A Magnifier Tool for Video Data

Michael Mills, Jonathan Cohen and Yin Yin Wong

Human Interface Group/Advanced Technology
Apple Computer, Inc.
20525 Mariani Ave., MS 76-3H
Cupertino, California 95014
Mills@apple.com

ABSTRACT
We describe an interface prototype, the Hierarchical Video Magnifier, which allows users to work with a video source at fine-levels of detail while maintaining an awareness of temporal context. The technique allows the user to recursively magnify the temporal resolution of a video source while preserving the levels of magnification in a spatial hierarchy. We discuss how the ability to inspect and manipulate hierarchical views of temporal magnification affords a powerful tool for navigating, analyzing and editing video streams.

KEYWORDS: Interface Metaphors, Time-Varying Data, Hierarchical Representation, Multimedia Authoring, Information-Retrieval, Video Editing, Granularity of Information

INTRODUCTION
Compared to a real desktop, today's electronic desktop is a cramped work area. There is not enough space on a typical computer screen to spread out the actual-size pages of a document nor to show a normal size photograph in its entirety. Hence, one of the classic challenges of human interface design is to overcome the spatial bounds of small, limited resolution screens—to design electronic workspaces which allow users to work with partial, detailed views of objects, while maintaining a sense of context.

In the last few years, there have been a number of innovative interface techniques designed to provide users a sense of simultaneous awareness of detail and context when working with large information spaces. For example, Spence and Apperley [6] built a "Bifocal Display" to help people work with office documents at different levels of granularity — journals, volumes, issues, etc.. In a similar vein, Furnas [2] has experimented with a Fisheye Lens in order to facilitate smooth navigation between detailed and global views of large amounts of information. More recently, Mackinlay and his colleagues [4] have begun to apply techniques of 3D modeling and animation (a manipulable 3D "Perspective Wall")

in order to make the integration of detail and context even more natural: closer to the way the human visual system provides fine fovealdetail, yet maintains awareness of peripheral context.

WORKSPACES FOR TIME-VARYING DATA
Just as users need tools for working with static images and text at different levels of detail, so they need *temporal* magnification tools when working with dynamic data types--movies, animations and sounds. In editing a movie, for example, someone might want to "zoom-in" on a small temporal chunk--say a second's worth of video -- in order to analyze subtle effects of a scene transition. At the next moment, they may want to switch to a high level overview of the movie in order to navigate to a neighboring scene.

Being able to vary the "grain" of detail in viewing an object, whether spatial or temporal, may not in itself provide an effective workspace. The problem is how to give users a sense of orientation--to help them understand how the part they are looking at (a small region of an image, a small chunk of video) relates to the whole. In working with spatial objects, for example, not having a context view can make navigation difficult. It can be a perceptual challenge to locate specific sections of a large image if one is limited to seeing, at any one time, blow-ups of smaller regions: i.e., where the image as a whole must be constructed in the "mind's eye" out of successive partial views [3]. Similarly, it can be hard to navigate, with any precision, to different segments of a long video source if one is restricted to viewing small temporal fragments.

To provide users a sense of spatial orientation when working with large images, many paint programs let them edit magnified, partial views, while providing a "proxy" of the entire image in a separate window or inset. A proxy is a kind of low-resolution surrogate for the real data. (See Moore, *et al* [5] for a discussion of proxies and their interface implications.] In this paper, we describe an interface prototype, the Hierarchical Video Magnifier, whose goal is to address the problem of providing users a sense of *temporal orientation*. How can we enable the user to work with large amounts of time-varying data at fine levels of detail, while at the same time maintaining a sense of temporal context?

Many present-day computer-based video editing systems attempt to provide the user a limited sense of temporal context by using a spatial timeline. This often takes the form of a slider or scrollbar whose horizontal axis represents the total duration of the event. An example is shown in **Figure 1**. Playing the video at normal speed will update the horizontal position of the indicator (the rectangular "thumb") along the timeline at the appropriate rate. If the video is a controllable source such as a laserdisk or digital movie, the user can quickly "scan" the video by dragging the indicator along the scrollbar. If there are fewer pixels constituting the horizontal length of slider than there are frames in the video, the player will "skip" frames during dragging—a form of temporal compression.

