.

3.1 Viewing Interface

We have created a series of viewing interfaces for Hyper-Hitchcock [8]. These interfaces vary in their complexity and the requirements they place on the player. The most recent, shown in Figure 3, is the result of a series of user studies [18].

We found that visualizing links in the timeline without labels or keyframes was confusing to users. However, putting link labels inside the timeline caused confusion between the actions of link following and skipping within the current video sequence. To address these issues, we placed keyframes for all links along the timeline and make all links available via keyframes and not just the link attached to the currently playing video segment. Users can follow the links by clicking on link keyframes or labels without having to wait for or move the video playback to the appropriate place. The keyframes for inactive links are reduced in size with faded link labels. The area of the active link, defined to be the innermost source anchor currently playing, is emphasized in the timeline and separators between links in the timeline are made stronger. The keyframe for the link under the mouse is also enlarged and the link is emphasized even more than the active link to indicate that mouse clicks will select it.

We display a stack of keyframes representing traversal history in the top-left of the player window. The keyframes for the older jump-off points are scaled down to enhance the history view. All keyframes are clickable, thus enabling the user to backtrack through several links at once.

3.2 Determining the Number of Summary Levels

The number and length of summary levels impact the number of links the user will have to traverse to get from the top-level summary to the complete source video. Having more levels provides users more control over the degree of summarization they are watching but also makes access to the original video less direct. To reduce the user's effort in navigation, a system can allow the user to traverse more than one link at once (moving more than one level deeper into the hypervideo).

Our current approach to determining the number of levels in the interactive summary is dependent on the length of the source video. For videos under five minutes in length, only one 30-second summary level is generated. For videos between 5 minutes and 30 minutes, two summary levels are generated — the first level being 30 seconds in length and the second being 3 minutes in length. For videos over 20 minutes, three summary levels are generated — one 30 seconds long, one three minutes long, and the last being one fifth the length of the total video to a maximum of 15 minutes. The length of the lowest summarization level is from one fifth to one tenth the length of the original video, except in cases of very short (less than two and a half minutes) or very long (more than 150 minutes) original videos.

This summarization ratio provides value over viewing a time-compressed presentation of the source video. Wildemuth and colleagues found significant performance drop-offs in comprehending a video when fast forward rates rose above 32–64 times normal speed [25]. For longer videos, our 30-second top-level summary provides speedups above 100-times with the option to see more detail for any interesting portion.

3.3 Segmenting Video into Takes and Clips

Our algorithms assume that the video has been first segmented into "takes" and "clips". For unproduced (home) video, takes are defined by camera on/offs. We do most of our work with DV format video that stores the camera on/off information, but when this metadata is absent we can apply automatic shot boundary determination (SBD) methods [6]. Clips are subsegments of takes generated by analyzing the video and determining good quality segments [7]. Here, good quality is defined as smooth or no camera motion and good lighting levels. We segment takes into clips in areas of undesirable quality such as fast camera motion and retain the highest-quality portion of the resulting clips.

For produced video, takes are defined as scenes and clips are the shots of the video. These are identified using well-known techniques [6, 28]. Sundaram and Chang [22] propose using consistency with regard to chromaticity, lighting, and ambient sound as a means for dividing a video into scenes. We are still experimenting with segmenting produced video but we plan to use a variation of published approaches.

3.4 Selecting Clips for Summary Levels

We have explored a number of algorithms for selecting clips from original source video. Selection of clips to use for each video summary is closely related to traditional video summarization. Unlike traditional summarization, selection of clips not only effects the quality of the generated linear summary but also impacts the potential for user disorientation upon traversing and returning from links.

We will describe three clip selection algorithms we have developed in this section. The first two approaches, the clip distribution algorithm and the take distribution algorithm, described below select clips based on their distribution in the video. They are geared for unproduced, or home video, where clips have been selected by their video quality. The third approach, the best-first algorithm, assumes that an external importance measure has been computed for the clips (shots) [4, 24]. This algorithm is more suitable for produced video.

