Optimal packetisation of MPEG-4 using RTP over mobile networks

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

Abstract: The introduction of third-generation wireless networks should result in real-time mobile video communications becoming a reality. Delivery of such video is likely to be facilitated by the real-time transport protocol (RTP). Careful packetisation of the video data is necessary to ensure the optimal trade-off between channel utilisation and error robustness. Theoretical analyses for two basic schemes of MPEG-4 data encapsulation within RTP packets are presented. Simulations over a GPRS (general packet radio service) network are used to validate the analysis of the most efficient scheme. Finally, a motion adaptive system for deriving MPEG-4 video packet sizes is presented. Further simulations demonstrate the benefits of the adaptive system.

1 Introduction

The imminent arrival of UMTS (universal mobile telecommunications service) has increased the need for efficient multimedia delivery mechanisms. One of the major issues with multimedia transmission over such networks appears to be the error resilience of video data, which has already led to some significant research [1]. Some of this research has been incorporated into codecs such as H.263++ and MPEG-4 [2]. In particular, MPEG-4 has been developed with wireless channels in mind [3]. Three error resilience tools are provided by the standard: regular insertion of resynchronisation words, data partitioning and reversible variable length codes (RVLC) [4].

Interoperability between mobile terminals and any other type of terminal is probably best achieved with IP (internet protocol). Thus, for real-time communications, it is likely that the real-time transport protocol (RTP) [5] will be exploited, which usually also entails the use of UDP (user datagram protocol). The way in which RTP is used for MPEG-4 has ramifications for channel utilisation and error resilience. Transmission of MPEG-4 over RTP has been considered [6], but without focusing on the error resilience issue for mobile networks.

A single MPEG-4 stream is usually broken up into a series of independently decodable video packets of regular lengths, with a resynchronisation word separating each packet. These video packets are created by the MPEG-4 encoder, and can be considered as part of the compression layer. It is important to distinguish between RTP packets, which are created separately from the MPEG-4 encoding process, and video packets generated by the MPEG-4 encoder. Setting the length of the video packets involves a trade-off between error resilience and an increase in overhead. This is the kind of situation that has led to research into a variety of techniques capable of adapting to video scene content and channel conditions.

Research into adaptive techniques can be grouped into a number of areas. Adaptation of the source-channel coding ratio to channel conditions has shown significant benefits [7, 8]. Another option is to adapt the manner in which a video frame is refreshed, either by looking at channel characteristics [9, 10], or using a hybrid ARQ (automatic repeat request) scheme [11]. Alternatively, the spacing of resynchronisation markers can be adjusted [12].

2 Analysis of RTP packetisation

Corruption of any part of an IP/UDP/RTP header results in the loss of a whole RTP packet. The situation is further complicated by the differing sensitivities of data within an MPEG-4 video packet. Data partitioned MPEG-4 packets are split into two partitions. The first partition contains header and motion data, while the second consists of texture data. Without the first partition, the second partition cannot be decoded. Thus, corruption of the first partition results in the loss of a whole video packet. Any analysis of RTP packetisation must take these factors into account.

Two RTP packetisation schemes are examined here. 'Scheme 1' requires encapsulation of a single video packet within a single RTP packet. In 'scheme 2', an RTP packet consists of a single video frame, each video frame containing a number of video packets. In both cases, decoding is assisted by the insertion of CRC (cyclic redundancy check) codes as described in [13]. Although RVLC use is not possible with CRC insertion, it facilitates more consistent video quality by aiding error concealment. It should be noted that while a 16-bit CRC code is used in [13], the schemes employed here utilise an 8-bit code, which is taken from the ATM HEC (asynchronous transfer mode header extension code) specification.

For both packetisation schemes, a number of rules are set for the behaviour of the decoder on encountering errors. Corruption of an IP/UDP/RTP header causes the whole RTP packet to be dropped. Errors detected in the first partition cause a whole video packet to be discarded. If an error is detected during decoding of the second partition,

IEE Proceedings online no. 20010398

DOI: 10.1049/ip-com:20010398

Paper first received 7th August 2000 and in final revised form 3rd April 2001

The authors are with the Centre for Communication Systems Research (CCSR), University of Surrey, Guildford, Surrey GU2 7XH, UK

the rest of the video packet is dropped. However, if no error is detected during decoding of the second partition, but the second partition CRC indicates the presence of errors, then the whole second partition is dropped.

2.1 Analysis of scheme 1

Fig. 1 shows packetisation scheme 1. L is the length of the MPEG-4 video packet, Y is the length of the first partition, and X is the second partition size. From this, average first partition length is

$$\overline{Y} = \frac{L\overline{Y_{MB}}}{\overline{X_{MB}} + \overline{Y_{MB}}} = LA \tag{1}$$

where $\overline{Y_{MB}}$ is the average number of bits per macroblock in the first partition, and $\overline{X_{MB}}$ is the average number of bits per macroblock in the second partition.

