Simplifying Video Editing Using Metadata

Juan Casares, A. Chris Long, Brad A. Myers, Rishi Bhatnagar, Scott M. Stevens, Laura Dabbish, Dan Yocum, and Albert Corbett

Human Computer Interaction Institute, Carnegie Mellon University[1]

ABSTRACT

Digital video is becoming increasingly ubiquitous. However, editing video remains difficult for several reasons: it is a time-based medium, it has dual tracks of audio and video, and current tools force users to work at the smallest level of detail. Based on interviews with professional video editors, we developed a video editor, called Silver, that uses metadata to make digital video editing more accessible to novices. To help users visualize video, Silver provides multiple views with different semantic content and at different levels of abstraction, including storyboard, editable transcript, and timeline views. Silver offers smart editing operations that help users resolve the inconsistencies that arise because of the different boundaries in audio and video. We conducted a preliminary user study to investigate the effectiveness of the Silver smart editing. Participants successfully edited video after only a short tutorial, both with and without smart editing assistance. Our research suggests several ways in which video editing tools could use metadata to assist users in the reuse and composition of video.

Keywords

Digital video editing, multimedia authoring, metadata, Silver, Informedia.

INTRODUCTION

Digital video is becoming increasingly ubiquitous. Digital video equipment is more accessible than ever, and there is an increasing amount of video material available on the World Wide Web and in digital libraries. As technology and content become available, users will shift from passive spectators to active creators of video. Forrester Research predicts that by 2005, 92% of online consumers will create personal multimedia content at least once a month [17]. However, although many exciting research projects are investigating how to search, visualize, and summarize digital video, there is little work on new ways to support the use of the video beyond just playing it.

This is unfortunate, because video editing has several unique challenges not found with other media. One is that digital video is a time-based medium. This property makes it difficult for users to browse and skim video. Users often must linearly search their source video to find the clip they desire.

Another challenge for editing video is that it is a dual medium. Most "video" actually consists not just of a video track but also an audio track. These tracks must be kept synchronized, but the user must also be able to overlay them when desired, for example during transitions from one shot to another. Further, when a shot is cut from a video for use elsewhere, the user must be able to disentangle overlaid audio and video.

A third problem is that the syntactic units that users want to edit are shots of video and words or sentences of audio, but current tools require users to examine video at the individual frame level and audio using a waveform. To perform most editing operations, such as cutting a shot, the user must manually pinpoint specific frames, which may involve zooming and numerous repetitions of fast-forward and rewind operations. Finding a specific word or sentence using a waveform is similarly tedious.

These problems make editing video a difficult, tedious, and error-prone activity. Commercially available tools, such as Adobe Premiere [2] and Apple iMovie [1], allow creation of high-quality video, but they do not adequately address the issues raised above, which makes them harder to use, especially for novices.

To better understand video editing, we visited a video processing studio and interviewed professional video editors. We also examined commercial and research video editing systems.

We created a system, called Silver, to explore techniques to make video editing easier. The key innovations in the Silver editor include: providing an editable transcript view; coordinating the selection across all views including partial selections and different selections in audio and video; and smart editing through smart snap, smart selection and smart cut and paste. Silver uses video and metadata from the Informedia Digital Video Library [22].

This paper is structured as follows. The next section discusses related work on video systems. Next, we discuss a specific challenge for video editing: L-cuts. Then we describe the Silver interface, followed by the evolution of the Silver interface. Next, we discuss issues in the

DIS2002, London © Copyright 2002 ACM 1-58113-2-9-0/00/0008 $5.00

[1] Pittsburgh, PA 15213 USA. juan.casares@cs.cmu.edu, http://www.cs.cmu.edu/~silver.

interface and implementation of Silver. Then we describe a pilot user study and its results. Finally, we discuss future work and conclude.

RELATED WORK

In this section, we describe different types of related work. We start with systems that use metadata for video editing. Next, we discuss systems that automate the process of video editing to some degree. Then we discuss work on video visualizations that address the issues of scale and time. Lastly, we describe Informedia, which is our source of video and its corresponding metadata.

Metadata and Video Editing

The notion of using metadata for editing video is not new. For example, Mackay and Davenport examined the role of digital video in several interactive multimedia applications and concluded that video is an information stream that can be tagged, edited, analyzed, and annotated [13]. Later, Davenport et al. proposed using metadata for home movie editing assistance [7]. However, they assumed this data would be obtained through manual logging or with a "data camera" during filming.

Currently, there is a large body of work on the extraction and visualization of information from digital video (e.g., [8, 19]) that make a data camera unnecessary.

One example of a system that uses metadata is IMPACT [21], which uses automatic cut detection and camera motion classification to create a high level description of the structure of the video. The user can organize the shots in a tree structure and then edit the composition by moving the branches [20]. IMPACT supports this process by recognizing objects across multiple shots.

Automation

Several systems edit video with varying degrees of automation. Fully automated systems may be used for news on demand [3] or quick skimming [5], but do not really support authoring.

The Video Retrieval and Sequencing System (VRSS) [6] semiautomatically detects and annotates shots for later retrieval. Then, a cinematic rule-based editing tool sequences the retrieved shots for presentation within a specified time constraint. Examples of cinematic rules include the parallel rule, which alternates two different sets of shots, and the rhythm rule, which selects longer shots for a slow rhythm and shorter shots for a fast one.

The Hitchcock system [9] automatically determines the "suitability" of the different segments in raw home video, based on camera motion, brightness, and duration. Similar clips are grouped into "piles." To create a custom video, the user drags segments into a storyboard and specifies a total desired duration and Hitchcock automatically selects the start and end points of each clip based on shot quality and total duration. Clips in the storyboard are represented with frames that can be arranged in different layouts, such as a "comic book" style layout [4].

