Generation of Interactive Multi-Level Video Summaries

Frank Shipman, Andreas Girgensohn, Lynn Wilcox
FX Palo Alto Laboratory, Inc.
3400 Hillview Avenue
Palo Alto, CA 94304, USA
{shipman, andreasg, wilcox}@fxpal.com

ABSTRACT

In this paper, we describe how a detail-on-demand representation for interactive video is used in video summarization. Our approach automatically generates a hypervideo composed of multiple video summary levels and navigational links between these summaries and the original video. Viewers may interactively select the amount of detail they see, access more detailed summaries, and navigate to the source video through the summary. We created a representation for interactive video that supports a wide range of interactive video applications and Hyper-Hitchcock, an editor and player for this type of interactive video. Hyper-Hitchcock employs methods to determine (1) the number and length of levels in the hypervideo summary, (2) the video clips for each level in the hypervideo, (3) the grouping of clips into composites, and (4) the links between elements in the summary. These decisions are based on an inferred quality of video segments and temporal relations those segments.

Categories and Subject Descriptors

H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems – *video.* H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia – *navigation, user issues.*

General Terms

Algorithms, Design, Experimentation, Human Factors.

Keywords

Hypervideo, video summarization, link generation, video editing.

1. INTRODUCTION

Finding the desired information in video takes time. This is because typical video must be viewed linearly, and tools for searching are limited to fast forward and rewind. As a result, there have been many approaches for summarizing video. One approach for providing more rapid access is to support skimming via shorter versions of the videos [2, 7, 15]. Another approach is to support access to pieces of video via keyframes [18]. Video libraries let users query for pieces of video with particular metadata, e.g., topic, date, length [8]. We are exploring an alternative method that employs interactive video to support viewers in watching a short summary of the video and in selecting additional detail on demand.

Our notion of *detail-on-demand video* has been influenced by interactive video that makes it possible for people viewing the video to make choices that impact what video they see. An example of interactive video are 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 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.

We use detail-on-demand video as a representation for an interactive multi-level video summary. This takes the form of a hypervideo comprising multiple video summaries 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 to the entire source video through the summary. For automatically generating such summaries, video clips have to be selected for the different levels of the summary such that they both summarize the video at the appropriate level and that they provide good link anchors and destinations. We incorporated these automatically generated video summaries into Hyper-Hitchcock [12], an editor and player for detail-on-demand video. Authors can refine generated summaries in the editor to improve the quality of the summary.

We briefly described an early technique for generating detail-on-demand video summaries in [11]. In this paper we present several improved techniques for selecting video clips to be included in different summary levels that are suitable for different types of video (e.g., home video or produced training video). We also describe different methods for placing links and setting link behaviors that improve interaction with the hypervideo summary.

The next section describes the detail-on-demand video model. This is followed by a description of the automated generation of detail-on-demand video for summarization and access purposes and a discussion of the related work. We then describe Hyper-Hitchcock, with emphasis on its support for authoring hierarchically organized streams of video and links among the elements of these streams.

MM'03, November 2-8, 2003, Berkeley, California, USA.

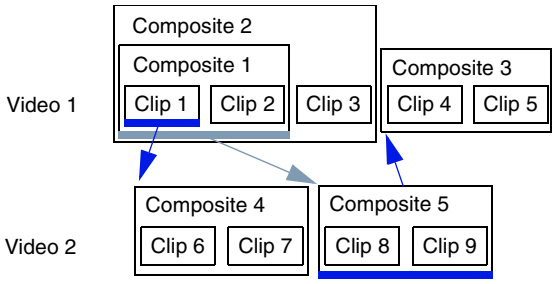

Figure 1: Hierarchically-organized videos and links

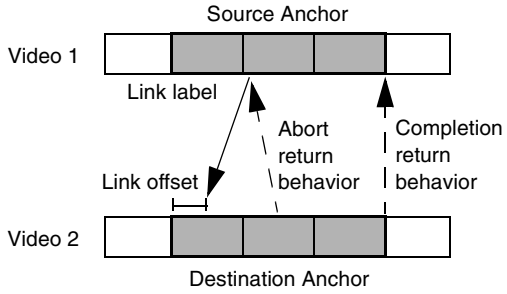

Figure 2: A Detail-on-Demand video link includes source and destination anchors, link label, and return behaviors.

