MOTION ADAPTIVE ERROR RESILIENT ENCODING FOR MPEG-4

S. T. Worrall, A. H. Sadka, P. Sweeney, A. M. Kondoz

Centre for Communications Systems Research (CCSR),
University of Surrey, Guildford,
Surrey GU2 7XH, UK
{s.worrall, a.sadka, p.sweeney, a.kondoz}@eim.surrey.ac.uk

ABSTRACT

Optimising delivery of video codecs such as MPEG-4 is vital to ensure that acceptable quality can be offered to 3G network customers. The error prone nature of mobile channels means that the video codec that is employed must be fairly robust. In the past, such research has required alteration to existing standards. Given the imminent implementation of 3G technologies, changing the standards is not a convenient option. This paper presents a technique for altering MPEG-4 encoding parameters to increase the error robustness of the output bitstream. It exploits the fact that frames with a high degree motion are often more sensitive to error than those with low amounts of motion. Using a simple model, the video packet length and number of AIR blocks are varied according to the amount of motion in a frame. Simulations using a GPRS channel are presented to confirm the benefits of the proposed scheme.

1. INTRODUCTION

The MPEG-4 standard [1] includes a number of error resilience features [2], and has been designed with mobile channels in mind [3]. However, tests have shown that the quality of the decoded MPEG-4 video after transmission over certain mobile channels can still be significantly degraded [4]. It is necessary to employ high channel coding rates, and have a reasonably high C/I ratio.

Many techniques for improving error resilience have been suggested [5]. However, implementing many of the proposed methods would require alteration to the MPEG-4 standard. Making such changes outside of the standardisation body would negate any benefits to be gained from exploiting a standardised codec. In particular, facilitation of interoperability between different users and different mobile networks would be much more complex. Thus, there is a need for techniques that either make no alteration to the standard, or produce bitstreams that retain backwards compatibility with standard MPEG-4 decoders. A backwards-compatible scheme is described in [6], and is employed in the simulations described in this paper. This is in addition to the technique proposed by this paper.

The technique proposed in this paper is entirely compatible with standard MPEG-4 decoders. It achieves this through altering only the parameters that are used to encode the video. There are two different parameters in MPEG-4 that affect the robustness of the video: video packet length, and frame refresh rate. The most basic error resilience option specified in the visual standard, allows frames to be split into a number of independently decodable video packets of a preset length. Increasing packet length reduces overhead, and thus increases error free quality. However, it also reduces the robustness of the bitstream to errors.

In this paper, refreshing of the video is accomplished through implementation of the Adaptive Intra Refresh (AIR) scheme outlined in Annex H of the MPEG-4 visual standard [1]. Varying the refresh rate of the video refers to altering the number of AIR macroblocks (MBs) that are encoded in each frame. Clearly, encoding large amounts of AIR MBs increases error resilience, but reduces error free quality.

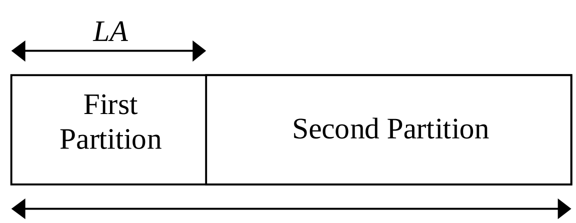

Fig. 1. MPEG-4 video packet.

This paper is split into four sections, including this introduction. The second section describes the proposed motion adaptation scheme. The third section contains a description of simulations, which demonstrate the effectiveness of the scheme over a General Packet Radio Service (GPRS) channel. It also features a brief analysis of the results that were obtained. Finally, a summary of the work is provided in the fourth section.

2. MOTION ADAPTATION

The technique is based upon an estimation of the amount of motion within a video scene. Estimation is achieved through observation of the amount of encoded motion data within a frame. There is a strong correlation between activity of a video scene and the size of the motion information required to encode it.

The data partitioning error resilience mode of MPEG-4 separates motion and header data from less sensitive texture data. Figure 1 shows a simple illustration of an MPEG-4 video packet. Motion and header data is placed into the first partition, while texture is positioned in the second partition. For the purposes of this paper, L is the length of a video packet, and A is the proportion of the video packet occupied by the first partition. As the size of motion data is related to the amount of motion within a scene, motion can be estimated from A.

Figure 2 shows how first partition size varies throughout the standard QCIF (176×144 pixels) sequence "Suzie". The sequence was encoded at 10 frames per second, and with a bit rate of 64 kbit/s. Video packet length was set to 700 bits, while 8 AIR macroblocks were encoded in each frame.

