Support Backward Interactive Functions of MPEG –2 Stream

Kostas Psannis Marios Hadjinicolaou
Department of Electronic & Computer Engineering
Brunel University
Uxbridge, Middlesex, UB8 3PH
United Kingdom

Abstract: - In this paper, we present an efficient approach for supporting rewind operations in MPEG-based stream. This approach is based on transcoding the Predicted (P)- frames of the original MPEG stream as intra coded (I)-frames in order to create an "interactive" stream. After a P-frame is retrieved, transmitted, decoded and played out, we encode this frame as a sequence of I-frames and store it back to an additional storage at the client. Transcoding only the Predicted (P)- frames of the primary stream to generate the secondary "interactive" stream results in less frame rate of the secondary stream. The frame rate of the secondary stream can be artificially increased by including so called P (M) marionette frames .The effectiveness of our approach in terms of reducing the allocated bandwidth of the interactive stream is evaluated through extensive simulations. Moreover the visual quality during the interactive mode is analyzed.

Key-Words: - MPEG- Reverse/Backward Operations--VCR control -Scanning operations- Video Browsing

1 Introduction

The rapid progress in high speed computing, networking, storage devices, compression techniques and broadcasting technologies, make it feasible for computer systems to deliver interactive multimedia services to end users via high-speed networks. Interactive television is a technology for delivering on demand television programs to households. People have been passive participants in receiving what TV service provides offer since televisions was introduced. Video On Demand (VOD) is a service that provides movies on an individual basis to television sets in people homes. VOD is highly supported by Hollywood since it can lead to new markets and can bring them unpredictable profits. Interactive Video On Demand (IVOD) provides users with flexibility in choosing kinds of information they like to receive An IVOD is basically an extension of VOD. In addition to the freedom of choosing movies, the exciting is that the user can interact with movies and decide the viewing schedule. In an interactive system additional functionalities are implemented.

This research was supported in part by Multimedia Systems & Parallel Computing Technology Research Center at Brunel University, Uxbridge, Middlesex, UB8 3PH, UK and British Council

These functionalities impose new requirements and challenges on the system implementation. Possible interactive functions include play/resume, stop, pause, Jump Forward (JF)/ Jump Backward (JB), Fast Forward (FF)/ Fast Rewind (FR), Slow down (SD), Reverse and Slow Reverse (SR). The difficulty of supporting interactivity varies form one interactive function to another. A stop or pause followed by resume are relatively easy to support since they don't require more bandwidth than that is already allocated for normal playback. On the other hand, fast playback operations {(FF), (FR), (JF), (JB)) involves displaying frames at multiple times the normal rate. When a P-/B frame is requested all the previous P-I-frames need to be sent over the network and decoded by the decoder. This requires the network to send all the related frames besides the actually requested frame at a much higher rate, which can be many times of that required by the normal playback. In addition when many clients request the interactive operations, it may result in much higher network traffics compared to the normal forward play mode and also require high computational complexity in the decoder module to decode all the additional frames. Hence, transporting and decoding frames at such rate is prohibitively expensive and is not feasible with today's hardware decoders. Backward and fast backward operations are even more difficult to support on compression schemes that involve motion-interpolated frames such as B-frames in MPEG scheme. A straightforward implementation is for the decoder to decode the whole (GOPs), store all the decoded frames in a buffer and play the decoded frames in the reverse order. The main drawback to this implementation is that will required an enormous buffer (N frame number- N is the size of a GoPs) in the client set top box to store all the decoded frames. Another approach is to decode the GoPs up to the current frame to be displayed and then go back to decode the GoPs again up to the next frame to be displayed This scheme does not require large buffer but needs the decoder to operate in tremendously high speed (N times more of the normal playback) which is not practical when the N (GoPs size) is large.