The clip distribution algorithm is based on the identification of an array of m high-quality video clips via an analysis of camera motion and lighting. The current instantiation assumes an average length of each clip (currently 3.5 seconds) so the number of clips n needed for a summary is the length of the

summary in seconds divided by 3.5. The first and last clips are guaranteed to be in each summary with the remainder of the clips being evenly distributed in the array of potential clips. The use of an estimate of average clip length generates summaries of approximately the desired length rather than exactly the requested length. The algorithm can easily be altered to support applications requiring summaries of exact lengths by modifying in/out points in the selected clips rather than accepting the in/out points determined by the clip identification algorithm.

Applying the clip distribution algorithm will cause the resulting summary levels to include more content from takes that include more clips. This is appropriate when the camera is left filming while panning from activity to activity, such as walking from table to table at a picnic or reception. If shot divisions are not reflective of changes in activity then a take is likely to be over or underrepresented in the resulting summary. For example, a single take that pans back and forth between two areas of a single activity (people having a discussion or sides of the net in a tennis match) is likely to be overrepresented. Similarly, a single take consisting of several activities filmed back-to-back at a particular location (e.g., video of a classroom lecture) is likely to be underrepresented.

The take distribution algorithm uses the same segmentation of the video of length L into takes and clips but balances the representation of takes in the summary. For the first level, set a length L_1 (e.g., 30 seconds) and a clip length C_1 (e.g., 3 seconds) to pick $n = (L_1/C_1)$ clips. Check the centers of intervals of length L/n and include a clip from each of the takes at those positions. Pick the clip closest to the interval center. If more than one interval center hits the same take, pick the clip closest to the center of the take. If fewer than n clips are picked, look for takes that have not been used (because they were too short to be hit). Pick one clip from each of those takes starting with the clip that is furthest away from the already picked clips until n clips are picked or there are no more takes that have not been used. If still fewer than n clips are picked, pick an additional clip from each take in descending order of the number of clips in a take (or in descending order of take duration) until enough clips are picked. Continue picking three and more clips per take if picking two clips per take is insufficient. The same method can be used for the second level with lengths L_2 (e.g., 180 seconds) and clip length C_2 (e.g., 5 seconds).

The take distribution algorithm emphasizes the representation of each take in the summary and the distribution of selected clips throughout the duration of the source video. This approach will provide greater representation of short takes as compared to the clip distribution algorithm. This is advantageous when a consistent number of distinct activities are included in takes. It is likely to underrepresent takes in cases where many activities are included in some takes and one (or few) are included in others. Different application requirements will make one or the other algorithm more suitable. The take distribution algorithm will better represent the tennis match or conversation

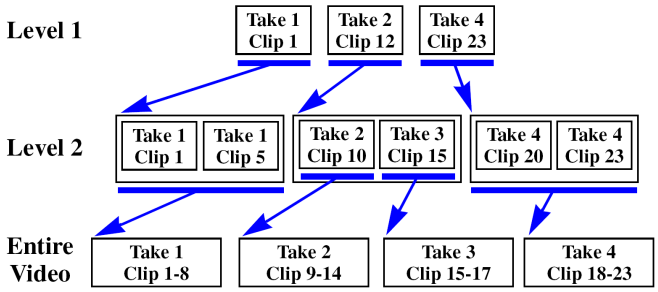

Fig. 4. Links generated by the simple take-to-take algorithm for a three-level video summary.

but the clip distribution algorithm better reflects the case of a camera moving between picnic tables.

Both of the above algorithms are designed to provide glimpses into the source video at somewhat regular intervals — with the first algorithm using the number of clips (or shots) as a measure of distance and the second algorithm using takes and playing time as a measure of distance. In contrast, the best-first algorithm for selecting clips uses an importance score for clips to select the most important video first. Importance scores for clips can be assigned automatically, using heuristics such as clip duration and frequency, as in [24]. Alternatively, scores can be assigned manually using the Hyper-Hitchcock interface. For example, in an instructional video, longer clips showing an entire process would be given a higher importance than shorter clips showing a particular subprocess in more detail. To generate a best-first summary, clips are added to the summary in order of their importance. This results in each level being a superset of higher levels (shorter) of the summary.