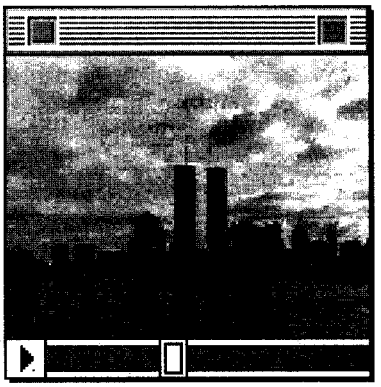

Figure 1. A real-time scrollbar for video data

The scrollbar *qua* timeline serves as a frame-of-reference for the currently displayed video frame (a detailed view). For example, the indicator's position on the timeline helps the user visualize the location of the currently visible frame relative to the total duration of the video. By observing this relationship the user can answer such questions as: "Given that I'm looking at the skyline scene, about how much of the video remains?" Or, "Show me the frame that is half-way into the movie." Hence, the timeline provides a temporal overview which can be used to navigate the video.

While the scrollbar does provide a temporal context, it is an extremely impoverished representation of the video source. Consider such questions as, "Does the carousel scene come before or after the roller-coaster scene?" "Does the stabbing scene occur in more than one place throughout the movie?" "Is the cigarette-lighting scene part of the bedroom scene or part of the eating-dinner sequence?" There is no explicit information on the scrollbar which provides answers to such questions. Nor does the scrollbar give any clues as to where one should click on the scrollbar to find the answers. In sum, the impoverished contextual information on the scrollbar limits its usefulness as a navigational aid. It contains no clues about the content and structure of the video source.

The Hierarchical Video Magnifier attempts to overcome some of the limits of the basic timeline representation described above. It combines a navigable timeline with a richer representation of the video source in order to provide the user a more powerful tool for working with the structure and content of video data.The basic idea is as follows. We begin with a timeline to represent the total duration of the video source. But instead of just using it to navigate a video source at a single level of granularity, we supply the user with: (1) a series of low-resolution video frame samples arrayed along the timeline to give some sense of the video content, (2) a "temporal magnifier"—a tool which can be used to "expand" or "reduce" the effective temporal resolution of any portion of the timeline; and (3) perhaps most uniquely, we maintain a context view — an explicit spatial hierarchical structure of the video source— which results from successive applications of the temporal magnifier.

HOW THE VIDEO MAGNIFIER WORKS

Let us examine how the Hierarchical Video Magnifier might be used to explore the contents of a fairly large video source— 30 minutes worth of full-motion video stored on a controllable laserdisc. (While our example concerns examining the contents of an analog source, the interface technique is general: i.e., could apply to a digital movie or even a waveform representation of a sound). The Magnifier application is currently implemented on a Macintosh-II equipped with a video digitizing card. **Figure 2** shows the initial state of the application when it is opened. There are three main windows. First, is a "video source" window for monitoring the real-time play back of the laserdisc contents. Second, is a software controller for navigating the analog video. Third, is the window containing the hierarchical magnifier.