2. DETAIL-ON-DEMAND VIDEO

Hypervideo allows viewers to navigate between video chunks. General hypervideo allows multiple simultaneous link anchors on the screen, e.g., links from actors on the screen to their biographies [4, 6, 13]. We have concentrated on a simpler form of hypervideo, *detail-on-demand* video, where at most one link is available at any given time. The representation's primary features are navigational links between hierarchical video compositions and link properties defining link labels and return behaviors. Authoring such video can be supported in a direct manipulation video editor rather than requiring scripting languages or other complicated tools that are unsuitable for a broad user base. This representation creates a natural mechanism for authoring "how to" videos where viewers can get the level of explanation they need. With the main video stream presenting the topic/process at a more abstract or courser-grained level, the viewer navigates to view the aspects of the topic for which they need more help. We previously discussed several applications for detail-on-demand video [12]. In this paper, we focus on its utility as a video summary.

2.1 Hierarchical Video with Links

In detail-on-demand video, 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 takes the link. The source and destination elements specify both a start and an end time.

Figure 1 shows three links between two video sequences. The links are from Clip 1 to Composite 4, Composite 1 to Composite 5, and from Composite 5 to Composite 3. Also, at most one link is made available to the viewer at any time. If multiple levels of the hierarchy have links that overlap, the viewer is provided with the most specific link — that is the link attached to the innermost element in the hierarchy. In the example in Figure 1, viewers see the link out of Clip 1 to Composite 4 at the beginning of Video 1 during Clip 1 and the link from Composite 1 to Composite 5 during Clip 2. Such link patterns occur when a more focused topic occurs in the context of a more general discussion — e.g., a link to more video on Baghdad within a longer link to more on Iraq.

2.2 Link Labels and Behaviors

Links in detail-on-demand video have a number of characteristics that impact the resulting presentation. Three such characteristics are the offset into the destination where playback will begin, the label shown to the viewer as an indicator of the content of the link, and the behavior when returning from links (see Figure 2). There are two independent link return behaviors: (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. When a sequence finishes playing, the four options are (1) play from where the viewer left the video, (2) play from the end of the source anchor sequence, (3) play from beginning of the source anchor sequence, and (4) stop playback. Playing from the end of the source anchor is the "Completion return behavior" for the link in Figure 2. The second link return behavior is determining what to do when the viewer decides to return early from a link destination. In hypertext, like the Web, pressing the "back" button takes the user back to where they last took a navigational link. Similarly, the "return" button in detail-on-demand video would most often return to the location of link navigation, as in Figure 2. In addition to this "intuitive" behavior, the same options for where to restart playback of the source anchor apply as above.

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 the topic of interest. Similarly, when viewing home video you may want to watch only the more interesting sections. Detail-on-demand videos can be structured as an interactive summary providing access into longer linear videos. Human authoring of such summaries is very time consuming and not cost effective if the summary will only be used a few times. We have been exploring design decisions in the automatic generation of hypervideo summaries composed of short clips from the original video. The interactive video summaries are generated to include several linear summaries of different lengths and links in between these summaries and the entire source video.

The techniques described here are improvements over a technique for generating detail-on-demand video summaries we described previously [11]. We now have several different clip selection techniques that are suitable for different types of video (e.g., home

video or produced training video). We also determined methods for placing links and setting link behaviors that improve interaction with the hypervideo summary. For example, we introduced the concept of link offsets because we wanted to give users the option to move to an earlier point in a link destination to see additional context. We also found that our initial approach does not work well for produced video (e.g., movies) and started developing alternative summarization techniques for such material. In addition, we developed design guidelines for choosing among the different clip selection and link generation techniques.

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 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 Determining Number and Length of Summary Levels

The decision about the number and length of summary levels impacts 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 users effort in navigation, a system can include the ability 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. 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 ratio (5x to 10x) is to provide value to summarization over viewing a time-compressed presentation of the source video. Wildemuth and colleagues found significant performance drop offs in comprehending video when fast forward rates rose above 32-64x normal speed [17]. For longer videos, our 30-second top-level summary provides speedups above 100x with the option to see more detail for any interesting portion.

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.

3.2 Segmenting Video into Takes and Clips

Our algorithms assume that the video has been first segmented into "takes" and "clips". For un-produced (home) video, takes are defined by camera on/offs. We do most of our work with DV video that stores the camera on/off information. Clips are sub-segments of takes generated by analyzing the video and determining good quality segments [3]. Here, good quality is defined by 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.

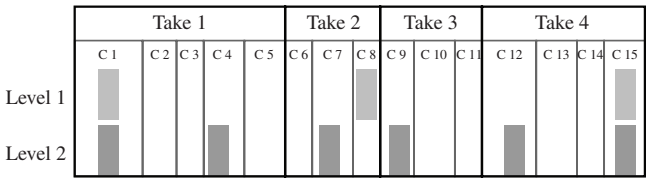

Figure 3: Selecting clips for two-level video summary.