The MovieShaker system automates video editing almost completely [18]. The user specifies multiple video clips and a "mood" (e.g., romantic, wild), and the program combines and edits the clips into a single video.

While automation makes editing faster, it usually involves taking away power from the user, which is not always desirable. In fact, user studies led to changes in Hitchcock to give more information and control to the user [10].

Visualizing Time and Scale

Video editing is difficult in part because it is hard to visualize time. Due to the semantic and syntactic nature of the video, where selecting a five-second piece involves fiddling with frames in $1/30^{th}$ of a second increments, it becomes important to present the information at different scales. No one scale is sufficient to efficiently perform all the operations needed.

The Hierarchical Video Magnifier [14] allows users to work with a video source at fine levels of detail while maintaining an awareness of the context. It provides a timeline to represent the total duration of the video source; a second timeline shows a detail of the first one and illustrates it with a frame sampling. The user can select which portion of the top timeline that is expanded on the next one. There is also a mechanism that can be used to add a new timeline with a higher level of detail. Successive timelines create an explicit spatial hierarchical structure of the video source.

The Swim Hierarchical Browser [23] significantly improves on this idea by using metadata. Swim displays automatically detected shots in the higher level layers instead of frame samples.

Hierarchical views in the Hierarchical Video Magnifier and Swim are used for navigation only. Silver extends hierarchical views to allow editing as well.

Informedia

Silver obtains video from the Informedia Digital Video Library [22], which currently has over 2,000 hours of material and is adding new material daily. Our primary motivation for using Informedia is that it generates several types of metadata, which Silver uses to enhance its editing capabilities.

A textual transcript of the audio track, generated from closed-captioning information and speech recognition is one type of metadata Informedia provides [8, 11]. The transcript is time-aligned with the video using the Sphinx speech recognition system [16]. Silver is the only video editor we know of that uses an audio transcript.

Informedia uses image analysis to detect shot boundaries and extract representative thumbnail images from each shot [5] and automatically creates titles and summaries for video segments.

Informedia also provides search infrastructure to allow users to find video clips based on keywords and to browse the results.

L-CUTS

When working with video that has been professionally edited, it often happens that the start and end times of a segment are different for the audio and the video. This is called an "L-cut". For example, in newscasts, the camera often cuts from an anchorperson to a field reporter while the anchorperson is still talking. L-cuts give the video a continuous, seamless feel, but make extracting pieces much harder because the video and audio portions must be adjusted separately.

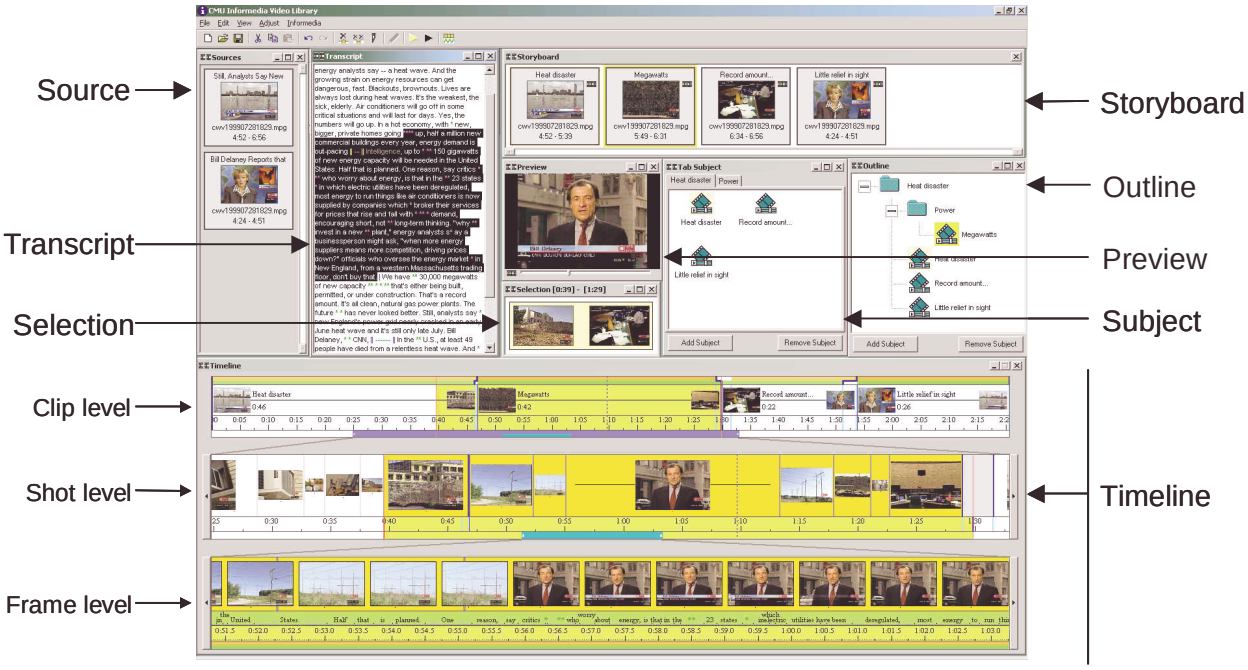

Figure 2 Overview of the Silver interface

This requires much fine-tuning to remove extraneous video or audio portions.

We conducted an informal study to investigate the prevalence of these cuts, and found it to be significant, at least in recorded news shows. In looking at 238 transitions in 72 minutes of video clips recorded from CNN by Informedia, 61 (26%) were L-cuts. Of the 61 L-cuts, 44 (72%) were cases where the audio of the speaker came in before the video but ended at the same time. Most of these were interviews where the voice of the person being interviewed would come in first and then the video of the person after a few seconds (see Figure 1, Shots A and E). Most of the other ways to overlap the video and audio were also represented. Sometimes, the audio and the video of the speaker came in at the same time, but the audio continued past the video (Figure 1, Shot B) or the video ended after the audio (Shot D). When an interviewee was being introduced while appearing on screen, the video might come in before the audio, but both end at the same time (Shot C). To copy or delete any of these would require many steps in other editors.