3.5 Placing Links Between Summary Levels

Once a multi-level summary has been generated, the next question is which links to generate between the levels. Links take the viewer from a clip at one level to the corresponding location in the next level below. In general, links are created in the multi-level summary for viewers to navigate from clips of interest to more content from the same period.

Generating links includes a number of decisions. A detail on demand video link is a combination of a source anchor, a destination anchor and offset, a label, and return behaviors for both completed and aborted playback of a destination. We will describe two variants for link generation — the simple take-to-take algorithm, and the take-to-take-with-offsets algorithm — to discuss trade-offs in the design of such techniques.

The simple take-to-take link generation algorithm creates at most one link per take in each level of the hypervideo. Figure 4 shows an example of a three-level summary created from 23 high-value clips identified in a four-take source

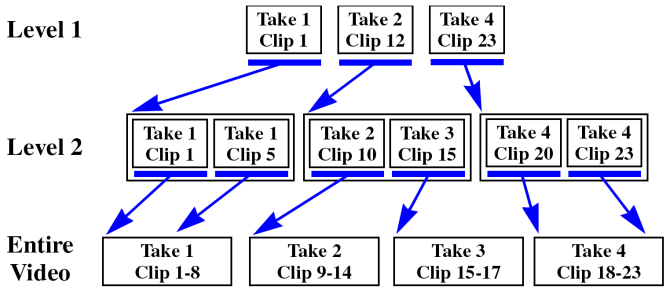

Fig. 5. Links generated by take-to-take-with-offsets algorithm for a three-level video summary.

video using this approach. All the clips from a particular take are grouped into a composite that will be a source anchor for a link to the next level. A composite in a higher-level summary (shorter) will be linked to the sequence of clips from the same take in the next level. If a take is not represented in a higher level, it will be included in the destination anchor for the link from the previous take. Otherwise, there would be no way to navigate to the clips from that take in the lower summary level. For example, let us consider the link from the middle clip in the top level of the summary shown in Figure 4. In this case, Clip 12 in Level 1 is the source anchor of the link. The destination anchor is a composite composed of Clip 10 and Clip 15. Clip 15, which is from Take 3, has been included because there was no clip from Take 3 in Level 1.

The take-to-take-with-offsets algorithm depicted in Figure 5 is similar to the simple take-to-take algorithm except that a separate link is generated for each clip in the source level of the summary. Clips are grouped into composites as in the above algorithm but links for clips other than the first clip representing a take in a particular level will include an offset to take the viewer to the (approximately) same point in the next level rather than returning to the first clip for that take in the next level.

The addition of more links is meant to keep users from having to rewatch video and to provide more precise temporal access into the video. The described method for setting offsets minimizes the "rewinding" that automatically occurs when a link is taken. If users do not realize they want material from this take until they view later clips, then they have to "rewind" the video by hand. An alternative method to setting offsets that is in between the no-offset approach of the simple take-to-take link generation and setting the offset to the same clip is to view the clip sequence as a timeline and take the user to the first clip that is closest to the clip being played. This would take the user to the midway point between the last available link and the start of the Clip being played when the link is selected.

The simple take-to-take link algorithm works well when the summary levels are being used as a table of contents to locate takes within the source video.

In this case, the user decides whether to get more information about a take while watching a montage of clips from that take. A difficulty with the simple take-to-take algorithm is that links to takes that include many activities will take the user to the take as a whole and not any particular activity. The take-to-take-with-offsets algorithm allows more precise navigation into such takes but assumes the user can make a decision about the value of navigation while viewing an individual clip.

Currently, link labels in the hypervideo summary provide information about the number of clips and length of the destination anchor. Algorithms that generate textual descriptions for video based on metadata (including transcripts) could be used to produce labels with more semantic meaning.

Link return behaviors also must be determined by the link generation algorithm. When the user "aborts" playback of the destination, the link returns to the point of original link traversal. To reduce rewatching video, the completed destination playback returns the user to the end of the source anchor (rather than the default of returning to the point of link traversal). Having links that return to the beginning of the source anchor could help user reorientation by providing more context prior to traversal but would increase the amount of video watched multiple times.