For produced types of video, takes are defined as scenes and clips are the shots of the video. These are identified using well-known techniques [19]. Sundaram and Chang [14] 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.

3.3 Selecting Clips for Summary Levels

We have explored a number of potential 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 then discuss trade-offs between these designs within the context of hypervideo generation in Section 3.5.1. 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 un-produced, 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) [2, 16]. 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 clip are guaranteed to be in each summary with the remainder of the clips being evenly distributed in the array of potential clips. Thus, our method selects one clip every $(m - 1) / (n - 1)$ potential clips. Figure 3 shows how two levels of a summary are selected from 15 high-value clips identified in a four-take source video. 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 selection algorithm.

Applying the clip distribution algorithm to video will cause the resulting summary levels to include more content for takes that include more clips. This is appropriate when the camera is left filming while panning from activity or activity, such as walking

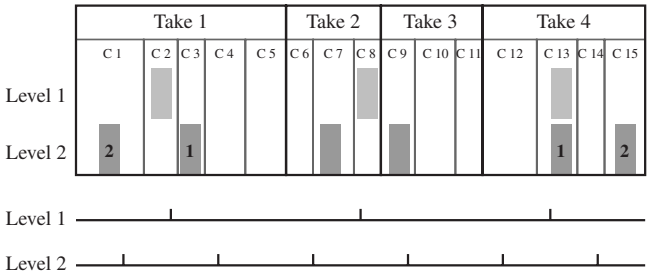

Figure 4: Alternative for selecting clips for two-level video summary.

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. 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 takes at those positions (see the bottom of Figure 4 for time lines with the centers of 3 and 6 intervals, respectively). 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 (see the clips labeled "2" in Figure 4 for examples of clips picked by this method). 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). Figure 4 shows how the alternative algorithm selects clips from the same source video as shown in Figure 3. Because the takes and clips are of fairly even lengths, both algorithms produce similar results.

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 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 [16]. Alternatively, scores can be assigned manually using the HyperHitchcock 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 sub-process 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.

3.4 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 lower level. 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 5 shows an example of a three-level summary created from 23 high-value clips identified in a four-take source 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 5. In this case, Clip 12 in Level 1 is the source anchor of the link. The destination

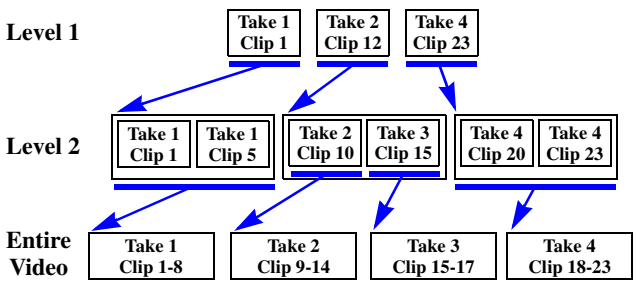

Figure 5: Links generated by the simple take-to-take algorithm for a three-level video summary.

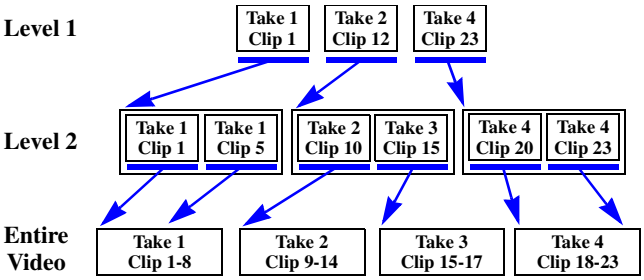

Figure 6: Links generated by the take-to-take with offsets algorithm for a three-level video summary.

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 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. Figure 6 shows the links generated for the same two summary levels from Figure 5. Links from the first level are identical to the earlier algorithm but now there are two additional links from Level 2 to the source video: Clip 5 now links to Take 1 with an offset to Clip 5 in the source video and Clip 23 now links to Take 4 with an offset to Clip 23 in the source video.

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 mid-way point between the last available link and the start of the Clip being played when the link is selected. For example, the links with offsets in Figure 6 would go from Clip 5 in Level 2 to Clip 3 in the source video and from Clip 23 in Level 2 to Clip 21 in the source video.

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.

Clip and take relation to activities	Impact on algorithm selection
single clip per take, single activity per take	all algorithms work equally well
single activity divided into multiple clips (e.g., the tennis video)	**Clip selection:** clip distribution overrepresents activity take distribution works well **Link generation:** simple take-to-take works well take-to-take with offset generates non-intuitive detail
multiple activities in take, activities well represented by clips (e.g., take of moving from table to table)	**Clip selection:** clip distribution works well take distribution underrepresents take **Link generation:** simple take-to-take inefficient take-to-take with offset works well
multiple activities grouped into single clip (e.g., the classroom video)	**Clip selection:** clip distribution and take distribution underrepresent take **Link generation:** simple take-to-take and take-to-take with offset inefficient

Table 1. Clip identification impact on clip selection and link generation algorithms.