Several approaches have been proposed to support interactivity in an MPEG-based compressed video streams. [1] describes a method of transforming an MPEG compressed bitstream in to and I-B bitstream by performing a P-to-I frame conversion in order to break the inter-frame dependencies between the P-frames and I-frames. However, this scheme requires higher decoder complexity to perform the P-to-I conversion and additional storage. Another method to implement reverse play presented at [2] divides the incoming I-B-P bitstream into two parts: I-P frames and B frames. A transcoding process is then used to convert the I-P frames into another I-P bitstream with a reverse order. Alternatively interactive functions can also be supported using separate copies of the movie that are encoded at lower quality of the normal playback copy [3], [4]. In addition interactivity can be supported by transmitting additional data of the same movie at the server to the Digital Storage Device (DSD) located at the customer premises [5] [6]. It is worth to point out that none of these schemes mentioned above support variable speedups rewind functions and also fully address the problem of extra network bandwidth and decoder complexity caused by implementing rewind operations.

In this paper, we introduce an efficient method to implement rewind interactive functions in an MPEG-2 based video stream. Similar to [7], our approach is based on transcoding the predicted frames of the MPEG-2 original stream as a sequence of I-frames in order to support backward interactive functions. Note that backward operations are desirable in many applications, e.g. Interactive Video On Demand System, Digital VCRs and DVD players. The rest of the paper is organized as follows. In Section 2 we discuss the MPEG-2 structure. In Section 3 describes the interactive bitstream and the rate control schemes. Section 4 presents the additional storage comparisons. In Section 5 outlines the variable supported speedups. In Section 6 the visual quality is analysed through simulations. Finally, conclusions and some points to open research are given in Section 8.

2 MPEG-2 Structure

An MPEG-2 sequence is typically partitioned into small intervals called GoPs (Group of Pictures), which in turn are categorized by I (intracoded or intrapicture), P (predicted), and B (bi-directional predicted) [8]. The number of bits per GoP is distributed such that allocation for an I-picture is more than that for P-picture This is because a P-picture uses motion estimation (ME) technique to estimate its content; as a result, a motion-compensated frame difference (MCFD) with lower entropy than the original source is encoded. B-pictures use the smallest number of bits because their (ME) techniques are more intensive than those for P-pictures

Let TF be the Total (T) number of Frames (F) in the normal version. The number of I-P-B frames can be defined as follows

$$I = \frac{TF}{N} \tag{1}$$

$$P = TF \times \left(\frac{1}{M} - \frac{1}{N}\right) \tag{2}$$

$$B = TF \times \left(1 - \frac{1}{M}\right) \tag{3}$$

where,

- N is the distance between two successive I frames, defining a “group of pictures” (GoP).
- M is the distance between consecutive I or P frames. Usually set to 3.

3 Backward operations at the client

In order to make possible backward operations (backward, fast/jump backward and slow backward), we propose a transformation of the MPEG-2 stream at the client end. Specifically, after a P-frame is retrieved, transmitted, decoded and played out, we encode this frame as a sequence of I-frames and store it back to an additional storage at the client. The uncompressed (P) frame is re-encoded as a sequence of I-frames with different encoding parameters. As Inter to Intra conversion is performed after a P-frame is decompressed and play-out no extra cost required for decoding.

Transcoding only the Predicted (P)- frames of the primary stream results in less frame rate of the secondary stream, increasing the decoder requirement. To minimize the decoder complexity, the frame rate of the secondary stream can artificially increased by including so-called P (M) marionette frames. Effectively this results in repeating the previous I-frame in the decoder. By transcoding the Predicted (P)-frames frames to Intra (I)-frames the bandwidth of the generated "interactive" bitstream is too big. To generate scan versions with minimum extra storage at the client the interactive bitstream must be rate control