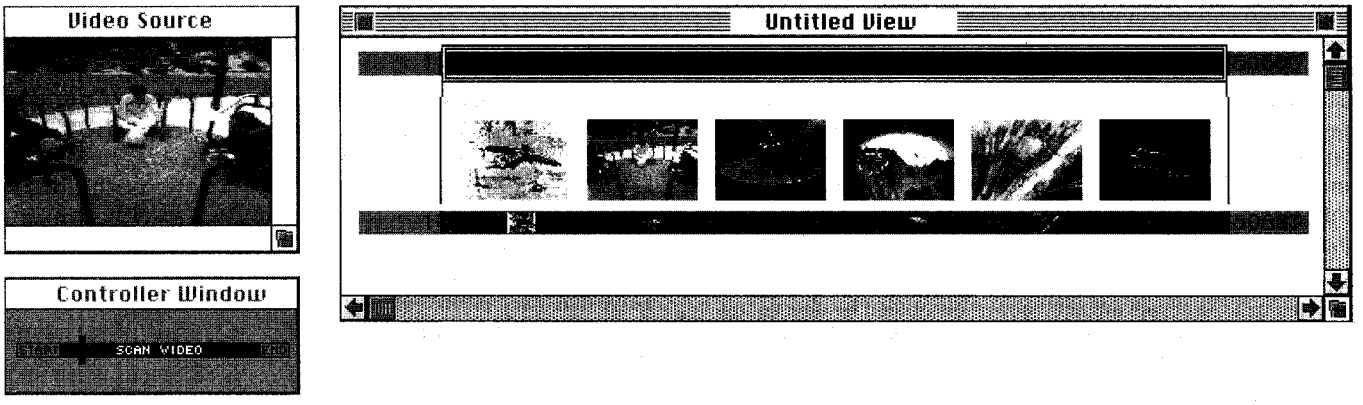

Figure 2. The three main elements of the video magnifier application.

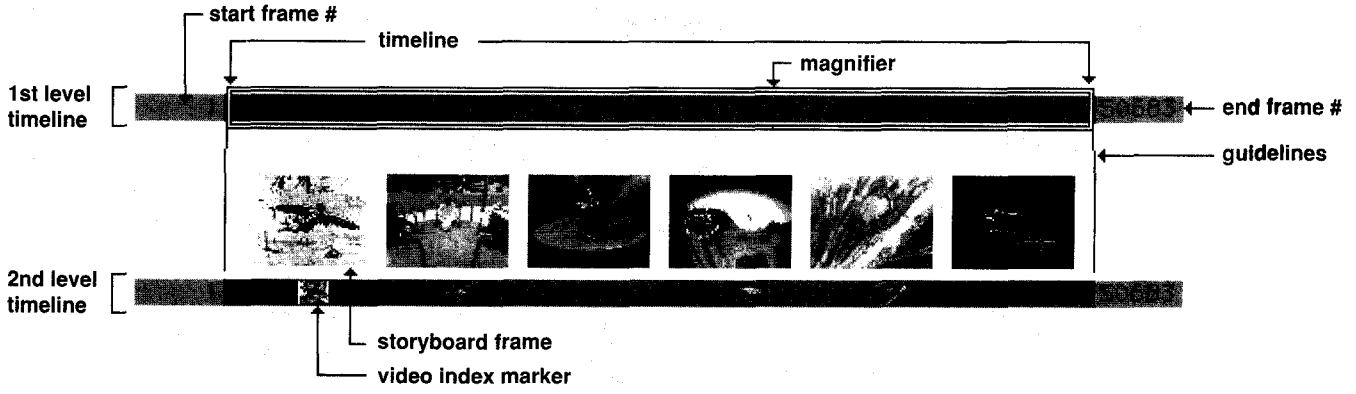

Figure 3. The hiearchical magnifier.

Let's look more closely at the elements of the Magnifier window. See **Figure 3.** At the very top is a timeline—the gray bar — which represents the total duration of the video source; in this case, a laserdisk containing 50,683 frames. The numbers corresponding to the first and last frames are shown at the boundaries of the timeline. A rectangular magnifier surrounds the entire length of this top-level timeline. Note that there are "guide wires" descending from both ends of the magnifier to a second timeline below. This second timeline represents the range of the magnifier. In this case, the magnifier covers the entire length of video. Below the gray magnifier are six low-resolution frames (60 x 80 pixels) equally spaced along the timeline. These six frames represent an extremely compressed view of the entire video source. They have been generated by digitizing still-frame samples at regular intervals from the video stream--in this example a sample for approximately every 8000 frames. This view is not meant to convey the precise temporal relationships among the frames. Rather, it serves as a kind of "storyboard" giving the user some information about the *order* of events across the entire video source. (In future implementations, we plan to use scene change detection algorithms to generate the storyboard frames. By adjusting the threshhold for scene-change detection, we can vary the number of frame samples per row.)

This second level timeline contains miniatures of the larger frames appearing the storyboard view. These miniature frames serve as "video markers"—position indicators which can be dragged along the timeline to "scan" the video source. Dragging a frame marker to the right, for example, will update: (1) the contents of the video source window, (2) the video marker's own contents (display the frame corresponding to its new position along the timeline) and (3) the larger storyboard frame to which it corresponds. By manipulating the video markers, the user can quickly customize the contents of the second-level timeline.