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.

3.5 Discussion

The above descriptions of algorithms for generating hypervideo summaries include some discussion of trade-offs between different approaches. In addition there are interactions between the clip selection and link generation algorithms chosen for particular stages of hypervideo generation. Depending on how the activities in a video are represented as clips and takes, different algorithms produce better results. Table 1 shows criteria for when the different clip selection and link generation algorithms are most applicable. While un-produced home or training video can fall into any category of that table, produced video will mostly fall into the second category (single activity divided into multiple clips). This is due to the fact that a scene usually depicts a single activity (e.g., a conversation) consisting of multiple shots (e.g., cutting back and forth among the participants in the conversation).

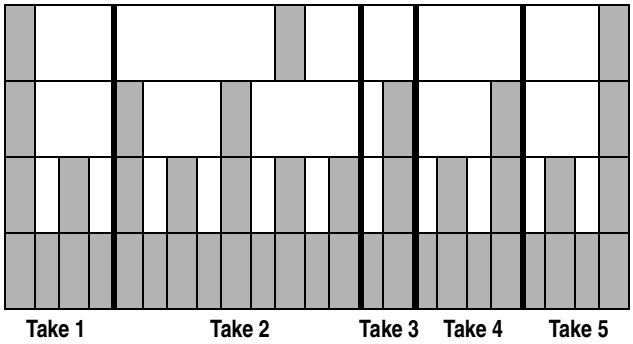

Figure 7: Timeline of clips contained in each summary level.

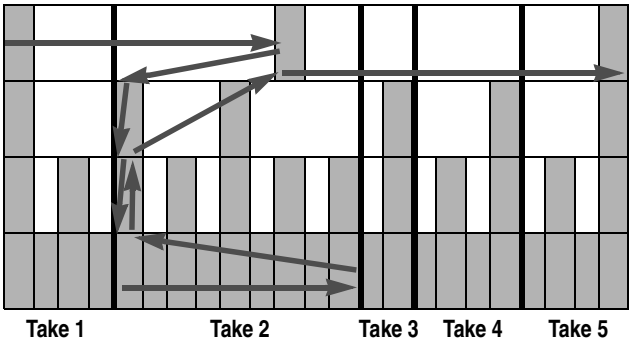

Figure 8: Timeline of clips contained in each summary level.

3.5.1 Clip Selection and Link Generation

Our existing algorithms for clip selection and link generation work serially, with clip selection not being influenced by link generation. But clip selection impacts the disorientation caused by link traversal in the summary.

One potential source of disorientation is that, in the case of the first two clip selection algorithms, more detailed summary levels may not have content included less detailed levels. Figure 7 presents the timeline of an example multilevel summary of a five take source video. Shaded regions indicate clips selected to be included in each level. The second (middle) clip in the top level of the hypervideo is not included in the next level of the summary. Users taking links in such a case may not be able to relate where they end up by taking the link to where they were when they took the link. Our take-based linking algorithm will cause the user to jump from this clip back to the first clip of Take 2. An algorithm that creates links to the closest clip (minimal temporal distance) would take the user to the prior clip in the source material in this case. Jumping about in the timeline is less of a concern for users reviewing video that they shot since the clips shown are acting as reminders into their memory of the material in the source video.

Even if clips are selected such that higher levels are subsets of the lower levels, as is the case in the second through fourth levels in Figure 7, there can be links that take the user back in time (from a later clip to an earlier one.) These occur when a higher-level summary does not include earlier clips in the next level of the summary or in the source video. The clip from Take 4 in the second level in Figure 7 is such a case. Taking the link will cause an earlier clip in Take 4 to begin playing. This results in users potentially viewing the same material more than once and an increase in disorientation since the user may expect to start at the same clip in the next level. Such links will occur with all three clip selection algorithms described. Only guaranteeing that the first clip of each take represented in a summary level is included in that level will remove this behavior. Even then, because the portion of clips included in the summary levels is the highest-quality segment, jumping from the last summary level to the source video will often cause playback to begin from content not seen by the user. This is the case for Takes 3 through 5 in Figure 7.