SILVER

To address the problems of video editing mentioned above, we developed Silver, a digital video editor. Silver simplifies video editing primarily by incorporating metadata into the interface and by providing multiple, synchronized views of the video composition. Silver also includes support for managing L-cuts.

We want to make video editing as easy as text editing. To that end,

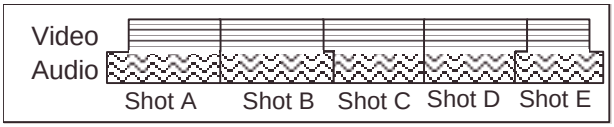

Figure 1 Different kinds of L-cuts

we have examined text editors and have adopted some of their techniques. Text editors have long had multiple representations in order to support different styles of editing and different tasks. For example, Microsoft Word provides “outline,” “normal,” “print layout,” and “print preview” views. The multiple views in Silver allow the user to browse the video by displaying it across space instead of time, to view and manipulate the audio and video separately, and to examine the video at multiple levels of detail. An overview of the Silver interface is shown in Figure 2.

The remainder of this section describes these views and how each supports video editing. The first subsection describes the two views for managing the video directly, the storyboard and the timeline. The next discusses a view for managing the audio track using a transcript. Following is a brief description of other views in the Silver interface. The final subsection describes the smart editing features in Silver.

Storyboard view

The storyboard view allows the user to arrange and rearrange the video composition at a large scale. It displays each *clip*[2] in the composition as a separate icon containing a title and a key frame. Initially, we dynamically analyzed each clip to identify a representative key frame. However, early users were confused when the displayed key frame changed as they edited the clip and preferred using a static frame close to the start of the clip. The clips are sorted chronologically, so moving clips within the storyboard changes the order in which they appear in the final video.

In addition to the composition, the storyboard also contains a “scratchpad,” or staging area. Clips in this area are not part of the composition, but are easily available, which is convenient for pieces

[2] A clip is a continuous segment of video from a single source file.

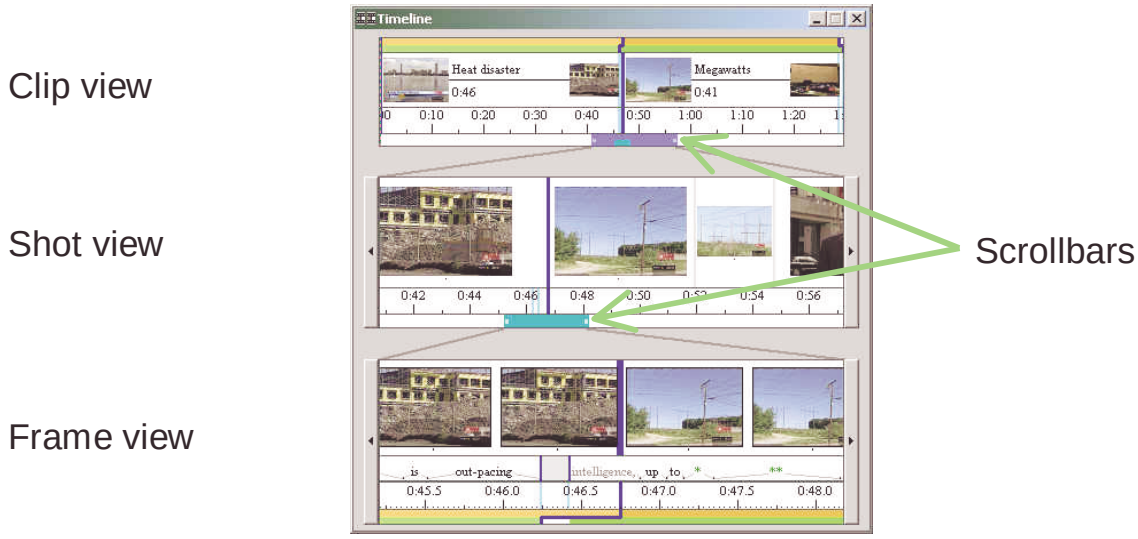

Figure 3 Timeline window. The scrollbar in the clip view can be moved and resized to scroll and zoom the shot view. The shot view scrollbar can be moved and resized to scroll and zoom the frame view.

under consideration or being edited. A vertical line separates clips of video in the scratchpad from clips currently in use. Dragging a clip across this line adds or removes it from the composition.

Hierarchical timeline view

Silver, like most video editors, provides a timeline view as the primary view for detailed editing. Silver allows users to easily view and manipulate video at different scales by providing a three-level view: clip, shot, and frame levels (see Figure 3). The timeline allows the user to work at a high level of detail (i.e., in the frame level) without losing the context within the composition (i.e., in the shot and clip levels).

The top line is the clip level which represents the entire video. At the top are two colored bars, one for audio and one for video, that show gaps in the tracks and clip boundaries. The central section of the clip level shows a more detailed view of each clip than is shown in the storyboard. In addition to a representative frame, it shows a timeline and if space allows it shows the end frame, title, and duration.

Users control how much is displayed in the other two levels of the timeline using a sophisticated scrollbar at the bottom of the clip level. In addition to the usual dragging behavior, the scrollbar may also be resized to zoom the shot level in or out. It also contains a miniature version of the shot level scrollbar, which shows the position of the frame level within the shot level and can be dragged to scroll the frame level.