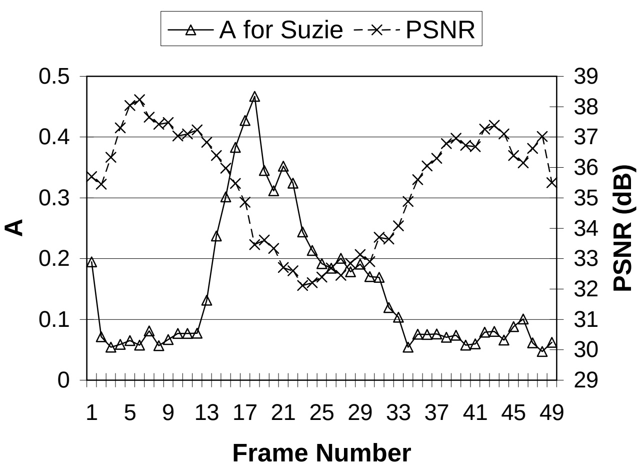

Fig. 2. First partition size for Suzie compared to the PSNR of Suzie after transmission over a GPRS channel (C/I = 12 dB, half rate convolutional coding).

In the middle of the video sequence, the subject shakes her head, which is clearly a high motion event. Either side of this there is very little motion at all. Figure 2 shows that the size of A increases significantly as the motion increases. It also demonstrates that the quality drops off significantly at the same time. Part of this fall in PSNR is due to the nature of fixed bandwidth video coding. High motion sections require more bits to encode, and therefore the quality has to be decreased to maintain a fixed throughput. However, most of the perceivable errors that occur due to channel errors also appear during these sections. This results in quality being pushed even lower.

In the proposed technique, video packet size is decreased linearly as A increases. For the simulations described later, the following scheme was used to set packet size:

$$L = 1200 - 2000A \text{ for } 0.1 < A < 0.4$$
$$L = 400 \text{ for } A > 0.4$$
$$L = 1000 \text{ for } A < 0.1$$

Where L is the length of the packet in bits. An integer number of bits is obtained by rounding L to the nearest bit. Limits are set to prevent the packet size from either becoming so small as to have a significantly negative impact upon compression efficiency, or from becoming so large that sensitivity to errors becomes too great.

Thus, as the sensitivity of the video increases (due to an increase in motion), video packet size is reduced to improve error robustness. The values shown above were arrived at experimentally; there is as yet no analytical scheme for selecting optimum values for each video sequence.

A	N_{AIR}
< 0.1	4
$0.1 - 0.15$	8
$0.15 - 0.2$	10
$0.2 - 0.28$	14
$0.28 - 0.35$	12
$0.35 - 0.4$	12
> 0.4	4

Table 1. Number of AIR MBs to be encoded in each frame for a particular range of A, as A is decreasing.

The amount of AIR macroblocks encoded within a frame is set using a slightly more complex scheme. It was observed by the authors that during periods of high scene activity, large numbers of AIR MBs caused a large drop in encoded quality, irrespective of channel conditions. This is mainly due to the coding inefficiency of Intra macroblocks.

To optimise quality, it was found from experimentation by the authors that it is necessary to increase the refresh rate after periods of high motion. This is accomplished by observing the change in A from frame to frame. Two different schemes are used for selecting the number of AIR macroblocks. If A is increasing then the following settings for N_{AIR}, the number of AIR MBs encoded within a frame, are used:

$$N_{AIR} = 8 \text{ for } 0.1 \leq A \leq 0.4$$
$$= 4 \text{ for } A < 0.1 \text{ and } A > 0.4$$

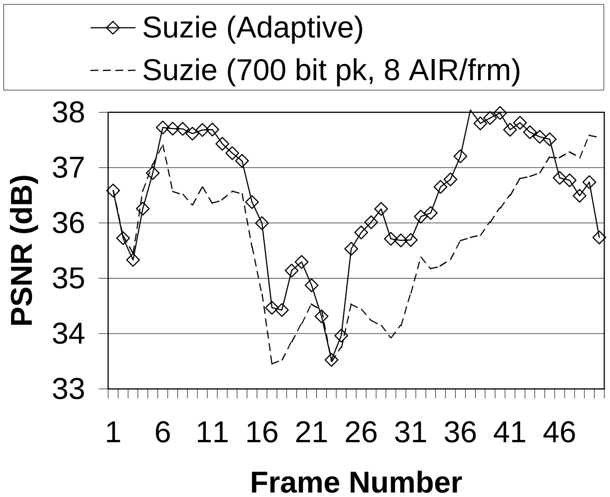

Fig. 3. PSNR comparison of Suzie using motion adaptive encoding, and a fixed parameter encoding case.

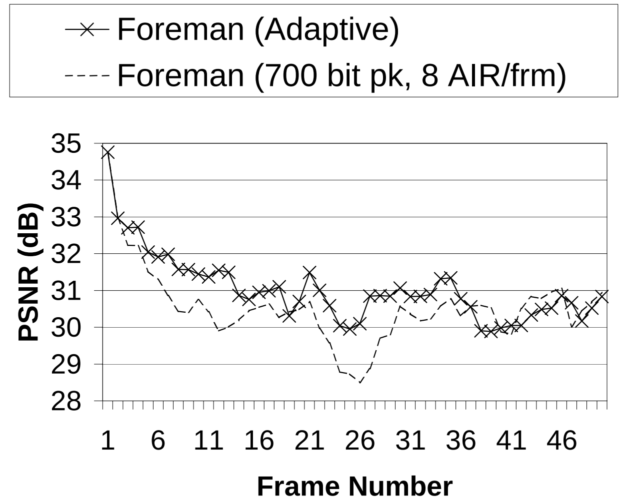

Fig. 4. PSNR comparison of Foreman using motion adaptive encoding, and a fixed parameter encoding case.