The Temporal Magnifier

In its initial state, the range of the temporal magnifier (the rectangular outline surrounding the top most timeline) extends across the entire length of the timeline. This gives the greatest degree of temporal compression — the coarsest view— of the video source. The six frames have been sampled from roughly 50,000 frames (≈ 8000:1 ratio). By changing the width of the magnifier, the user applies the magnifier to a smaller region of the timeline, as shown in **Figure 4.** This effectively lowers degree of temporal compression (≈4000:1 ratio) and hence, gives a more fine-grained view of a smaller chunk video (≈ 25,000 frames). Note that the contents of the frames, the video index markers and the frame numbers of the second level timeline have been updated to reflect the smaller range of the magnifier. In addition to adjusting its width, the user can position the magnifier over any segment of the timeline. Hence, the magnifier is a tool which can be used to sample the entire stream of video at different effective levels of temporal resolution.

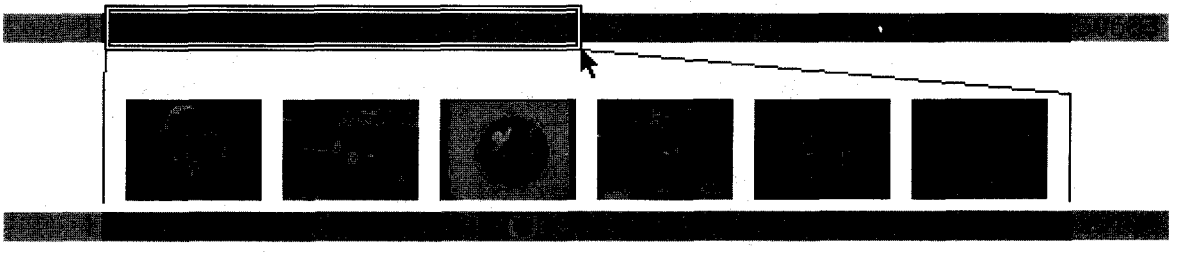

Figure 4. Adjusting the range of the magnifier.

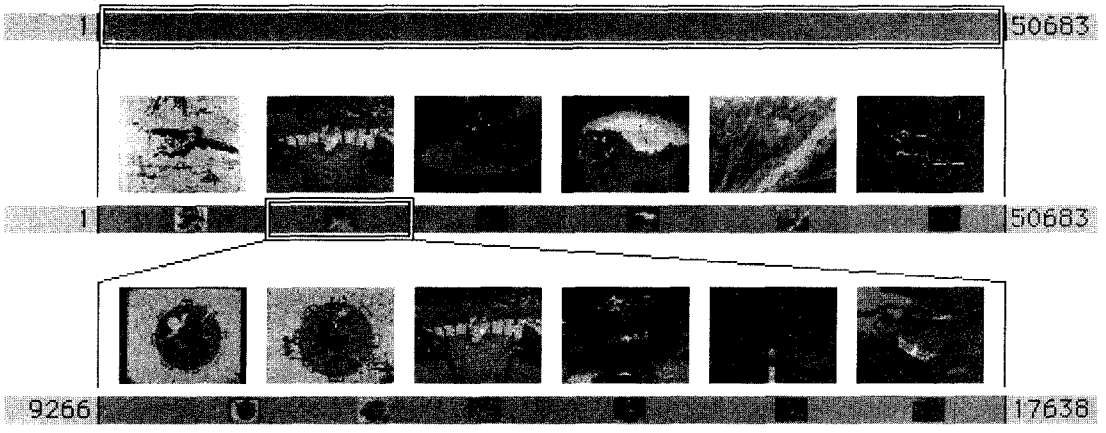

Figure 5. An expanded view of the region surrounded by the magnifier.

Generating Hierarchical Views of Temporal Data

We have seen how the user can vary the temporal resolution of the top-level storyboard. But now s/he may want to "zoom-in" on one part of the timeline—to expand the level of detail surrounding one of the video proxies. Suppose that the user wants to see, in greater detail, what kinds of events surround the carousel scene (the second frame from the left). By clicking on the second-level timeline in the region of the carousel video marker, the user can bring up a new magnifier positioned over this smaller region. See **Figure 5**. A new storyboard and timeline, corresponding to the region surrounded by the new magnifier, appears below. The "guide wires" descending from the ends of the magnifier help the user visually parse the display—see the second level storyboard as an "expansion of detail" of the magnified region.