The shot level allows users to interact with video at a higher level than individual frames by displaying video as individual *shots*[3]. Algorithms that detect significant between-frame image changes identify the shot boundaries [12]. Each shot is displayed using the representative frame for the shot as chosen by Informedia. Shots smaller than a certain threshold are displayed with proportionally smaller thumbnails (for example, the third shot in Figure 3). As in the clip level, there is an augmented scrollbar at the bottom of the shot level used to scroll and zoom the frame level.

The frame level can display the individual frames of the video, so the user can quickly find particular cut points. The frame level addresses the dual medium issue by displaying the transcript, including the start and end times of every word. In other editors, there is little information about the audio, so users must repeatedly play the audio to find a desired part. The time-aligned transcript Silver provides allows the user to locate the desired part of the audio and video more quickly and easily than is possible with conventional editors.

At all levels, boundaries between clips are shown as blue lines. The user can resize a clip by dragging the clip boundary.

A key feature of the Silver timeline is that it shows different representations together, allowing the user to see the clip boundaries, the time, samples of frames, the cuts, the transcript, annotations, etc. In the future, other representations could be added, such as audio waveform or labels for recognized faces.

Transcript view

Unlike any other editor of which we are aware, Silver shows a time-aligned textual transcript alongside the video, under it in the frame level of the timeline view (see Figure 3). This view is very useful for fine-grained audio and video synchronization, but little text can be shown there simultaneously because it is mapped to time in a single line.

So that the user can more easily view and edit the transcript, Silver offers a text-editor-like window dedicated to the transcript (see Figure 4). As in the timeline view, the text in the transcript window is tied to the video, so the user can edit the *video* by cutting and pasting the text in the *transcript*.

One challenge for displaying transcripts is the need to display information in addition to the spoken words. Our solution is to insert special characters into the text. For example, the boundary between clips is shown as a blue double bar ("‖"), and sections without audio

[3] A shot is a continuous segment of video from a single camera.

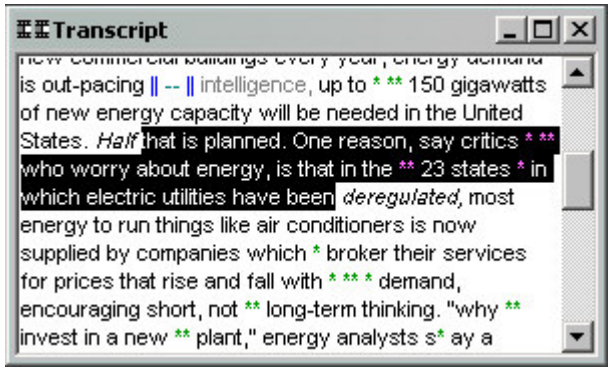

Figure 4 Transcript window. In the top line, we can see blue clip breaks ("||"), enclosing a gap in audio ("- -"). The word *intelligence* was cut in the middle. The green stars represent noise or recognition errors. At the edges of the selection, *half* and *deregulated* are partially selected.

are represented with dashes ("-"). This is shown near the top of Figure 4.

Because speech recognition can contain mistakes, Silver inserts a green star ("*") where there appears to be a gap, misalignment, or noise. Silver allows the user to manually correct mistakes in timings or recognition. Users can type the correct text and indicate what video is associated with it

Other views

Silver provides the user with many other views. The Outline and Subject views permit the classification of clips in a hierarchical or flat taxonomy to help users organize their composition. The Source view provides access to all the source material in one convenient place. The Selection view shows the first and last frames of the selected video. The Preview window is used for video playback.

Silver is built on top of the Informedia client interface, so the search, browse, and summarization views provided by Informedia are available to the user. For example, the user can use the Informedia Search window to enter a keyword and then drag one of the clips displayed in the Search Results to one of the Silver windows.

These views are described in more detail in another paper [15].

Smart Editing

Silver tries to reduce the tedium of editing video by using metadata to provide more intuitive interaction. One large source of this tedium is L-Cuts, which are a common feature of edited video. In this subsection, we describe smart snap, selection, and cut and paste.

Smart Selection

One reason that video editing is much more tedious than text editing is that with video the user must select and operate on a frame-by-frame basis. Simply moving a section of video may require many minutes while the beginning and end points are laboriously located.

For selections in an audio-based view, Silver may snap the edge of video selection to a shot break. To avoid flashing a few frames of another shot, we use a threshold of one second when the shot break is outside of the audio selection. For example, the right edge of the video selection in Figure 5 is snapped to a shot boundary. As shots

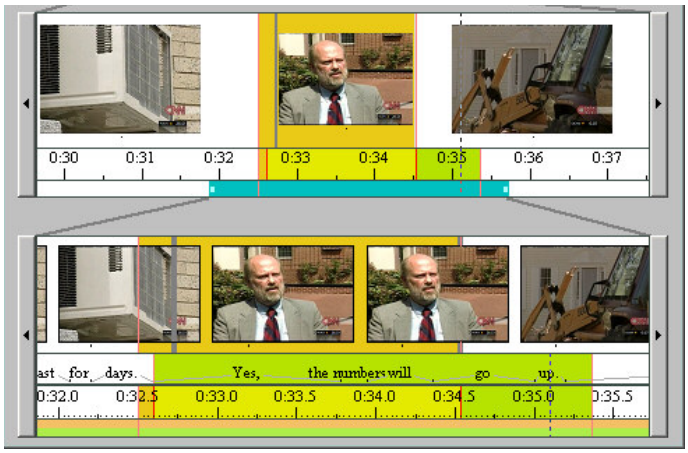

Figure 5 Smart selection. Audio and video can be selected separately, and video selection snaps to shot boundaries.

become longer than a second, they no longer are objectionable. We use a threshold of 0.3 seconds when the shot break is inside the selected region. It is preferable not to jump to the middle of a shot.

