MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Augmenting Fast-Forward and Rewind for Personal Digital Video Recorders

Ajay Divakaran, Clifton Forlines, Tom Lanning, Sam Shipman, and Kent Wittenburg

TR2004-136   January 2005

Abstract

This paper describes a set of interfaces for augmenting fast-forward and rewind on consumer digital video recorders. Our method overlays a series of images sampled from the video over top of the traditional full screen accelerated playback. This sequence creates a trail that provides contextual information and highlights upcoming scene changes in the video stream. With this augmentation, consumers are more accurate at traversing to a desired location in a recorded video. This advantage is achieved by taking advantage of compressed-domain processing and adds little computational and storage overhead.

The International Conference on Consumer Electronics, (ICCE)

Augmenting Fast-forward and Rewind for Personal Digital Video Recorders

Ajay DIVAKARAN, Clifton FORLINES, Tom LANNING, Sam SHIPMAN, and Kent WITTENBURG
Mitsubishi Electric Research Laboratories, Cambridge Massachusetts, USA

***Abstract*— This paper describes a set of interfaces for augmenting fast-forward and rewind on consumer digital video recorders. Our method overlays a series of images sampled from the video over top of the traditional full screen accelerated playback. This sequence creates a trail that provides contextual information and highlights upcoming scene changes in the video stream. With this augmentation, consumers are more accurate at traversing to a desired location in a recorded video. This advantage is achieved by taking advantage of compressed-domain processing and adds little computational and storage overhead.**

I. INTRODUCTION

Digital consumer video devices have all but replaced their analogue ancestors, and have brought with them the benefits of smaller device sizes, lower costs, and larger storage mediums. While these measurements have improved from this analogue to digital transition, many portions of the interface for accessing recorded video have remained the same. Some notable exceptions include what are the most exciting changes to video browsing interfaces; changes that have come from exploiting the qualities of digital media, and include such features as browsing by chapters on a DVD and the 30-second skip ahead button found on personal video recorders; however, the basic method for fast-forwarding, browsing in full screen one frame at a time, has remained the same.

With an analog tape, a device can only display an image from one position in the video at a time. This constraint is lifted for random-access digital video devices that can display multiple portions of a recorded video stream at the same time. We have found that this type of simultaneous-points-of-playback interface especially helpful for the tasks of fast-forwarding and rewinding through recorded video. In addition to the familiar accelerated playback of the video that is provided by VCRs, PVRs, and DVD players, our interface simultaneously shows a number of upcoming and previous frames from the video stream in the periphery of the screen (Figure 1). By presenting multiple frames from the video stream simultaneously, we give the consumer a view into the future of the video they are traversing. Camera movement and scene boundaries are clearly visible and aid in the navigation. With this extra information, consumers are better able to identify upcoming points of interest in a recorded video, and are able to more quickly and more accurately traverse to a desired location in the stream.

II. TECHNICAL DESCRIPTION

Capturing and rendering full-screen high-definition images is computationally expensive and requires large amounts of storage. Rather than trying to decode multiple portions of the recorded video stream simultaneously or trying to store a collection of full-resolution images while capturing video, our approach takes advantage of compressed-domain processing. For I-Frames, the dc value of each 8x8 block can be read directly from the compressed bitstream without having to carry out computationally expensive inverse DCT's, which enables simple generation of an 1/8 scale version of the full-size image, termed a dc-image [2]. By storing these lightweight dc-images to disk while we record video content, we give ourselves a means for presenting sequential low-resolution images during playback with very little additional CPU or storage overhead. Note that the color information is at quarter resolution, so we have to appropriately convert the dc images before displaying on screen. Note that since the images are small compared to the frame size, the computational overhead added by scaling them is small. Our product platform provides the required scalers. The images do not need to be decoded when shown, and the encoding is already taking place for the purpose of recording the MPEG stream. We are effectively "piggybacking" on an already occurring operation. Additionally, the storage of these images adds only 5-10% to the file size of a recorded program.

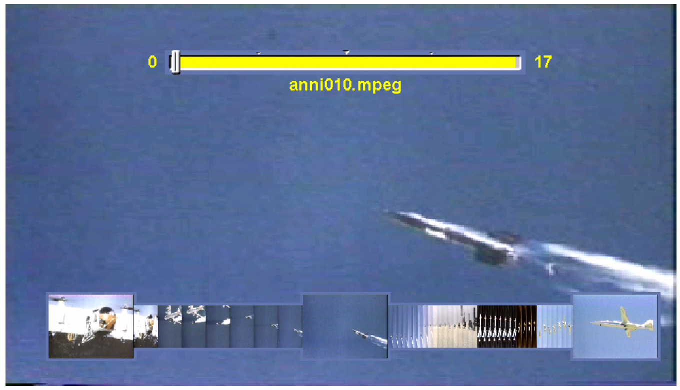

Fig. 1. The familiar full screen, frame-at-a-time fast-forward is displayed in the background of the screen. In the foreground, a timeline displays upcoming and previous images from the video stream. In this example, the middle frame is in sync with the background fast-forward frame.

Despite the piggybacking, it is a challenge to implement the parsing of the compressed bitstream on the CPU since its resources are occupied by the management of the entire application Furthermore, the memory footprint also has to be within the limits of the platform. We have been working on optimizing our implementation to meet such constraints.