4 Video Search User Interface

Different approaches are required to search and browse larger or less structured video collections. A typical search in a moderate to large video collection can return a large number of results. Our user interface directs the user's attention to the video segments that are potentially relevant. We present results in a form that enables users to quickly decide which of the results best satisfy the user's original information need. Our system displays search results in a highly visual form that makes it easy for users to determine which results are truly relevant.

Initially the data is divided into programs, typically 20 to 30 minutes long; and video shots, which vary from 2 to 30 seconds. Because the frames of a video shot are visually coherent, each shot can be well represented by a single keyframe. A keyframe is an image that visually represents the shot, typically chosen as a representative from the frames in the shot [1]. A time-aligned transcript obtained through automatic speech recognition (ASR) is used to provide material by which to index the shots, but because shots and their associated transcript text are typically too short to be used as effective search units, we automatically groups shots into stories and use these as the segments over which to search. Adjacent shots with relatively high text-based similarity are grouped into stories. These stories form the organizing units under which video shots are presented in our interface. By grouping related shots together we arrive at transcript text segments that are long enough to form the basis for good keyword and latent semantic indexing. Because each story comprises

Fig. 6. The video search application interface. (A) Query results, (B) query terms and images, (C) the TRECVID topic text and example images, (D) thumbnail zoom and media player area, (E) timeline, (F) expanded shots, (G) selected shots.

several shots, it cannot be well represented by a single keyframe. Instead, we represent stories as collages of the underlying shot keyframes.

Figure 6 shows the interactive search interface. The user enters a query as keywords and/or images (Figure 6B). Keywords are typed and images are dragged into the query section from other parts of the interface. For the TRECVID interactive search task [23], the text, images, and clips describing the information need are displayed in Figure 6C. In this case, the user can select keywords and images from the topic description. Once the user has entered a query and pressed the search button, story results appear in Figure 6A, displayed in relevance order. The user can explore a retrieved story by clicking on the collage. The parent video is opened and the selected story is highlighted in the video timeline (Figure 6E). Below the timeline the keyframes from all the shots in the selected story are displayed (see Figure 6F). The shot or story under the mouse is magnified in the space in Figure 6D. A tool tip provides additional information for the shot or story under the mouse. The user drags any shots of interest to the area shown in Figure 6G to save relevant results. Another novel aspect of our system is that we mark visited stories so that the

Fig. 7. Story keyframe collage. 9 shots shown on the left are summarized by cropped sections of the 4 shots most relevant to the query "condoleeza rice".

user can avoid needless revisiting of stories. We present the three types of UI elements that we developed:

- Three visualizations provide different information perspectives about query results.
- Tooltips and magnified keyframes provide users with document information relevant to the query.
- Overlays provide cues about previously visited stories, current story and shot in video playback, and the degree of query relevance on story and shot.

4.1 Query Result Visualizations: Story Collage, Shot Keyframe, Video Timeline

As shown, query results are returned as a set of stories, sorted by relevance. A novel feature of our system is that retrieved stories are represented by keyframe collages where keyframes are selected and sized by their relevance to a query so that the same story may be shown differently for different queries. The size of the collage is determined by the relevance to the query so that one can see at a glance which stories are most relevant. We use a collage of four keyframes to indicate the different shots in a story without making the keyframes too small for recognizing details. We use rectangular areas for the keyframes for the sake of fast computation but we could instead use other collages such as a stained glass window visualization [2].

In addition to determining the relevance of stories with respect to the query, we also determine the relevance of each video shot. While the shot relevance does not provide good results on its own, it can be used to determine which shots within a story are the most relevant. The keyframes corresponding to the most relevant shots are combined to form a story keyframe-collage. The size allotted to each portion in this 4-image montage is determined by the shot's score relative to the query. Figure 7 shows an example of this where the query was "Condoleeza Rice" and the shots most relevant to the

Fig. 8. Timelines color-coded with query relevance.

query are allocated more room in the story thumbnail. In this case 3 of the 4 shots selected for the story collage chosen from the original 9 shots depict Condoleeza Rice. Rather than scaling down the keyframes to form the collage, they are cropped to preserve details in reduced-size representations. In the current implementation, the top-center portion of the cropped frame is used but a sensible alternative would be to crop around the main region-of-interest as determined by color and motion analysis or face detection.