For video-based views, if the selection cuts a word in half then Silver extends or reduces the audio selection to the nearest word edge. This is shown in Figure 5 where the both edges of the audio selection are snapped to a word. This might take the audio selection across a shot break.

Selecting in a mixed media view causes the selection to snap in both audio and video. However, the user is always in control: the smart selection can be disabled by pressing a modifier key.

Silver eases selection by providing a mechanism to select a meaningful segment based on the view. For example, in the shot level of the timeline, a double click selects the corresponding shot. In some cases, a larger segment is selected with a triple click, such as in the transcript view (where a double click selects a word and a triple click selects a sentence).

When the user selects a region, Silver may adjust the audio or video boundaries based on editing rules of thumb and the available metadata. This is especially useful when the source contains L-cuts.

Smart Snap

When moving the cursor around in the timeline view, the cursor snaps differently to features of the composition depending on which row the mouse is moving over. In video-based rows, the cursor snaps to shot breaks if close by, or to the nearest frame otherwise. In the transcript row, the cursor snaps if close to a word boundary. In the case of silence, the cursor does not snap. In addition, the cursor always snaps to clip boundaries and the edges of the current selection.

Smart Cut and Paste

When a segment with non-aligned audio and video edges is deleted or pasted, the resulting ragged edges would typically require the user to again go down to frame level and adjust the edges or create a special effect to avoid gaps in the composition. For example, if a segment with shorter video than audio (such as Shots A, B, or E in Figure 1) is inserted, the user would need to close the gap in video by either extending the video or shrinking the audio on either end of the gap.

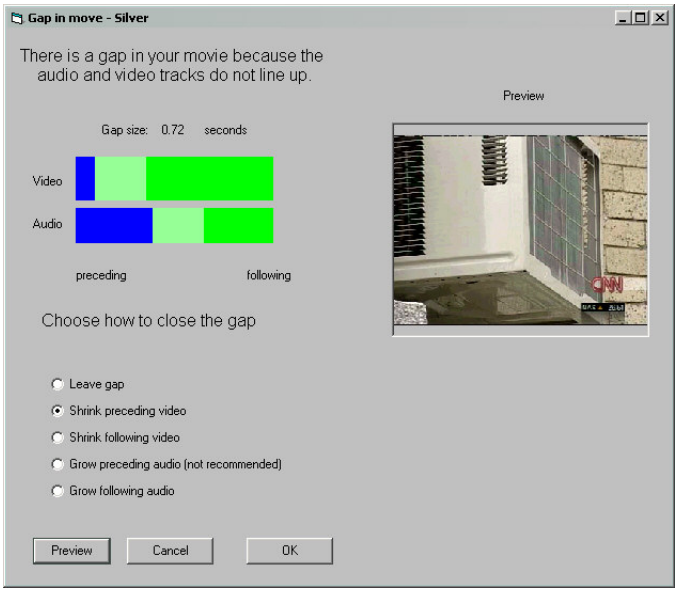

Figure 6 Close gap dialog. This dialog would appear if the user deleted the selection in Figure 5.

Silver makes this task much easier by semi-automatically determining how to deal with the uneven ends. It presents a dialog box to the user with different options to close the gap. This dialog displays the current gap and how the different options would close it. For example, after deleting the segment that is shown selected in Figure 5, a dialog box like the one in Figure 6 would appear. This dialog is needed because more audio than video is selected in Figure 5, so deleting the selection would leave a gap in the audio. Deleting Shot E in Figure 1 would have the same effect.

If there is a gap in audio (e.g., if Shots C or D from Figure 1 are pasted), then Silver recommends leaving it because silence is generally not disruptive. However, it is not acceptable to leave a gap in video, which would show up as black or blank video. Silver recommends extending the edges of involved video as long as it does not take them across a shot boundary.

In cases where the editing operation leaves two gaps, such as when a clip with jagged edges on both ends is inserted, Silver presents a double version of the dialog box where the user can look at both ends simultaneously.

THE EVOLUTION OF SILVER

This section describes how the Silver interface changed over time as a result of informal usage we observed.

Timeline

The most complex part of the Silver interface is the timeline. Much of the time we spent designing Silver focused on this view.

Our timeline started with a simple scrollbar, a video track, and an audio track. It showed the clips in the composition, representing them by stretching a representative key frame. However, we wanted to represent more detailed information in the timeline. Our simple timeline could not show different levels of granularity simultaneously. Our approach was to show context and detail in a multiple-level hierarchical timeline.

Other video systems also use multi-level timelines. Some editors (like Premiere) have a two-level timeline. A detailed timeline represents a short amount of time and is used for most editing tasks. It can display different tracks, including multiple video and audio channels, effects, etc. The second timeline is a very abstract representation of the piece. This timeline is only used to determine what is being shown in the detailed timeline.

Timeline Scrollbars

There are many interaction challenges when designing a multi-level view like our timeline. In particular, the correlations between the scrollbars required several iterations and fine-tuning. Many of our initial assumptions needed to be changed in response to user testing.

Originally, the scrollbar thumbs were designed to represent a constant proportion and position in relation to the whole scrollbar. So, for example, the top scrollbar thumb could cover the second half of the scrollbar, meaning that the next level represents the second half of the composition. When the window is resized or the composition is edited, the scrollbar would still cover the second half and the next level would still show the second half of the composition, although with a different zoom resolution. This behavior was extremely annoying to users, especially after they had carefully set the zoom level to show, say, shots at a particular resolution.