The third clip selection algorithm guarantees longer levels are a superset of the shorter levels. This can be used to help in reducing jumps back in time and disorientation by always including the first clip of the source video and modifying the link generation algorithm to link from the currently playing clip to the same clip in the next level summary or source. This results in a hypervideo that ignores take boundaries unless the goodness value of the first clip of each take is guaranteed to have the highest goodness value of all clips for that take.

3.5.2 Composite Generation and Link Return Properties

All of the above algorithms use take boundaries for grouping clips into composites. These composites become the destinations of links from higher levels in the hypervideo summary. Following a link causes the composite to begin playing, potentially from some offset in the middle of the composite. In this design, the user returns to the higher level of the summary after watching some or all of the composite. But where do they return? As previously described, link return properties allow different behaviors for when the user aborts the playback by pressing the “back” button and when the composite finishes playback and returns automatically.

The current approach sets the abort behavior to return to the point of link departure and the completion behavior to return to the end of the link source. When the source composite is short, these choices provide intuitive interactions. But as link sources and destinations get longer, either because more clips are grouped together into a composite or because there are long takes in the source video, the abort behavior of returning to the point of departure will cause more rewatching of video. For example, when a user navigates down through the hierarchy and watches a significant number of clips from a single take and then presses “back”, they will be reshown all the clips included in the higher level summary that they just watched. The arrows in Figure 8 show the video playback sequence of a user navigating down through the second clip in the top level to watch all of Take 2 in the source video and then deciding to abort playback when it gets to Take 3. In this case the user will be taken back to the point of link navigation in the previous layers of the hypervideo.

Approaches to reducing this effect of large composites involve the extension of the detail-on-demand video representation. The first, simpler approach is to assume that the link’s source and destination anchors represent the same content at the same ratios of duration. With this assumption, a return behavior which returns to the same percentage offset in duration will approximately get the viewer to

the shorter version of the same content. For example, if 45 seconds of a 60 second destination composite has been watched, the user would be returned to the 9 second point in a 12 second source anchor. When composites are made up of few clips, the assumption is not correct and may cause greater confusion due to the seemingly random behavior. The assumption is more accurate with composites including more component clips.

Another approach to returning within large composites is to include markers in the destination video which set the position that the user will return to. For example, if a clip is represented in both the source and destination anchors, watching the clip in the destination will cause the return location to be at the end of this clip is in the source. This level of detailed specification is unlikely for any human author but can be generated by the link generation algorithm. With such a behavior, the destination composite for all links could become the entire next level of the hypervideo summary. In such a case, links would take the user to the more detailed view of the material they are currently viewing using the link offsets. Also, they would only return to a higher level of the summary through the "back" button, which would take them to the location in the shorter summary representing the same content from the source video.

4. RELATED WORK

Authoring interactive video is a relatively new area of research. While there are a number of hypervideo authoring tools, DVD authoring tools, and nonlinear video authoring tools, there is little work on automatically generating hypervideo. The closest areas of research are (1) the generation of linear video summaries, (2) hierarchy-based interfaces for accessing video, and (3) link generation for video.

4.1 Linear Video Summarization

One approach for providing more rapid access is to support skimming via shorter versions of the videos [2, 5, 7, 15]. Linear summaries of video are generated by a wide variety of applications. While the methods used to generate these summaries are of interest for generating individual levels of the hypervideo summary, these efforts do not include the generation of multiple summaries and the generation of links between these summaries. The key difference between linear and interactive multi-level summaries is that users can request additional detail for parts of the video rather than being restricted to a predetermined level of detail.

4.2 Hierarchic Interfaces to Video

Another approach is to support access to pieces of video via keyframes [18]. Video libraries let users query for pieces of video with particular metadata, e.g., topic, date, length [8]. 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 [9, 10]. Selecting a label or keyframe in a tree view is used to select a point for playback.

A difference between interfaces supporting hierarchical access to video and detail-on-demand video is 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.

Figure 9: Player with Keyframes, Navigation Buttons, and Link Indicators in Timeline

Also, the hierarchical video representation of these tools does not include semantics beyond simple hierarchical composition. 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. Links between clips or composites in a hypervideo support the viewing of additional detail and the automatic return to the main video thread.

4.3 Automatic Linking of Video

There are a few hypervideo link generation algorithms that are loosely related to our approach. Most of these are aimed at generating links between video about common content as recognized by some (semi-)automatic means [4, 6]. OvalTine [13] tracks objects in video so that they can be used as link anchors for both manually and automatically created links. Algorithms for the automatic creation of links in video have focused on specific settings, such as news video [1, 19].