Because the automatic story segmentation is not always accurate and related stories frequently are located in the same part of the video program, we provide easy access to the temporal neighborhood of the selected story. First, the timeline of the video containing the story color-codes the relevance of all stories in the video (see Figure 6E and Figure 8). This color-coding provides a very distinct pattern in the case of literal text search because only few stories contain the exact keywords. After a search using the latent semantic index (LSI), all parts of the timeline indicate some relevance because every term has some latent relationship to all other terms. We experimentally determined a nonlinear mapping of the relevance scores from LSI-based text search that highlights the most related stories without completely suppressing other potentially related stories. Immediately below the timeline in Figure 6E collages of neighboring stories around the selected story are displayed. This provides quick access to keywords in those stories via tool tips. By clicking on the timeline or the neighboring collages, the corresponding story can be expanded for closer inspection.

The keyframes for the shots comprising the selected story are shown in a separate pane (see Figure 6F and Figure 9). Double-clicking a keyframe plays the corresponding video shot. The expanded view provides access to the individual shots for playback, for adding them to the results, and for displaying information about the shots. One or more keyframes of shots can be dragged into or out of the result area to add or remove them from the

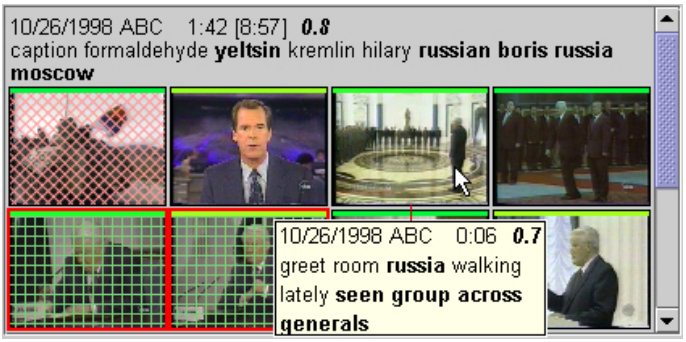

Fig. 9. Tool tip showing distinguishing keywords and bold query keywords.

result list, or into or out of the image search area to add or remove them from the image search. Shots can also be marked explicitly as irrelevant.

4.2 Feedback on Document Relevance: Tooltips and Magnified Keyframes

It is useful to provide feedback to the user to indicate why a particular document was deemed relevant to the query and how the document is different from others in the collection. Tooltips for story collage and video shot keyframes provide that information to the user in the form of keywords that are distinctive for the story and keywords related to the query (see the plain text in Figure 9). Terms that occur frequently in the story or shot and do not appear in many other stories or shots are most distinguishing. While words such as "lately" do not really help in distinguishing the video passage from others, words such as "russia" are helpful.

The terms in bold are most related to the query and indicate why the document is relevant to the query. We decided against displaying the terms with surrounding text as is frequently done in Web search engines. The reason is that we do not want the tooltips to be too large. Furthermore, the automatic speech recognition makes mistakes that are more noticeable when displaying whole phrases. By displaying five keywords from each of the two categories, it is likely that at least one or two are truly useful.

With a literal text search approach, the terms most related to the query are the query terms appearing in the story. When the latent semantic index is used for search, a relevant document may not contain any of the query terms but only terms that are closely related. We use the latent semantic index to identify terms in the document that are most similar to the query.

In an earlier version of our application, we displayed keyframes as part of the tooltips. Users interacting with that version of the application found that the keyframes were either too small to be useful or that the tooltips covered up too much the window. To address this issue, we reuse the video player area

as a magnifier for the keyframe under the mouse or the selected keyframe (see Figure 6D). Usually, the video player will be stopped while the user inspects keyframes so that the user can see a magnified version of the keyframe or collage without the need to dedicate some window area to that purpose.

4.3 Overlay Cues: Visited Story, Current Playback Position, Query Relevance

Semi-transparent overlays are used to provide three cues. A gray overlay on a story icon indicates that it has been previously visited (see Figure 6A and E). A translucent red overlay on a shot icon indicates that it has been explicitly excluded by the user from the relevant shot set. A translucent green overlay on a shot icon indicates that it has been included in the results set (see Figure 6F). Figure 9 shows the use of patterns instead of translucent overlays for color-blind users and grayscale reproduction of the image. Red diagonal lines indicate exclusion and green horizontal and vertical lines indicate inclusion.