Sequence	Av. Fixed PSNR (dB)	Av. Adaptive PSNR (dB)
Suzie	36.02	36.48
Foreman	30.52	31.02

Table 2. Comparison of average PSNR values for fixed and motion adaptive encoding.

(a) Fixed parameter. (b) Motion adaptive.

Fig. 5. Frame 28 of suzie after simulation over a GPRS channel (C/I = 12 dB, half rate convolutional code).

The settings for N_{AIR} for when A is decreasing can be found in Table 1. Thus, the scheme facilitates fast updating of a video scene following periods of high motion. It also reduces the number of AIR macroblocks during periods of high motion, when large numbers of AIR MBs would cause quality to drop. Simulation results showing the performance of the proposed scheme are described in the next section.

3. SIMULATIONS

Corruption of the video data was performed using error patterns derived from models for a GPRS channel [4]. The simulated radio conditions were those for an interference-limited scenario, which is the most common operating scenario for mobile terminals. The propagation conditions were those specified in GSM 05.05 as TU50 Ideal Frequency Hopping at 900MHz. The TU50 channel model represents multipath propagation conditions found in typical urban conditions, with a mobile terminal velocity of 50 km/hr. The model includes the use of a half-rate convolutional code. The simulation also assumed transmission using the Real-time Transport Protocol (RTP) over UDP on top of IP [7]. A single frame was encapsulated within a single RTP packet. Corruption of any part of the RTP/UDP/IP header resulted in loss of a whole frame.

Twenty four simulations were run for each condition shown in the results. The simulation results were then averaged to produce PSNR values for each frame, along with average PSNR values for the whole sequence.

The decoder employs simple error concealment upon encountering errors, which involves replacing incorrect or missing macroblocks with their corresponding motion compensated macroblocks. This is accomplished in the decoder implementation by setting the prediction errors for the motion vectors and pixel values to zero for each affected macroblock. This is a standard technique employed in many decoder implementations. Error detection is aided by the insertion of CRCs into the bitstream [6].

Tests were performed to obtain results with fixed parameters, and with the motion adaptive scheme. All tests were performed using a C/I ratio of 12 dB, which gives a Bit Error Ratio (BER) in the region of 3×10^{-4}. Two standard QCIF sequences were employed in the simulations, Foreman and Suzie. Both sequences were encoded at 10 frames per second, with a bit rate of 64 kbit/s. For both sequences, 150 frames were encoded from the original sequences, which meant that 50 frames were produced at the output. The encoding parameters for the fixed case were set to: 700 bit video packets, and 8 AIR blocks per frame. These parameters were found to produce approximately optimum results for a C/I of 12 dB.

Results for Suzie are displayed in Figure 3, while Foreman results are shown in Figure 4. The graphs demonstrate a significant improvement in quality for the adaptive technique when compared to fixed parameter encoding. Note that for Suzie, the PSNR recovers rapidly after the high motion section due to the increase in AIR macroblocks after the motion peak. An example frame of Suzie is shown in Figure 5. Both bitstreams had the same error pattern applied to each. However, the frame from the motion adaptive version demonstrates that the proposed technique facilitates speedier recovery from error than with regular MPEG-4 encoding.

4. CONCLUSION

A proposal for improving the robustness of MPEG-4 has been described in this paper. The technique exploits a correlation between the amount of motion within a scene and the amount of errors. It varies video packet size and the number of AIR macroblocks within a frame as the motion within the video scene changes.

Simulations demonstrated the advantages of the scheme over a GPRS channel. The technique described produces bitstreams that are compatible with standard MPEG-4 decoders.

5. REFERENCES

[1] ISO/IEC JTC 1/SC 29/WG 11, "Information technology - generic coding of audio-visual objects - part 2: Visual," *ISO/IEC 14496-2*, March 2000.

[2] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine*, pp. 112–119, June 1998.

[3] A. Puri and A. Eleftheriadis, "MPEG-4: An object-based multimedia coding standard supporting mobile applications," *ACM Mobile Networks and Applications*, vol. 3, pp. 5–31, June 1998.

[4] S. Fabri, A. Cellatoglu, and A. Kondoz, "Transmission of multimedia services over GPRS using MPEG-4 coded video," in *IEEE VTC-Fall*, 1999, pp. 401–405.

[5] Y. Wang and Q-F. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[6] S. T. Worrall, A. H. Sadka, P. Sweeney, and A. M. Kondoz, "Backward compatible user-defined data insertion into MPEG-4 bitstreams," *IEE Electronics Letters*, vol. 36, no. 12, pp. 1036, June 2000.

[7] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," *Audio-Video Transport Working Group, RFC 1889*, Jan. 1996.