What is unique about the current approach is that the first storyboard — the coarsest view— is not simply replaced by the second, more finely-detailed storyboard. The first storyboard remains visible and *can be inspected* -- serves as a

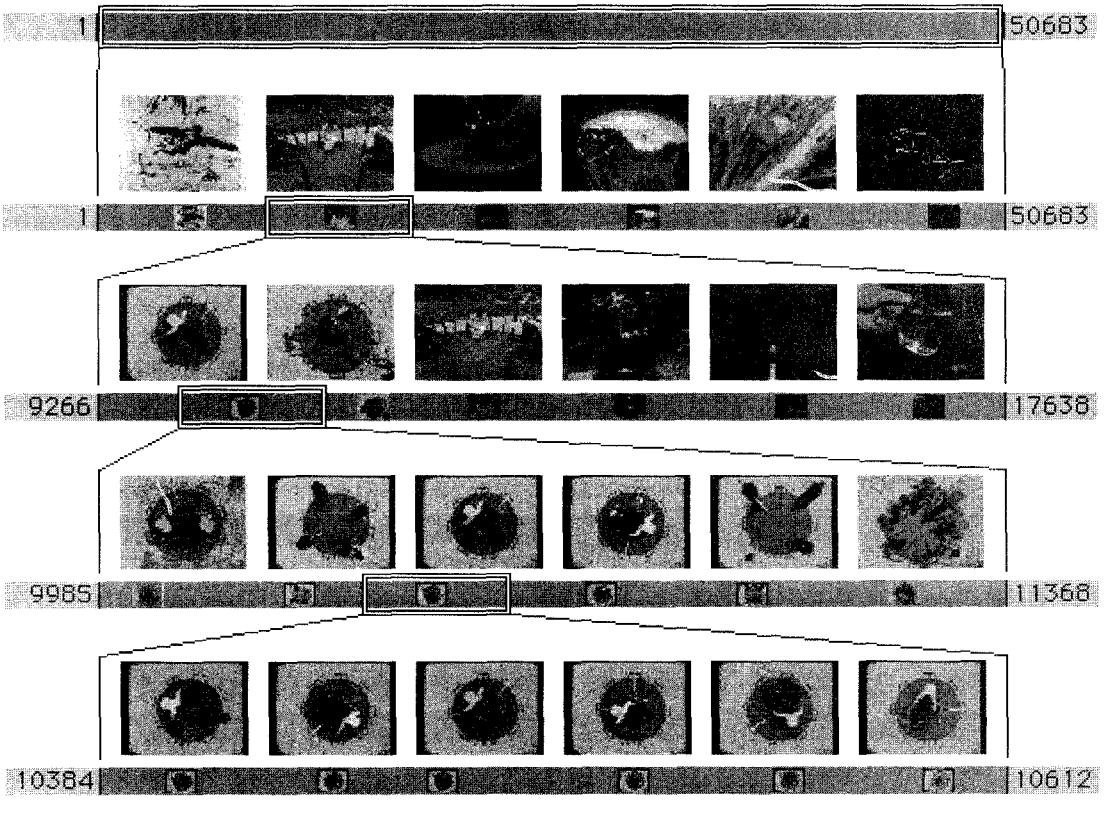

Figure 6. Applying the magnifier two more times.

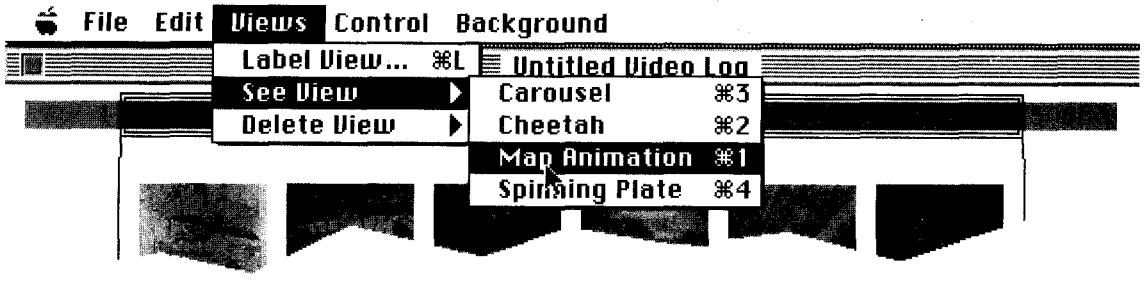

Figure 7. Switching views via the pull-down menu.