3.1 Rate Control Schemes

To minimize the size of the interactive stream the transcoded (P)-frames as (I)-frames must be rate-controlled. A common approach to control the size of an MPEG frame is to vary the quantization factor on a per-frame basis. This result in variable video quality during the Rewind modes (however the quality is still constant during the normal version). The amount of quantization may be varied. This is the mechanism that provides constant quality rate control. The quantized coefficients $QF(u,v)$ are computed from the DCT coefficients, the quantization scale $MQUANT$, and a quantization_matrix $W(u,v)$, according to the following equation

$$QF(u,v) = \frac{16 \times F(u,v)}{MQUANT \times W(u,v)} \qquad (7)$$

The normalized qunatization factor is

$$w(u,v) = \frac{MQUANT \times W(u,v)}{16} \qquad (8)$$

The quantization step makes many of the values in the coefficient matrix zero, and it makes the rest smaller. The result is a significant reduction in the number of coded bits with no visual apparent differences between the decoded output and the original source data.
The quantization factor may be varied in two ways

- Varying the quantization scale ($MQUANT$)
- Varying the quantization matrix ($W(u,v)$)

For $W(u,v)$ =1 for all the u, v we have loss-less encoding.

To bound the size of new $I_{transcoded}$ -frames of the interactive mode, the encoder uses two predefined (higher-lower) thresholds. An I-frame is re-encoded such that its size is between

$$Threshold^{lower} \leq I_{transcoded} \leq Threshold^{higher} \qquad (9)$$

After a frame of a scan version has been re-encoded as an I-frame the encoding algorithm checks whether the size of the compressed frame is between the two pre-defined thresholds. If it is not, then the quantization factor for the corresponding frame is increased/decreased and the frame is re-encoded.

Two different schemes can be used to initialise the quantization factor when a frame is to be re-encoded. In the first scheme, when a frame is to be re-encoded for the first time the encoder starts with a fixed quantization factor. The main problem with this method is the quantization value might be kept unnecessarily high resulting in low quality during the interactive operations. Moreover the encoder in this scheme uses one predefined threshold ($I_{transcoded} \leq Threshold^{higher}$)

In the second scheme, the encoder tries to track the nominal quantization value, which was used in encoding the same type of frame in the normal version. After the first encoding attempt, if the resulting frame size is between the two pre-defined thresholds (9), the encoder proceeds to the next frame. Otherwise, the quantization factor {quantization matrix ($W(u,v)$ } varies and the same frame is re-encoded. The quantization matrix can be modified by maintaining the same value at the near-dc coefficients but with different slope towards the higher frequency coefficients. This procedure is repeated until the size of the compressed frame is between the two predefined thresholds (9). The advantage of this scheme is that it tries to produce interactive operations with the same constant quality of normal playback, but when it is not possible it minimizes the fluctuation in video quality during the rewind mode. On the other hand, since re-encoding is done online the encoding time may be in more important than the video quality.

4 Comparison the Extra Storage

It can be shown that the additional storage required due to Predicted- Intra Sequence (P-IS) conversion is small. The storage requirement at the server for normal playback is given by

$$W_{server} = TF \times Average(IPB)Size \qquad (10)$$

where,

$$Average\text{(IPB)Size} = \frac{\mathrm{I}_{\text{average}}}{\mathrm{N}} + P_{average}\,\mathrm{x}\left(\frac{1}{\mathrm{M}} - \frac{1}{N}\right) + B_{average}\,\mathrm{x}\left(1 - \frac{1}{M}\right)$$

The extra-storage at the client-end using the proposed schemes for the interactive mode is given by

$$W_{P \to IP(M)} = TF^{Interactive}_{I,P(M)} \; x \; (P - IS) \; Size \qquad (11)$$

where,

$$(P-IS)Size = \begin{cases} \text{Average(P - IS)} & \text{1st - algorithm} \\ \text{Constant(P - IS)}, & \text{2nd - algorithm} \end{cases}$$