While video is playing, the shot and the story containing the current playback position are indicated by placing a red dot on top of their keyframes. The playback position is also indicated in the timeline by a vertical red line.

Horizontal colored bars are used along the top of stories and shots to indicate the degree of query-relevance, varying from black to bright green. The same color scheme is used in the timeline depicted in Figure 8.

5 Conclusions

In this chapter, we presented two approaches enabling users to quickly identify passages of interest within a potentially large collection of videos. The first approach automatically generates an interactive multi-level summary in the form of a hypervideo. It lets users watch short video clips and request more detail for interesting clips. Two goals for the design of hypervideo summaries are to minimize user disorientation resulting from link navigation and to minimize the rewatching of video segments. These goals are sometimes in conflict. Playing the same clip multiple times can be used to provide greater context and thus reduce disorientation. Clip selection and link generation algorithms interact in determining the degree to which a hypervideo generation approach will meet these goals.

The second approach allows users to search large video collections. We process the linear text transcript associated with the videos and segment it into stories to create three levels of segmentation (program, story, shot). Visualization techniques take advantage of the segmentation to draw the users' attention to promising query results and support them in browsing and assessing relevance. Rather than relying on elaborate media analysis techniques, we have used simple and proven automatic methods. We integrate the analysis

results with an efficient user interface to enable users to both quickly browse retrieved video shots and to determine which are relevant.

Modern tools for browsing and searching video have the potential to dramatically improve access into large video libraries and to aid location of desired content within long videos. The results from Hyper-Hitchcock and our interactive video search application help establish the basis for the next generation of access tools combining searching and browsing of video.

References

1. J. S. Boreczky and L. A. Rowe. Comparison of video shot boundary detection techniques. In *Storage and Retrieval for Image and Video Databases (SPIE)*, pages 170–179, 1996.
2. P. Chiu, A. Girgensohn, and Q. Liu. Stained-Glass visualization for highly condensed video summaries. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo, ICME 2004*, pages 2059–2062. IEEE, Jun. 2004.
3. M. Christel, J. Yang, R. Yan, and A. Hauptmann. Carnegie mellon university search. In *TREC Video Retrieval Evaluation Online Proceedings*, 2004.
4. M. G. Christel, M. A. Smith, C. R. Taylor, and D. B. Winkler. Evolving video skims into useful multimedia abstractions. In *CHI '98: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 171–178, New York, NY, USA, 1998. ACM Press/Addison-Wesley Publishing Co.
5. E. Cooke, P. Ferguson, G. Gaughan, C. Gurrin, G. J. F. Jones, H. Le Borgue, H. Lee, S. Marlow, K. McDonald, M. McHugh, N. Murphy, N. E. O'Connor, N. O'Hare, S. Rothwell, A. F. Smeaton, and P. Wilkins. Trecvid 2004 experiments in dublin city university. In *TREC Video Retrieval Evaluation Online Proceedings*, 2004.
6. M. Cooper. Video segmentation combining similarity analysis and classification. In *MULTIMEDIA '04: Proceedings of the 12th annual ACM international conference on Multimedia*, pages 252–255, New York, NY, USA, 2004. ACM Press.
7. A. Girgensohn, J. Boreczky, P. Chiu, J. Doherty, J. Foote, G. Golovchinsky, S. Uchihashi, and L. Wilcox. A semi-automatic approach to home video editing. In *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pages 81–89, New York, NY, USA, 2000. ACM Press.
8. A. Girgensohn, L. Wilcox, F. Shipman, and S. Bly. Designing affordances for the navigation of detail-on-demand hypervideo. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces*, pages 290–297, New York, NY, USA, 2004. ACM Press.
9. Google video search. http://video.google.com.
10. N. Guimãres, T. Chambel, and J. Bidarra. From cognitive maps to hyper-video: Supporting flexible and rich learner-centred environments. *Interactive Multimedia Journal of Computer-Enhanced Learning*, 2((2)), 2000. http://imej.wfu.edu/articles/2000/2/03/.
11. D. Heesch, P. Howarth, J. Megalhaes, A. May, M. Pickering, A. Yavlinsky, and S. Ruger. Video retrieval using search and browsing. In *TREC Video Retrieval Evaluation Online Proceedings*, 2004.

