Temporal-Context-Based Video Browsing Interface for PVR-Enabled High-Definition Television Systems

Sam Shipman, Ajay Divakaran, Mark Flynn, Antul Batra

Abstract

We describe an implementation of our previously developed temporal-context-based video browsing interface, on a PVR-enabled high-definition television system. The implementation poses significant challenges such as consuming minimal computational resources so as to avoid disrupting any of the existing functionalities, sampling the stored video frames using a 1394 AV/C connection, and rendering the sampled images in a visually pleasing and useful way. Our implementation successfully addresses the aforementioned challenges by taking advantage of the target platform's facilities for high-quality video scaling and frame sampling.

IEEE International Conference on Consumer Electronics (ICCE)

Temporal-Context-Based Video Browsing Interface for PVR-Enabled High-Definition Television Systems

Sam SHIPMAN[1], Ajay DIVAKARAN[1], Mark FLYNN[2], and Atul BATRA[2]

[1]Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA

[2]Mitsubishi Digital Electronics America (MDEA), Irvine, CA, USA

Abstract—**We describe an implementation of our previously developed temporal-context-based video browsing interface, on a PVR-enabled high-definition television system. The implementation poses significant challenges such as consuming minimal computational resources so as to avoid disrupting any of the existing functionalities, sampling the stored video frames using a 1394 AV/C connection, and rendering the sampled images in a visually pleasing and useful way. Our implementation successfully addresses the aforementioned challenges by taking advantage of the target platform's facilities for high-quality video scaling and frame sampling.**

I. INTRODUCTION

In a previous paper [1], we presented a method of browsing stored digital video in fast-forward and rewind modes that provides temporal context by displaying trails of overlapping sequential future (and past) images in reduced size (Figure 1). We described a PC-based prototype implementation and reported results of user studies showing that subjects achieved greater accuracy in navigating to a desired location than they could using traditional fast-forward and rewind functions.

Fig. 1. Temporal-Context-Based Video Browsing Interface

The remainder of this paper explains how we adapted this feature to a consumer product, a high-definition television equipped with an internal digital Personal Video Recorder (PVR) that uses a hard disc drive as the storage medium for recorded digital video. We discuss implementation challenges and how they were met.

II. INTERFACE DESCRIPTION

The browsing interface is invoked by pressing the FF/FWD or REW/REV key on the television's remote control while the hard disc is selected as the input source. An example of the picture displayed on the television screen during fast-forward playback is shown in Figure 1. The images are arranged in an asymmetric V layout and range in size from 1/9 that of the full screen (1/3 scale) at the lower vertex of the V, to 1/16 that size (1/12 scale) at each end.

New images are first displayed at the upper right end of the V and make their way down the right side of the V to the vertex. The image displayed at the vertex, called the focus image, is considered the current image, representing the point at which normal-speed full-screen playback resumes when the viewer presses PLAY. Images to the right of the vertex are of the future with respect to the focus image; those to the left are of the past. No other image overlaps the focus image; watching it is roughly equivalent to watching traditional fast-forward video at 1/3 scale. Although the future images overlap, scene boundaries and camera movement are clearly visible, which helps the viewer identify upcoming points of interest in the video and react accordingly. (In Figure 1, a scene boundary is visible a little less than half-way down the right side of the V.) The layout is asymmetric because it is more useful to see farther into the future than the past. During fast-reverse playback, the V layout is reversed end-for-end; new frames come in at the left end of the V and exit to the right.

III. IMPLEMENTATION

The implementation of temporal-context-based video browsing must

- obtain periodic sequential samples of reduced-size images that are in the "future" with respect to the current image
- store the last *n* of these images, where *n* is the number of images displayed in the V layout
- periodically render the V layout display from the stored images at least five times per second
- return to normal-speed playback when the viewer presses PLAY

In [1] we suggested an implementation that obtains the sample images by reading ahead in the MPEG-2 stream and extracting dc-images (1/8 scale images easily extracted from the compressed data [2]) from the I-frame pictures. However, the PVR in our target platform uses IEEE 1394 Audio/Video Control (AV/C) discs [3] to store video. AV/C discs provide a high-level interface for recording and playback of audio and video content. For playback, in normal and fast-forward and rewind trick-play modes, the AV/C disc provides video streams only at the real-time rate using 1394 isochronous

transfer protocols. This makes it impossible to read ahead in the input during playback.