In the same sense, the middle level's scrollbar thumb would represent a fixed proportion and position in relation to the top scrollbar. To change the resolution of the middle level the user needed to resize the top scrollbar by dragging on one of its edges. However, the middle scrollbar remained the same, meaning that the bottom level's resolution and position changed as well. So, changing the top scrollbar would often change what was displayed in the bottom level and users would need to readjust the middle scrollbar.

Since the middle scrollbar did not change in response to changes in the top scrollbar, displaying the last frame of the composition in the bottom level involved dragging the top scrollbar to the right edge, and then dragging the middle scrollbar to its right edge.

We changed the interaction model so each scrollbar kept its zoom-level constant. We assumed users did not want the bottom level to change when the top scrollbar was resized. So, growing the top scrollbar would also shrink the middle scrollbar so that the bottom level would maintain its resolution. Furthermore, the resizing needed to be centered on the thumb, so that dragging one edge would similarly change the other edge. Otherwise, the bottom level would move even if it maintained its resolution. Pressing the *alt* key while resizing disables the centering feature, allowing the user to change just one side of the scrollbar thumb. This is useful when the user is interested in something that is not in the center.

Users had problems recognizing the relationship between the different levels. That is, they did not understand that each level was a blowup of the previous one. We added guiding lines from the scrollbars to the edges of the next level to help indicate this. Also, we color-coded each thumb, and showed a miniature version of the middle thumb in the top-level thumb.

Timeline Views

Each part of the timeline represents video at different levels of detail.

Frame View

Initially, we showed a frame sample row in every level. However, in the less-detailed levels (i.e., shot and clip), samples could seem almost random since a single frame sample represented a large amount of time. Users were confused because they expected the samples to be representative of the time they covered on the timeline. The confusion ended when we removed frame samples from the other levels.

In our original presentation of the frame view, users would assume that the video samples in the frame view represented individual frames. However, the samples only showed individual frames if the level was at the highest level of detail. To make the difference between showing a sample and all individual frames, we added a double-line border to frames when shown individually.

Clip view

We represented clips by stretching the picture of a representative frame to fill its corresponding space in the timeline. For long clips, this distortion made the frame unrecognizable. Now we represent the clip with the first frame. If there is enough space, we also show the last frame. When several clips were shown side by side, it became hard to distinguish start frames from end frames. So we now display the last frame at a reduced size. If there is enough space between the start and end frames for a clip, we also show textual information about the clip such as its name and duration.

Since the composition length, window size, and zoom level may change, an important consideration for our views was that they should always show something meaningful. Clips and shots are displayed in more detail if more space is available, but there is always some indication of the presence of each shot and clip. For example, if a clip has very little space, it only shows that a clip exists (with a thick blue line). With more space, it shows the first frame. With more space, the first and last frames. Finally, it includes title and duration if there is a lot of space. The timeline also shows varying amounts of information depending on available space. As space increases, words go from small black dots to the word with starting and end times. If some words do not fit, we try to rearrange them in two levels. With frames we sample to show as many frames as we can, until we can show each individual frame.

INTERFACE AND IMPLEMENTATION ISSUES

This section discusses issues we encountered in our design of the Silver interface and in its implementation.

Many views, one model

All of Silver's views actually represent the same composition. Any change to the composition is reflected immediately in every view. For example, when the user drops a new clip into one of Silver's views, all others are instantly updated.

In addition, selection in one view is mirrored across the others. This feature brings up a number of interesting user interface design challenges.

First, the selection in video may begin and end at different times from the selection in audio. As is described in the smart editing section, this feature is critical to achieving high quality when composing previously edited content. Some views only show the audio or the video track; these views only show their corresponding selection.

We use color to show selections of different media types: orange for video, green for audio, and yellow for segments selected in both (see Figure 5). Usually, users want to select audio and video together, but sometimes they need to separate the two. Single-track selection is done with the right mouse button instead of the left, which selects both. For example, dragging the right button across the frame view of the timeline would select a piece of video without selecting audio.

The second problem that arises from synchronized selection is defining what the "equivalent" portion is. The different views show different levels of granularity, so it may not be possible to represent the selection accurately in some views. For example, if a few frames are selected in the timeline view, as shown in yellow in Figure 5, it is not obvious what should be shown in the storyboard view, since it shows only complete clips. Silver uses a dark shade to highlight a fully-selected item, and a lighter shade to highlight a partially-selected one.

Another example of this problem arises between video and audio views. Individual words in the transcript span multiple frames. When a word is selected in the transcript, it is straightforward to select all the corresponding frames. However, selecting only one of those frames in the video will correspond to only part of that word. This partial selection is shown in the transcript by making the word italic, for example the words "half" and "deregulated" at the edges of the selected text in Figure 4.[4] In this case, if the selected video is moved, the word will be cut into two pieces. Silver represents this by repeating the word in both places, but showing them in a faded color. For example, the word "intelligence" in Figure 4 was split.

Information Overload

Video is time-based, which makes it difficult to display well statically, and the more comprehensive the static display, the more screen real estate it requires. Also, the dual nature of video demands additional screen space. Organizing the large amount of information users need while editing video has been a significant challenge in the design of the Silver interface.

The problem of information overload is exacerbated by the inadequacy of auxiliary media, such as paper, to supplement the interface. In editing a large text document or program, people often print sections so that they can refer to them while editing without using up screen space. However, this strategy is not effective for editing video because of its dynamic nature.

The primary mechanism we have used to address the information overload problem is the scrollable, zoomable timeline view. With this

[4] We would prefer the selection highlight to be green over fully selected words and light green over partially selected words, to be consistent with other views. Unfortunately, the Visual Basic text component does not support this.

timeline, the user need not view the entire video at the highest level of detail, but may see only the currently relevant part in detail, and still see the context around it in the other, less-zoomed-in views. This approach may be useful in other information-intensive editing environments.