5. DETAIL-ON-DEMAND VIDEO PLAYER

Browsing detail-on-demand video combines interaction characteristics from browsing the Web and changing channels on TV. As the viewer watches a video, the player indicates when links are available and presents labels for them (see Figure 9). The viewer can follow the link to see the destination video or let the original video keep playing. The destination video will play until completion, at which time the original video will continue. If the destination video is not of interest, the viewer can press a "back" button to return to the source video similar to in a Web browser. With such a simple viewing interface, interactions could be performed with a DVD remote control.

The timeline of the video player in Figure 9 shows the link labels generated by the summarizer including the position in the video, the link offset, and the duration of the link destination. Changing the link labels to something more meaningful would be worthwhile during the manual reauthoring of an automatically generated summary. Figure 9 also shows keyframes to the left and right of the video display. The keyframe on the right represents the currently active link destination. The keyframes on the left represent the

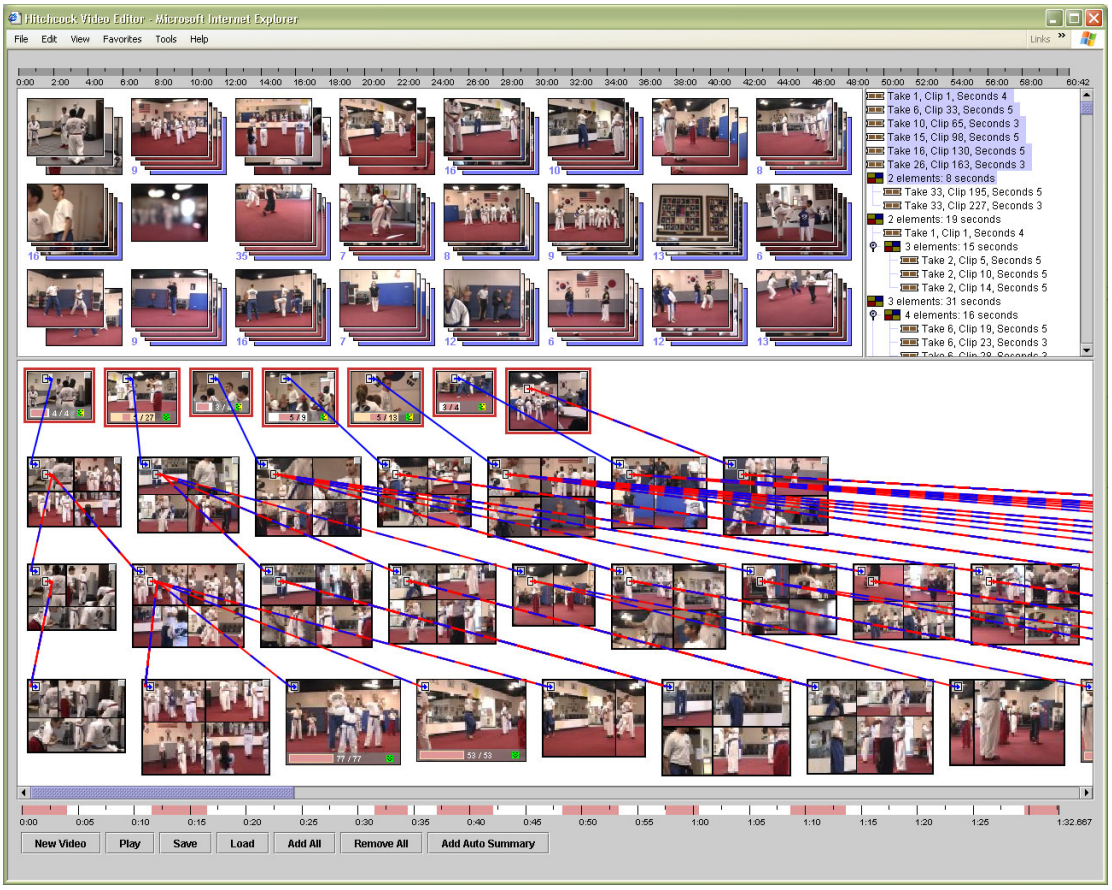

Figure 10: Automatically-generated interactive video including three summary levels and the complete source video.

stack of traversed link anchors. Both sets of keyframes are clickable for link traversal or return from a link.

When a viewer takes a link or returns from a link, a brief video icon is played to provide an indication to the viewer as to what has happened. Video icons are short (less than 1 second) video clips with distinctive audio tracks played in between the two pieces of authored video. These video icons were added because jumping straight from the source video to the destination video happened too quickly and confused viewers. There are different video icons for taking links, returning due to completion of the destination payback, and returning due to the user aborting playback.