The total number of I-P(M)-frames $TF^{Interactive}_{I,P(M)}$ in the interactive mode can be computed as follows

$$TF^{Interactive}_{I,P(M)} = P \times N_{ineractive} \qquad (12)$$

where,

- P is the number of P-frames (2).
- $(N_{\text{interactive}} = \text{var}\,iable, M_{\text{interactive}} = 1)$ are the new re-encoding parameters

According (10), (11), (12) we get

$$\frac{W_{server}}{W_{ExtraStorage}} = \frac{1}{\left(\frac{1}{M} - \frac{1}{N}\right) \times N_{\text{interactive}}} \times \frac{Aver(I,P,B)Size}{(P-IS)Size}$$

The additional storages required at the client-end by $P \to I$ conversion method [1] can be computed as follows

$$W_{P \to I} = TF \times Average(IB)Size \qquad (13)$$

where,

$$Average(IB)Size = \frac{I_{average}}{M} + B_{average}(1 - \frac{1}{M})$$

For our experiment we used the same "Motor Race " sequence with 180 frames, which was encoded at 2Mbps, with a frame rate of 30frps

For $TF = 180$, $N_{\text{interactive}} = 2, M_{\text{interactive}} = 1$ we get from (12)

$TF^{Interactive}_{I,P(M)} = 96$

The bit rate allocation during the interactive mode depends on the proposed encoding scheme. Figure 6 depicts the bit rate allocation for the proposed schemes.

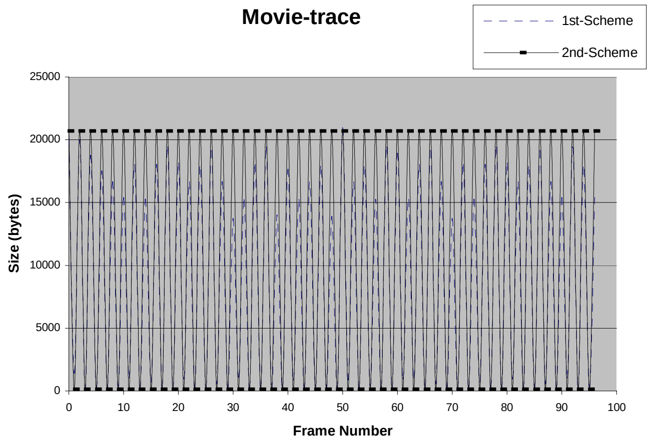

Figure 6: Frame sizes of interactive mode

1st Scheme

- $\{ w(u,v) = \text{fixed} \}$
- $I_{transcoded} \leq Threshold^{higher} = I_{average}$

The $Average\{IP(M)\}Size$ is given by

$$Average(IP)^{interactive} Size = \frac{I^{interactive}_{average}}{N_{interactive}} + P^{interactive}_{average} \text{ x } \left(\frac{1}{M_{interactive}} - \frac{1}{N_{interactive}}\right)$$

2nd Scheme

- $\{ w(u,v) = \text{variable} \}$
- $\frac{9}{10} I_{average} \leq I_{transcoded} \leq I_{average}$

The $Cons\tan t\{IP(M)\}Size$ is given by

$$Cons\tan t\{IP(M)\}Size = 10500 bytes / frame$$

From Table I, it is seen that the additional storage required by $P \to IS$ method is small comparing with the storage at the server (W_{server}) and the extra storage due to $P \to I$ conversion method ($W_{P \to I}$) [1]

Table I: Comparison the extra-storage

	1st-Scheme	2nd-Scheme
$W_{server} / W_{P \to IS}$	2.03	1.51
$W_{P \to I} / W_{P \to IS}$	2.6	1.91