USER STUDY

We conducted a pilot study to evaluate the smart editing features of Silver. Participants were asked to use Silver to perform a series of editing operations on one video with smart editing enabled, and another series on a different video, this time with smart editing disabled. Our participants were seven paid subjects, graduate students from the School of Computer Science.

Design

This study was a within-subjects design. All participants did both experimental tasks. Each performed one task with smart editing enabled and one with smart editing disabled. The order of the tasks and the order of smart editing were counterbalanced across participants.

Procedure

The experiment had three parts: tutorial, experimental tasks, and post-experiment questionnaire.

In the first part, participants read the Silver tutorial and performed training exercises in it. They learned how to scroll and zoom the timeline, select video, and play video. They also learned how to delete segments of video and clean up the cuts. They performed a practice task in which they manually cleaned up a cut and a practice task in which they used the smart cut and paste.

In the second part of the study, each participant performed the two experimental tasks. For each task, participants were given one of two video compositions and a corresponding list of edits to be performed. One video composition was about the Mars polar lander; the other was about forest fires in Russia. The source videos for each task were obtained from CNN footage. The videos and shots to cut for each of the two tasks were selected to be equally difficult and to be entertaining.

Participants were asked to cut specific shots from the composition. After each cut, they were instructed to play the transition and, if they judged it to be “rough”, to edit it to make it “smooth”.

At the end, participants filled out a paper questionnaire about the study and demographic information. Users were asked about the system in general, the smart editing features, the tutorial, and each of the two tasks.

Results and Analysis

On average, both experimental tasks together took participants 33 minutes. There were no statistically significant differences in time to complete either task due to task, order, or smart editing.

We did find significant correlations between two questions on the questionnaire and the time to complete the task with smart editing disabled. Participants who rated “exploration of features by trial and error” as more “encouraging” than “discouraging” (on a scale of 1–9) took longer to complete the task (correlation 0.82, $p < 0.046$). This may be due to participants who felt encouraged to explore the interface spending time with parts of the interface that were not directly related to the task.

Qualitatively, nearly all participants made a positive comment about Silver, such as describing it as “cool software, very intuitive to use”. Two users believed they experienced a learning effect; one said that one task was “easier because it was the second task”. Another user mentioned, “after I could familiarize myself with it, it became easier to use and not as intimidating”. However, the times recorded show no significant learning effect.

We were very encouraged by the fact that all of the study participants were able to succeed in both of their editing tasks, especially given the brevity of the tutorial.

The purpose of the timelines and what they displayed were not immediate obvious to any of the participants. Participants recognized that one timeline represented the video, but did not know what the other two were. For all but 2 participants, the tutorial cleared up this confusion.

The existence of L-cuts in video was also not apparent to participants until they encountered a task in the tutorial where they deleted a shot that had an L-cut. The tutorial did not specifically talk about L-cuts, and it seemed that participants were not as watchful for them during the experimental task as they should have been. In retrospect, this is not surprising considering they were novices in video editing.

We believe that the timelines would have been easier to understand and users would have been more attentive to L-cuts if the tutorial had explained what clips, shots, and frames were and what L-cuts were conceptually before describing how the interface worked.

Some users had difficulty with the scrollbars at times because the thumbs were very small. It was difficult to grab the thumbs and move them and even more difficult to resize them. Other than that problem, we were pleased to see that users understood how to scroll and zoom the multiple levels.

We added a double border to individual frames to make it more apparent when individual frames were showing in the frame level. However, some users did not seem to be sure when individual frames were being shown and when they were not. The frame view may need an even more overt visual cue to make this distinction obvious.

Users felt disoriented after deleting a shot. When a shot is deleted, timelines in other systems typically shrink, whereas in Silver it maintains its size by readjusting its zoom level. This sometimes led to participants having difficulty locating its former position. A brief animation might make this explicit to users.

We expected to find a greater effect of using the smart editing options. The few video editing experts that have been involved with this project were enthusiastic about incorporating these features. Apparently, our naïve users could not distinguish between the effects of the different options offered by the smart cut and paste dialog, possibly because in many of the L-cuts they encountered in the experiment, the gap in the audio or video track that was created was

very small, which caused the different fixes to be only very subtly different from one another. Users did say that the schematic view of the video was intelligible and helpful. This suggests that we should automatically carry out the recommended action when the effect is small. Also, we should reevaluate the dialog box using video where the differences among fixes is clearly perceptible.

We suspect that participant motivation was an important variable in performance. Some users seemed to try to finish as soon as possible, while others experimented with the interface. Time to complete a task might not be the best measurement of the effectiveness of a creative authoring tool, since it penalizes exploring different options and paying attention to detail. This might explain the significant correlation described in the previous section. In the future, we hope to study the quality of the produced video which might provide richer information.

FUTURE WORK

There are many possible directions for future work with Silver. For example, corrections to the transcript by users could guide the speech recognizer in the text alignment, so users would not need to align the text manually. The transcript view could also support the authoring of new productions from scripts. The user could type or import a new script, and then later the system would automatically match the script to the video as it is shot.

We are exploring when these adjustments can be fully automated in a reliable way. We are also looking at other metadata that might be helpful in giving a recommendation. For example, we are considering mixing the audio of both edges when one audio track is music or other background sounds. We also want to increase the palette of options for handling gaps in the composition. For example, a third piece of video footage can be placed to cover the gap area, or a "special effect" like a dissolve can be used to smooth the transition over the gap.

Silver could also incorporate other metadata in the future, such as face detection and recognition, and voice matching and recognition. With these data, Silver could allow users to easily perform editing operations like "Cut all video where person X is talking."