We are also experimenting with an alternative user interface metaphor that lets users increase or decrease the amount of detail by several steps at once. This is different from traditional navigational hypertext, where the user takes links to go to another chunk of information somehow related to the material they are currently viewing. Thus, users take one link at a time because taking more than one link would frequently result in seemingly random behavior. Unlike general hypertext, taking multiple links in the hypervideo summaries is more predictable due to the consistent structure that results from their automatic generation. In this case, users may want to jump more than one level deeper into the hypervideo, or return more than one level that they have taken.

6. HYPER-HITCHCOCK

The algorithms described in the previous section generate multi-level summaries with navigational links between the levels of summary to support video browsing. In cases where the interactive summary will be used many times, such as in the case of an index into a training video, authors can refine the automatically-generated interactive summary in Hyper-Hitchcock, a prototype editing and viewing environment for detail-on-demand video. A layout of the summary is generated in its workspace and the human editor can then modify the selection of clips, the link anchors, and the link properties including labels and return properties.

Hyper-Hitchcock's user interface is described in more detail in [12]. Figure 10 shows the Hyper-Hitchcock editing interface. Hyper-Hitchcock provides authors with a two dimensional workspace in which to collect, organize, and interlink clips. The top-left pane groups the source video clips by criteria such as recording time or color similarity. Authors drag clips from this pane to the workspace below where they can be ordered and grouped into composites. Links can be placed between any two clips or composites in the workspace. Authors change the length of video clips and composites by resizing the video frames.

Authoring involves generating sequences of video clips. Given the limited space of the screen, it is convenient to create an iconic representations of a video composite after authoring. Hyper-Hitch-

cock represents the composite as a single image consisting of a collage of images from individual video clips. As an alternative representation, Hyper-Hitchcock provides a tree view of the hierarchy of composites and clips (see top-right pane in Figure 10).

Any video clip or composite can be a link anchor or link destination. In the Hyper-Hitchcock workspace, links are represented as colored arrows into or out of a video frames and iconic representations. Color and line placement provide information about whether the link is into or out of an element in the workspace and the color of the link indicates if the link is connected to the whole element or to a component of the element in the workspace.

To support the use of the generated hypervideo as the starting point for an authored summary, a graphical layout for editing the hypervideo summary is automatically generated in the workspace. Each layer of the summary is presented as a horizontal list of clips and/or composites. Links are represented in the workspace through the normal link visualization of arrows into and out of the keyframes and composite visualizations. The bottom pane in Figure 10 shows part of a four-level summary generated by the first clip distribution clip selection algorithm and the take-to-take with offsets link generation algorithms for a one-hour, 33-take martial arts training video.

In preparation for a future user study, we conducted a pilot study to determine how users would interact with Hyper-Hitchcock [12]. Both participants succeeded in authoring a short hypervideo with several branches. Observations made during the study indicate the need for a study of the player interaction that we are currently preparing. Navigational links in video present a new experience for most people and there are no consistent intuitions as to the behavior of these links. As such, Hyper-Hitchcock needs to be as clear as possible about the effects of links. Early hypervideo viewers will likely experience similar problems as with early hypertext users becoming "lost in hyperspace" or reaching dead ends. In response to this observation, we already made all link labels visible rather than just the active one and added keyframes for links.

7. CONCLUSIONS

In this paper, we presented techniques for automatically generating interactive video summaries. These summaries are represented as detail-on-demand video, a form of hypervideo where video clips are hierarchically organized into video composites, links can exist between any two clips/composites, and only one link may be active at any given time. We previously described an early version of our use of detail-on-demand for video summaries [11]. We have since improved our approach to include several different techniques for selecting video clips to be included in different summary levels that are suitable for different types of video (e.g., home video or produced training video). We also added different methods for placing links and setting link behaviors.

The interactive video summaries are generated to include several linear summaries of different lengths and links in between these summaries and the source video. Generating this multi-level summary consists of deciding how many summary levels to create and their length, determining which video clips to include in each summary level, and generating links between clips and composites within the summary levels and the source video. We developed three different algorithms for selecting clips that are suitable for either inclusion in video summaries. Links in detail-on-demand video have link return properties that determine where the playback continues after the destination video completes or the user aborts playback. Two link generation algorithms have been designed — one for navigating to the beginning of takes or scenes and the other providing more fine-grained access.

Two goals for the design of hypervideo summaries are to minimize user disorientation resulting from link navigation and to minimize the rewatching of video. 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.

We incorporated the algorithms for automatically generating video summaries into Hyper-Hitchcock [12], an editor and player for detail-on-demand video. Authors can refine automatically generated summaries in the editor to make them summarize the source material even better. The player provides information about links available and traversed and interaction possibilities while people watch hypervideo.