5 Support Variable Speedups

It is useful to derive a closed-form formula to show the number of the supported speedups of the proposed method. The speedups can be computed as follows

$$SpUps = \frac{\frac{RR\_InteractiveMode}{N_{interactive}}}{RR\_NornalPLay \times (\frac{1}{M} - \frac{1}{N})} \qquad (14)$$

where, $RR = RecordingRatio$

To minimize the complexity at the decoder module the frame rate during the interactive mode is the same as the one during the normal playback. If the decoder consumes data at higher or lower rate than the one specified it would result in slight hiccups at the client end. This phenomenon will occur in any system where the server's production rate differs from the consumption rate of the decoder. Either the decoder will eventually starve or overrun its buffers. It is certainly conceivable that different end users have different hardware decoder cards or even software decoders, each with different consumption rates.. Hence from (14) we get

$$SpUps = \begin{cases} \dfrac{\frac{1}{N_{interactive}}}{(\frac{1}{M} - \frac{1}{N})}, FastReversePlay \\ S_p \times (\dfrac{\frac{1}{N_{interactive}}}{(\frac{1}{M} - \frac{1}{N})} - \dfrac{1}{\Omega}) + \dfrac{\frac{1}{N_{interactive}}}{(\frac{1}{M} - \frac{1}{N})}, JumpReversePlay \end{cases}$$

where,

$S_p < TFx(\frac{1}{M} - \frac{1}{N})$ is the number of skipped P-frames of the normal stream and $\Omega = RR\_NormalPLay \times (\frac{1}{M} - \frac{1}{N})$

Note that

- Fast Rewind is an operation in which the client browses the presentation in the backward direction with normal sequences of pictures. This function is supported by trasncoding all the P-frames (S_p =0) in the backward order of the normal stream.

- Jump Backward is an operation in which the client jumps to a target time of the presentation in the backward direction without normal sequence of pictures. Therefore the jump operations let the users jump directly to a particular video location. Jump. Backward operation is supported by skipping backward some P-frames (S_p) of the normal stream.

- Slow Reverse is an operation in which the video sequence is presented backward with lower playback rate. This functions can be supported by transcoding all the P-frame of a (GoPs) in the backward order and generate P(M) frames as many as P and B frames in a Recording Ratio (RR) of normal playback ($N_{interactive} = RR$)

- Reverse is an operation in which the presentation is played in the reverse direction with the same playback rate as during the normal play (Forward Playback). This functions can be supported by transcoding all the P-frame in the reverse order and generate P (M) frames as many as P and B frames in a Group of Pictures (GoP) of normal playback ($N_{interactive} = N$)

Figure 3 depicts the number of supported speedups as a function of $N_{interactive}$ for various numbers of skipped P-frames (S_p) of normal stream.

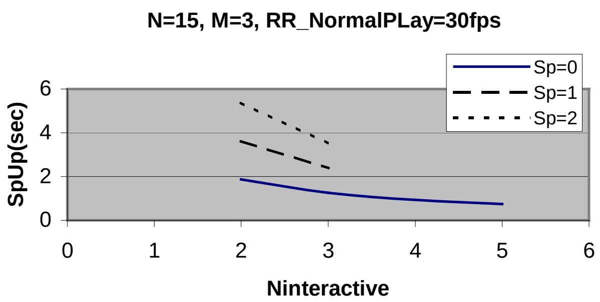

Figure 4: Supported Speedups

6 Analyse the Visual Quality

There are two types of criteria that can be used for the evaluation of video quality; subjective and objective. It is difficult to do subjective rating because it is not mathematically repeatable. For this reason we measure the visual quality of the interactive mode using the Peak Signal-to-Noise ratio (PSNR). The two approaches can be constrained with respect to video quality using the (PSNR). We use the PSNR of the Y-component of a decoded frame. The PSNR is obtained by comparing the original raw frame with its decoded version with encoding being done using the proposed schemes.