Another type of analysis that would assist editing is object recognition and tracking. Silver could use this information to allow users to make moving objects hyperlinks, or cut and paste all video related to specific objects.

Tracking of people, faces, or objects would also allow interesting synthetic graphics to be overlaid, for example to explain motion in a video for physics instruction.

CONCLUSIONS

Silver is a system designed to make high quality video editing easier. It demonstrates several techniques to overcome some of the main difficulties of authoring in this domain. Namely, it uses multiple views that display not only the raw audio and video but also higher-level metadata. A single selection is coordinated across all views, even when the audio and video are unaligned. A textual transcript is linked to the video, providing a view that affords quick and accurate editing in the audio domain. Silver provides a three-level timeline, presenting both context and detail, which can be a useful tool for video editing.

Silver shows that video metadata can be used to create an advanced interface for aiding novice users in the performance of otherwise difficult or tedious editing tasks. It demonstrates three forms of smart editing, with selection, snap, and cut and paste. Currently the metadata used is only the clip titles, the transcript, and shot breaks. However, as more metadata becomes available, the smart editing features can be improved. For example, we could adjust the selection to match speaker changes or to avoid cutting in the middle of a panning or zooming camera operation.

The Silver project is investigating some exciting ways to make high quality video editing easier, and points the way to a future in which everyone can be as comfortable editing video as they are today editing text.

ACKNOWLEDGMENTS

We would like to thank our participants. The Silver Project is funded in part by the National Science Foundation under Grant No. IIS-9817527, as part of the Digital Library Initiative-2. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the National Science Foundation.

REFERENCES

1. Apple Computer, Inc. iMovie. http://www.apple.com/imovie/.
2. Adobe Systems Incorporated. Adobe Premiere. http://www.adobe.com/products/premiere/main.html.
3. Ahanger, G. and Little, T. D. C. "Automatic Composition Techniques for Video Production." IEEE Transactions on Knowledge and Data Engineering, 10(6):967-987, 1998.
4. Boreczky, J., Girgensohn, A., Golovchinsky, G., and Uchihashi, S. "An Interactive Comic Book Presentation for Exploring Video." CHI Letters: Human Factors in Computing Systems (SIGCHI), 2(1):185–192, April 2000.
5. Christel, M., et al., "Techniques for the Creation and Exploration of Digital Video Libraries." Chapter 8 of Multimedia Tools and Applications, B. Furht, ed. Kluwer Academic Publishers. Boston, MA. 1996.
6. Chua, T. and Ruan, L. "A video retrieval and sequencing system." ACM Transactions on Information Systems, 13(4):373–407, 1995.
7. Davenport, G., Smith, T.A., and Pincever, N. "Cinematic Primitives for Multimedia." IEEE Computer Graphics & Applications, 11(4):67–74, 1991.
8. Gauch, S., Li, W., and Gauch, J. "The VISION Digital Video Library." Information Processing & Management, 33(4):413–426, 1997.
9. 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." CHI Letters: Symposium on User Interface Software and Technology (UIST), 2(2):81–89, 2000.
10. Girgensohn, A., Bly, S., Shipman, F., Boreczky, J. and Wilcox, L. "Home Video Editing Made Easy — Balancing Automation and User Control." In Human-Computer Interaction INTERACT '01. IOS Press, 464–471, 2001.

11. Hauptmann, A. and Smith, M. “Text, Speech, and Vision for Video Segmentation: The Informedia Project.” In AAAI Symposium on Computational Models for Integrating Language and Vision, Cambridge, MA, Nov. 10–12, 1995.
12. Hauptmann, A.G. and Smith, M. “Video Segmentation in the Informedia Project.” In IJCAI-95: Workshop on Intelligent Multimedia Information Retrieval. Montreal, Quebec, Canada, 1995.
13. Mackay, W.E. and Davenport, G., “Virtual Video Editing in Interactive Multimedia Applications.” Communications of the ACM, 32(7):832–843, 1989.
14. Mills, M., Cohen, J., and Wong, Y. “A Magnifier Tool for Video Data,” in SIGCHI '92 Conference Proceedings of Human Factors in Computing Systems. 1992. Monterey, CA: ACM. pp. 93–98.
15. 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 first ACM / IEEE joint conference on digital libraries, 2001, pp. 106–115
16. Placeway, P., et al. “The 1996 Hub-4 Sphinx-3 System,” in DARPA Spoken Systems Technology Workshop. 1997.
17. Schwartz, J., Rhinelander, T. and Dorsey, M., “Personal Rich Media Takes Off”, The Forrester Report, Forrester Research Inc., October 2000.
18. Sony Electronics, Inc. MovieShaker. http://www.ita.sel.sony.com/jump/movieshaker/ms.html
19. Stevens, S.M., Christel, M.G., and Wactlar, H.D., “Informedia: Improving Access to Digital Video.” Interactions: New Visions of Human-Computer Interaction, 1994. 1(4): pp. 67–71.
20. Ueda, H. and Miyatake, T. “Automatic Scene Separation and Tree Structure GUI for Video Editing,” in Proceedings of ACM Multimedia '96. 1996. Boston:
21. Ueda, H., Miyatake, T., Shigeo Sumino and Akio Nagasaka. “Automatic Structure Visualization for Video Editing,” in Proceedings of INTERCHI'93: Conference on human factors in computing systems. 1993. Amsterdam: ACM. pp. 137–141.
22. Wactlar, H.D., et al., “Lessons learned from building a terabyte digital video library.”, IEEE Computer, 1999. 32(2): pp. 66–73
23. Zhang, H., et al. “Video Parsing, Retrieval and Browsing: An Integrated and Content-Based Solution,” in ACM Multimedia 95: Proceedings of the third ACM international conference on Multimedia. 1995. San Francisco, CA: pp. 15–24.