ACKNOWLEDGEMENTS

We thank Jonathan Helfman for his helpful suggestions on letting users increase or decrease the amount of detail by several steps at once and on using keyframes in the player interface.

REFERENCES

[1] Boissière, G. Automatic Creation of Hypervideo News Libraries for the World Wide Web. *Hypertext '98 Proceedings*, ACM, New York, 1998.

[2] Christel, M.G., Smith, M.A., Taylor, C.R,. and Winkler, D.B. Evolving Video Skims into Useful Multimedia Abstractions. *Proceedings of CHI'98*, ACM Press, pp. 171-178, 1998.

[3] Girgensohn, A., Boreczky, J., Chiu, P., Doherty, J., Foote, J., Golovchinsky, G., Uchihashi, S., and Wilcox, L. A Semi-Automatic Approach to Home Video Editing. *Proceedings of UIST '00*, ACM Press, pp. 81-89, 2000.

[4] Grigoras, R., Charvillat, V. and Douze, M. Optimizing Hypervideo Navigation Using a Markov Decision Process Approach, in *Proceedings of ACM Multimedia*, ACM Press, pp. 39-48, 2002.

[5] He, L., Sanocki, E., Gupta, A., and Grudin, J. Auto-Summarization of Audio-Video Presentations, in *Proceedings of ACM Multimedia*, ACM Press, pp. 489-498, 1999.

[6] Hirata, K., Hara, Y., Takano, H., and Kawasaki, S. Content-oriented Integration in Hypermedia Systems, *Hypertext '96 Proceedings*, ACM, New York, pp. 11-21, 1996.

[7] Lienhart, R. Dynamic Video Summarization of Home Video, *SPIE 3972: Storage and Retrieval for Media Databases 2000*, pp. 378-389, 2000.

[8] Marchionini, G., and Geisler, G. The Open Video Digital Library, *D-Lib Magazine*. Vol. 8, No. 12, http://www.dlib.org/dlib/december02/marchionini/12marchionini.html, 2002.

[9] Myers, B., Casares, J., Stevens, S., Dabbish, L., Yocum, D. and Corbett, A. A Multi-View Intelligent Editor for Digital Video Libraries, in *Proceedings of the ACM / IEEE Joint Conference on Digital Libraries*, ACM Press, pp. 106–115, 2001.

[10] Rui, Y., Huang, T.S., and Mehrotra, S. Exploring Video Structure Beyond the Shots. *International Conference on Multimedia Computing and Systems*, pp. 237-240, 1998.

[11] Shipman, F., Girgensohn, A., and Wilcox, L. Creating Navigable Multi-Level Video Summaries. In *IEEE International Conference on Multimedia Computing and Expo*, vol. II, pp. 753-756, 2003.

[12] Shipman, F., Girgensohn, A., and Wilcox, L. Hyper-Hitchcock: Towards the Easy Authoring of Interactive Video. In *Human-Computer Interaction INTERACT '03*, IOS Press, 2003.

[13] Smith, J.M., Stotts, D., and Kum, S.-U. An Orthogonal Taxonomy of Hyperlink Anchor Generation in Video Streams Using OvalTine, *Proceedings of ACM Hypertext 2000*, pp. 11-18, 2000.

[14] Sundaram, H. and Chang, S.-F. Determining Computable Scenes in Films and their Structures using Audio-Visual Memory Models, in *Proceedings of ACM Multimedia*, ACM Press, pp. 95-104, 2000.

[15] Sundaram, H. and Chang, S.-F. Condensing Computable Scenes Using Visual Complexity and Film Syntax Analysis. *Proceedings of ICME 2001*, pp. 389-392, 2001.

[16] Uchihashi, S., Foote, J., Girgensohn, A., and Boreczky, J. Video Manga: Generating Semantically Meaningful Video Summaries, in *Proceedings of ACM Multimedia*, ACM Press, pp. 383-392, 1999.

[17] Wildemuth, B., Marchionini, G., Yang, M.,Geisler, G., Wilkens, T., Hughes, A., and Gruss, R. How fast is too fast? Evaluating fast forward surrogates for digital video, *Proceedings of the 2003 ACM and IEEE Joint Conference on Digital Libraries (JCDL '03)*, pp. 221-230, 2003.

[18] Yeung, M.M. and Yeo, B.-L. Video Visualization for Compact Presentation and Fast Browsing, *IEEE Transactions on Circuits and Systems for Video Technology.* Vol. 7, no. 5, 1997.

[19] Zhang, H.J., Tan, S.Y., Smoliar, S.W., and Yihong, G. Automatic Parsing and Indexing of News Video, *Multimedia Systems*, 2 (6), pp. 256-266, 1995.