Figure 4 depicts the PSNR values for motor race movie. Both schemes achieve acceptable visual quality because the PSNR is sufficiently large. The average PSNR value for the 96- frames is 43.03 dB for the first scheme and 47.06 dB for the second scheme, i.e., the average quality is better under the second scheme

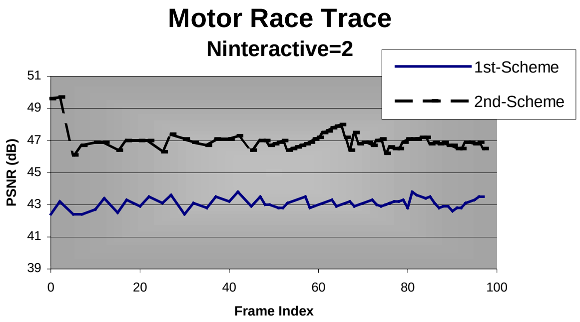

Figure 4: PSNR for encoded frames in the interactive mode

Clearly the second scheme achieves better visual quality than the first one, but at expense of more re-encoding attempt. It is worth mentioning that the drift during the Fast/Jump Rewind mode is relatively insensitive to human eye due to the fast change of the content displayed

8 Conclusions

In this paper, we presented an approach for supporting rewind operations such as Fast/Jump/Slow Rewind and Rewind in a Video On Demand System. Rewind functions are supported by transcoding the Predicted (P)-coded frames of the original MPEG-2 stream as Intra (I) -coded frames in order to generate the interactive bit-stream. During the transcoding process the interactive stream can be performed in a Group of Pictures (GoPs) structure in an independent fashion. The frame rate of the secondary stream can be artificially increased by including so called P (M) marionette frames. Effectively this results in repeating the previous frame in the decoder. Simulation results show that the visual quality of the secondary bit stream is in acceptable quality.. It can be seen that the buffer space required for rewind operations will be the amount for storing one decompressed frame, i.e., less than forward play. Moreover $P \rightarrow IS$ method requires significant smaller amount of secondary storage than $P \rightarrow I$ conversion scheme [1] One limitation of the proposed approach is that the viewer must see the movie first in the forward direction before switching to rewind operations. Future research will focus on applications of the proposed concept to MPEG-4 stream.

ACKNOLEGDMENT

The authors wish to thank all who reviewed this paper, especially Dr A Krikelis the Chief Technology Officer of Aspex Technology for his helpful comments.

References:

[1] M-S.Chen, D.D. Kandlur. "Downloading and Stream Conversion: Supporting Interactive Playout of Video in a Client Station". In: Proc IEEE Multimedia Conference (1995).

[2] S.J. Wee, B Vasudev "Compressed-domain reverse play of MPEG video streams ". In: Proc SPIE Conf. Multimedia Systems and Applications, Nov 1998, pp 237-248

[3] Michael Vernick, Chitra Venkatramani, Tzi-cher Chinueh. "Adventures in Building the Stony Brook Video Server". In: Proc. ACM Multimedia, November 1996 Boston, MA pp 287-295

[4] Marwan Krunz, George Apostolopoulos. "Efficient Support for interactive scanning operations in MPEG-based video on video on demand". Multimedia Systems, vol.8, no.1, Jan. 2000, pp.20-36

[5] Kostas Psannis Marios Hadjinicolaou " A New Methodology for Implementing Interactive Operations of MPEG-2 Compressed Video", In Proc IEEE Second International Workshop on Digital and Computational Video, Tampa, USA, Feb2001, pp 21-28

[6] Kostas Psannis, Marios Hadjinicolaou "Transmitting Additional Data of MPEG-2 Compressed Video to Support Interactive Operations". In: Proc. IEEE International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, May 2001, pp 308-311

[7] Kostas Psannis, Marios Hadjinicolaou "Implementing Interactive Operations of MPEG-2 Stream", In Proc: WSES/IEEE Multimedia, Internet, Video Technologies, Sept2001 Malta, pp 262-267

[8] International Standard 13818-2.Information Technology- Genetic Coding of Moving Pictures and Associated Audio: Video

[9] MPEG Software Simulation Group, "MPEG-2 Encoder/ Decoder Version 1.2" ftp://ftp.mpeg.org/pub/mpeg/mssg/