frame of reference for the expanded sub-part. Of course, the user does not have to stop after a single level of magnification. In this example, the user could continue to expand the magnification of the carousel event by recursively applying a new magnifier to successive levels. **Figure 6** shows the effects of applying the magnifier two more times. Note that the six frames of the bottom-most storyboard represent a very small chunk of video (about 8 seconds worth), but at a high sampling rate.

HOW WOULD THE VIDEO MAGNIFIER BE USED?

At the very least, the current approach provides a powerful tool for navigating a video source. The primary cognitive benefit is that the user does not have to depend solely on the "mind's eye" to recall what other scenes surround the segment which s/he is currently examining. In **Figure 6**, for example, I can examine the low-level structure of the bottom storyboard (showing a top-view of the carousel event). Yet I can see, two rows up, a that there is a frame showing this carousel scene from a different camera angle--a side view. I could easily expand this side-view scene by repositioning the magnifier on top of its corresponding video marker.

An Iconic Outliner

Beyond navigation, the hierarchal magnifier could be used to construct a customized iconic "outline" of the video source. The top row of frames, maximally compressed, provides a set of iconic "chapter headings"—a high level overview. Successive rows, providing increasingly greater amounts of temporal resolution, serve as pictorial "sub-headings". The last row might be an uncompressed layer, a "continuous" set of frames constituting a short video chunk — the equivalent of a video "word" or "phoneme". To build an iconic outline of a large video source will require more than a single hierarchical view. Hence, the application allows the user to label each view and store it in a list. The user can switch among views on the list by selecting its label under a pull-down menu. **See Figure 7.**

A powerful feature of the application is that the hierarchical views do not just provide information about the video source; they can also be used to *control* it. See **Figure 8.** For example, suppose the user wanted to play a real-time segment in the video window which corresponds to a given video frame on a storyboard. To do this, the user does not have to punch in a frame number nor enter time code to reset the laserdisc. The user simply drags a copy of the particular frame from the storyboard and drops it into the video source window. (8a) Because the digital frame has stored information about where it came from (its location in the original video source), it can reset the laserdisk to begin playing at this location. (8b) In this way, someone could customize a set of views — build an iconic log or outline — of a source which could be used as a front-end to navigate the video without having to deal with frame numbers. Finally, the ability to use the hierarchical views to control a video source would make the magnifier a useful component of a video editing system. For example, frames from any row could be selected to define "begin" and "end" points of a segment which could then be digitized from the original video source.

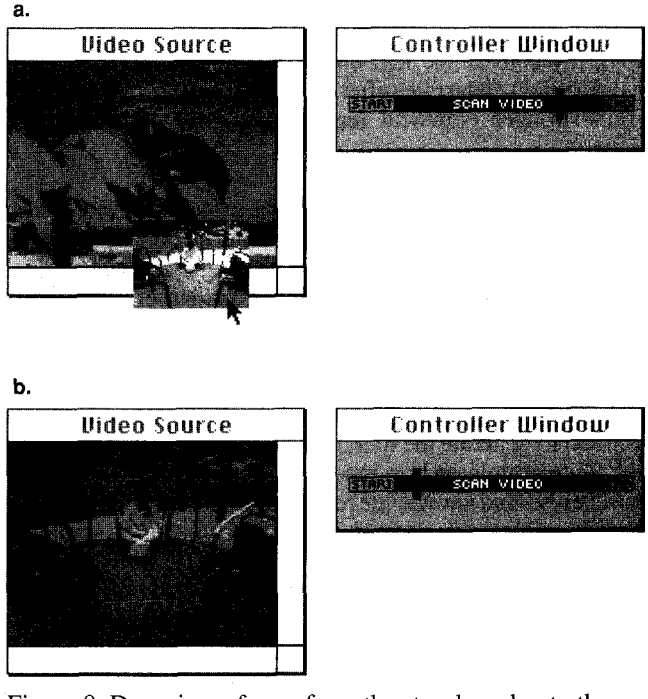

Figure 8. Dropping a frame from the storyboard onto the video source window (a) resets the laserdisk to begin playing at this location (b).