Another challenge is to deal with the bandwidth and other constraints placed by the interface to the hard drive on which our content is stored. Our algorithm lends itself well to adaptation to such constraints since it adds a very small bandwidth overhead since the images are small.

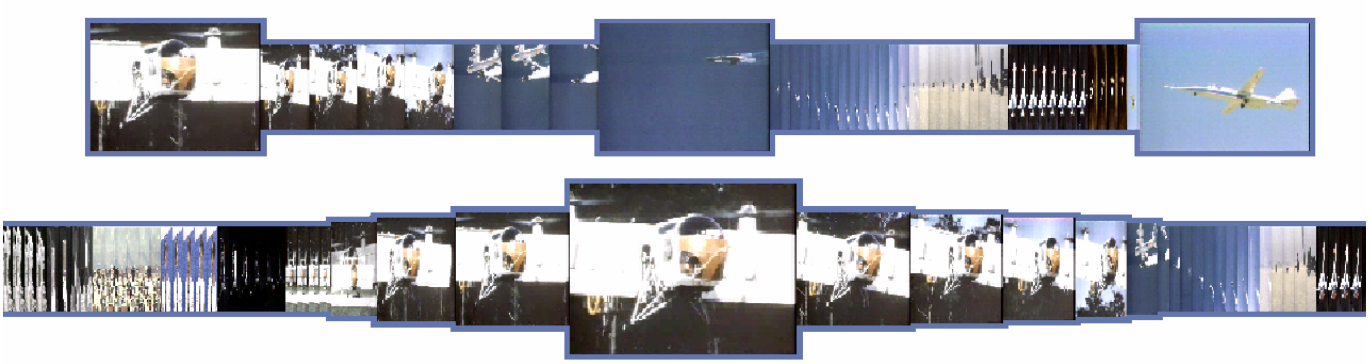

Fig. 2. Two of the many timeline variations evaluated. Timelines such as those above were overlaid on top of a traditional full screen fast-forward / rewind interface. The “Squeeze” layout (top) shows one DC image 30 seconds ahead of the current frame on the right of the timeline, and one image 5 seconds behind the current frame. DC images between these frames are squeezed into the space provided. The “Fisheye” layout (bottom) shows the current frame at full size, and gradually scaled the DC images to the left and right. Even with the highly scaled images on the ends of the layout, scene changes are immediately visible.

III. User Evaluation

Our prior user testing with pre-computed full-size images revealed that subjects were significantly more accurate using our technique than when using a standard VCR interface. Subjects were asked to perform typical fast-forwarding tasks such as skipping through a block of commercials, or fast-forwarding to the weather report while watching the evening news. On average, subjects resumed playback 25% closer to the intended position (e.g. the first frame after a set of commercials) when using our augmented fast-forward than when using the standard fast-forward. A detailed description of the results of this study is found in [1].

For this paper we used actual DC Images taken from "worst-case" standard-definition video and ran the interface on a PC that mimicked the CPU and memory constraints of a consumer-level personal video recorder. The DC images extracted from the MPEG-2 video were stored at 1/8 the resolution of SD video, 80x60 pixels. For the 40-minute 2-gigabite video files used in these experiments, these images required an additional 90-megabites of storage space (an addition of about 4.5%). The experiment had subjects traversing videos using a variety of interfaces, and focused mainly on soliciting preferential feedback on the many variations in layout.

While differences in preference existed among the many variations used, all of our subjects could successfully traverse to a desired location in a video stream using a trail of these low-resolution images. The experiment included variations in which the full screen background was omitted; however, subjects strongly preferred variations that overlaid the trail of images on top of the traditional fast-forward interface. Additionally, several subjects stated that different layouts would be more or less useful for different types of content. Specifically, the horizontal resizing of images was popular for content that includes long, fixed camera shots, such as sporting events and dramas, but deemed less appropriate for content that includes rapid scene changes and aggressive camera movement, such as commercials and music videos.

We will continue with this cycle of testing and redesign as we work toward identifying the optimal layout with the design space.

IV. Conclusion

We presented a novel approach to browsing of stored video content that preserves temporal context by displaying past and future frames in reduced size. We found that it enables greater accuracy in getting to a desired location using trick play as compared to traditional fast-forward/rewind. We have adapted the proposed technique to fit the severe constraints of the target platform by using dc images obtained through simple parsing of the digital video stream. We find that our adaptations of our original algorithm do not adversely affect the effectiveness of the browsing interface. We are using the results of our ongoing user studies to improve our system.

In future work, as our product platform becomes more computationally powerful, we will further enhance the browsing by introducing a 3-D approach to displaying the time-tunnel, by using larger images and by incorporating other video browsing techniques.

V. Acknowledgment

We would like to thank our colleagues Mr. Masahayu Hayakawa, Mr. Atul Batra and Mr. Mark Flynn of Mitsubishi Electric Digital Electronics, America for their support and encouragement.

References

[1] K. Wittenburg, C. Forlines, T. Lanning, A. Esenther, S. Harada and T. Miyachi, "Rapid Serial Visual Presentation Techniques for Consumer Digital Video Devices," *Proceedings of the 16th annual ACM symposium on User Interface Software and Technology*, 2003, pp. 115-124.

[2] Boon-Lock Yeo and Bede Liu, "Rapid scene analysis on compressed videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, Dec. 1995, pp. 533-544.