It was a basic requirement that our implementation have as little impact on the structure, function, and performance of the existing system as possible. Buffering the MPEG-2 video stream before decoding (to accumulate "future" video) would have added another complex step to the critical path for decoding.

The target platform already supports high-quality video down-scaling, so we decided to use that functionality to obtain the reduced-size images for use as focus (vertex) images and to use the graphics scaler to reduce them further for display in either side of the V layout. Hardware and software support for sampling the current decoded and scaled-down video image and converting it from YUV format to RGB graphics format is also provided (to support an existing feature of the television, the Picture-Outside-Picture or POP screen format).

In lieu of read-ahead or buffering of the MPEG-2 video stream, we buffer the sampled 1/3 scale images, storing in a circular buffer as many as needed to fill the V layout. The V layout display is constructed entirely as a graphics overlay, so synchronization with the decoded video output is not required. The time required to fill the V layout with images at the current sampling rate is sufficient to buffer a useful number of future images (11 or so). The fill time is short (about 2 seconds) and the display is visually interesting, so the typical viewer is willing to tolerate the slight delay.
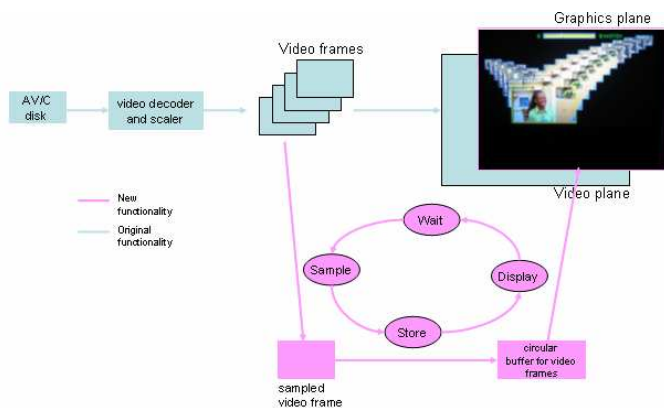

Fig. 2. Implementation Diagram

Aside from the hardware-supported graphics operations referred to above, all computation for the browsing interface is performed on the platform's general purpose CPU. A substantial fraction of the CPU's capacity is already used in support of decoding and for other functions. To avoid disrupting any existing functionality, the thread of execution for the browsing interface was assigned an appropriate scheduling priority and allowed to free-run (up to a maximum rate enforced by a timer event). The exact update rate and timing are not critical, and in practice a usable update rate (currently about 7 Hz) is achieved, and jitter is not noticeable.

Because the playback position in the video stream is a few seconds ahead of the focus image, the playback position must be adjusted to correspond with that of the focus image when the viewer presses PLAY to return to normal-speed full-screen playback. Each sampled image is time-stamped as it is sampled and the difference between the time-stamps of the focus image and the image at the upper end of the "incoming" side of the V is passed to the AV/C device control function, which uses that value to adjust the playback position before switching to normal-speed playback.

IV. CONCLUSION

We successfully adapted our temporal-context-based video browsing interface to a consumer product, implementing it as a feature of a high-definition television equipped with an internal PVR. We were able to leverage the features of the target platform to create an implementation quite different from that of our previous prototype, but better suited to the strengths and weaknesses of the platform. Because the feature requires no additional hardware, it incurs no additional manufacturing cost.

V. ACKNOWLEDGMENT

REFERENCES

[1] Divakaran, A., Forlines, C., Lanning, T., Shipman, S., Wittenburg, K., "Augmenting Fast-forward and Rewind for Personal Digital Video Recorders", *IEEE International Conference on Consumer Electronics (ICCE), Digest of Technical Papers*, pp. 43-44, January 2005.
[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.
[3] 1394 Trade Association Document 2002001, AV/C Disc Subunit General Specification 1.2, September 13, 2002.