CONCLUSION

We have described a magnifier tool for video data—one which allows users to vary the temporal resolution of a video source, while maintaining an awareness of the hierarchical structure of the levels of magnification.

Preliminary user testing of the technique indicates that while people grasp the basic concept—the recursive magnification of portions of a timeline to generate hierarchical structure — other parts of the interface need to be improved. For example, in the current implementation, people have some difficulty parsing the display—especially understanding how the miniature video markers on the timelines map to their corresponding frames in the storyboard view. This difficulty arises because a video marker, which can be dragged along the timeline, can become spatially displaced vis-a-vis its matching frame on the storyboard. For example, instead of appearing directly beneath its matching frame, a video marker might be dragged to a position almost directly beneath an adjacent, but non-matching, storyboard frame. In this case, the user's eye is presented with a perceptual dilemma--a contest between matching content vs. spatial proximity. Does it perceive the miniature marker as being paired with the spatially displaced frame on the basis of identical video content? Or does it pair the marker with the spatially proximal, but non-identical frame, under which it appears. We need to investigate better methods of helping users perceive the links between video markers and their matching frames.

We also need to explore methods for strengthening the graphical depiction of hierarchical nesting. At present, the only graphical devices used to convey "belongingness" are the "guide wires" which emanate from the edges of the magnifier to the next-level timeline. Frame numbers at either ends of the timelines specify segment boundaries. But the numbers do not graphically convey the fact that a particular timeline is an expanded view of the magnifier to which it belongs. We need to find stronger graphical representations of hierarchical structure.

Another avenue for future work might be to look at how recent techniques (Perspective Wall, Fisheye Lens) mentioned earlier might enhance the representation. For example, mapping the 2D hierarchical view to a 3D Perspective Wall might allow us to fit more levels of magnification in the available window space.

Finally, the present technique uses only the metric of temporal resolution to generate the hierarchical structures. It might be worthwhile examining how recent work in image processing, scene analysis and knowledge representation could help build richer hierarchical descriptions. Some work has already started along these lines. For example, Ueda, *et al* [7], have built a multimedia authoring system which uses motion and object detection algorithms to generate editable storyboard views of a video stream. In a related vein, Davenport, *et al* [1], are investigating the use of enhanced camera and sound input techniques to annotate video streams: to enable the user to attach, at the time of recording, information about semantic and structural variables (camera position, lighting, causal actions, narrative structures). The goal is is to represent these annotations in "strata" -- hierarchical layers -- which can be used to enhance video editing and multimedia browsing applications.

Bringing together richer semantic representations of a video source with the ability to manipulate hierarchical temporal views, should lead to even more powerful workspaces for video material.

REFERENCES

1. Davenport, G., Aguierre-Smith, T., and Pincever, N. Cinematic primitives for multimedia. 67-75. *IEEE Computer Graphics and Applications., 11(4,)*, July 1991.

2. Furnas, G.W. Generalized fisheye views. *Proceedings of CHI '86 Human Factors in Computing Systems*, 16-23. New York: ACM, 1986.

3. Hochberg, J. In the mind's eye. In R.N. Haber (Ed.) *Contemporary Theory and Research in visual perception.* New York: Appleton-Century-Crofts.

4. Mackinlay, J.D., Robertson, G. G., and Card, S. K. Perspective wall: detail and context smoothly integrated. *Proceedings of CHI '91 Human Factors in Computing Systems*, 173-179. New York: ACM, 1991.

5. Moore, R., Morrison, J. and Oren, T. Proxies and their applications. *Apple Viewpoints.* July 10, 1989. Apple Computer, Inc.

6. Spence, R. & Apperley, M. Data base navigation: An office environment for the professional. Behaviour and Information Technology 1 (1), 43-54, 1982.

7. Ueda, H., Takafumi, M. and Yoshizawa, S. Impact: An interactive natural-motion-picture dedicated multimedia authoring system. *Proceedings of CHI '91 Human Factors in Computing Systems*, 343-350. New York: ACM, 1991.