12. K. Hirata, Y. Hara, H. Takano, and S. Kawasaki. Content-oriented integration in hypermedia systems. In *HYPERTEXT '96: Proceedings of the the seventh ACM conference on Hypertext*, pages 11–21, New York, NY, USA, 1996. ACM Press.
13. R. Lienhart. Dynamic video summarization of home video. In *Storage and Retrieval for Media Databases*, volume 3972 of *Proceedings of SPIE*, pages 378–389, 2000.
14. G. Marchionini and G. Geisler. The open video digital library. *D-Lib Magazine*, 8(12), 2002.
15. B. A. Myers, J. P. Casares, S. Stevens, L. Dabbish, D. Yocum, and A. Corbett. A multi-view intelligent editor for digital video libraries. In *JCDL '01: Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, pages 106–115, New York, NY, USA, 2001. ACM Press.
16. Y. Rui, T. S. Huang, and S. Mehrotra. Exploring video structure beyond the shots. In *International Conference on Multimedia Computing and Systems*, pages 237–240, 1998.
17. N. Sawhney, D. Balcom, and I. Smith. Hypercafe: narrative and aesthetic properties of hypervideo. In *HYPERTEXT '96: Proceedings of the the seventh ACM conference on Hypertext*, pages 1–10, New York, NY, USA, 1996. ACM Press.
18. F. Shipman, A. Girgensohn, and L. Wilcox. Hypervideo expression: experiences with hyper-hitchcock. In *HYPERTEXT '05: Proceedings of the sixteenth ACM conference on Hypertext and hypermedia*, pages 217–226, New York, NY, USA, 2005. ACM Press.
19. F. M. Shipman, A. Girgensohn, and L. Wilcox. Hyper-Hitchcock: Towards the easy authoring of interactive video. In *Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction*. IOS Press, Sep. 2003.
20. J. M. Smith, D. Stotts, and S.-U. Kum. An orthogonal taxonomy for hyperlink anchor generation in video streams using OvalTine. In *HYPERTEXT '00: Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 11–18, New York, NY, USA, 2000. ACM Press.
21. C. G. M. Snoek, M. Worring, J. M. Geusebroek, D. C. Koelma, and F. J. Seinstra. The MediaMill TRECVID 2004 semantic video search engine. In *TREC Video Retrieval Evaluation Online Proceedings*, 2004.
22. H. Sundaram and S.-F. Chang. Condensing computable scenes using visual complexity and film syntax analysis. In *Proceedings of the 2001 IEEE International Conference on Multimedia and Expo, ICME 2001*. IEEE Computer Society, Aug. 2001.
23. TREC video retrieval evaluation. http://www-nlpir.nist.gov/projects/trecvid/.
24. S. Uchihashi, J. Foote, A. Girgensohn, and J. Boreczky. Video Manga: generating semantically meaningful video summaries. In *MULTIMEDIA '99: Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 383–392, New York, NY, USA, 1999. ACM Press.
25. B. M. Wildemuth, G. Marchionini, M. Yang, G. Geisler, T. Wilkens, A. Hughes, and R. Gruss. How fast is too fast?: evaluating fast forward surrogates for digital video. In *JCDL '03: Proceedings of the 3rd ACM/IEEE-CS joint conference on Digital libraries*, pages 221–230, Washington, DC, USA, 2003. IEEE Computer Society.
26. Yahoo! video search. http://video.search.yahoo.com.

27. M. M. Yeung and B.-L. Yeo. Video visualization for compact presentation and fast browsing. *IEEE Transactions on Circuits and Systems for Video Technology*, 7(5):771–785, Oct. 1997.
28. H. J. Zhang, S. Y. Tan, S. W. Smoliar, and G. Yihong. Automatic parsing and indexing of news video. *Multimedia Syst.*, 2(6):256–266, 1995.