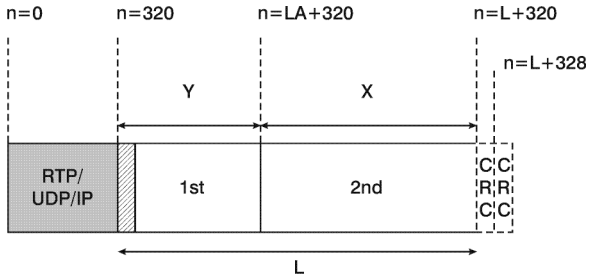

Fig. 1 *RTP packetisation scheme 1*

Therefore, A corresponds to the proportion of a video packet that is occupied by the first partition. If C is the proportion of a video packet assigned to the second partition, then average second partition length is

$$\overline{X} = \frac{L\overline{X_{MB}}}{\overline{X_{MB}} + \overline{Y_{MB}}} = LC \tag{2}$$

These definitions can be used to derive the effective error rate of each section of the packet when subjected to a channel error rate e_{ch}. Taking a simple example, the effective error rate of the packet displayed in Fig. 2, e_{eff}, is

$$e_{eff} = \sum_{m=0}^{L-1} p_m \frac{L-m}{L} \tag{3}$$

where p_m is the probability of m being the first corrupted bit, such that

$$p_m = (1-e_{ch})^m e_{ch} \tag{4}$$

Thus, eqn. 3 can be written as

$$e_{eff} = \frac{e_{ch}}{L} \sum_{m=0}^{L-1} (1-e_{ch})^m (L-m) \tag{5}$$

Eqn. 5 can be used as an example for the derivation of effective error rates for each part of the packetisation scheme demonstrated in Fig. 1. In each case the channel error rate is multiplied by the proportion of the packet that each part occupies. This ensures that the effective error rates receive weightings proportional to the likelihood that the corresponding section of the bitstream will be corrupted.

Thus, the effective error rate for the RTP/UDP/IP header is

$$e_{eff1} = \sum_{n=0}^{319} (1-e_{ch})^n e_{ch} \tag{6}$$

The result of the summation can be expressed as

$$e_{eff1} = 1 - (1-e_{ch})^{320} \tag{7}$$

The effective error rate for the first partition can be expressed as

$$e_{eff2} = (1-e_{ch})^{320}\left(1-(1-e_{ch})^{LA}\right) \tag{8}$$

For the first partition CRC, the effective error rate can be shown to be

$$e_{eff4} = (1-e_{ch})^{L+320}\left(1-(1-e_{ch})^{8}\right) \tag{9}$$

If X is equal to $\bar{X}$ as defined in eqn. 2, the second partition CRC effective error rate is

$$e_{eff5} = \frac{LC+8}{L+336}(1-e_{ch})^{L+328}\left(1-(1-e_{ch})^{8}\right) \tag{10}$$

The derivation of the effective error rate for the second partition is more complex. However, it can be shown to be

$$e_{eff3} = \frac{a^{320+LA}}{e_{ch}(L+336)}\left(a^{X+1} + e_{ch}(X+1) - 1\right) \tag{11}$$

where a is defined as

$$a = 1 - e_{ch} \tag{12}$$

Finally, the overall effective error rate is

$$e_{eff} = e_{eff1} + e_{eff2} + e_{eff3} + e_{eff4} + e_{eff5} \tag{13}$$

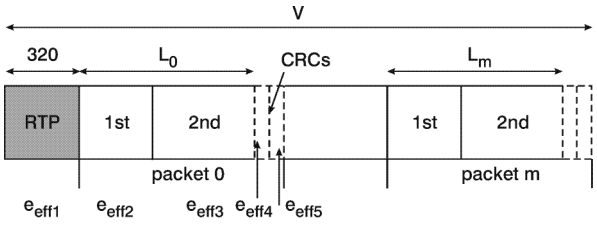

Fig. 3 *RTP packetisation scheme 2*

2.2 Analysis of scheme 2

The second packetisation scheme is demonstrated in Fig. 3. Note that, like scheme 1, there are still resynchronisation codewords between each video packet. For this scheme it shall be assumed that all packet lengths except the last one are equal:

$$L_0 = L_1 = \ldots = L_{m-1} \equiv L_P \tag{14}$$

L_P is the video packet length. The final video packet contains any data left in the VOP (visual object plane), and can be shown to be of length:

$$L_m = V - 336 - (L_P + 16)\left\lfloor \frac{V-320}{L_P+16} \right\rfloor \tag{15}$$

where V is the VOP (frame) length.

As before, the effective error rates for each section of the RTP packet are derived individually. First, the RTP/UDP/IP header effective error rate is

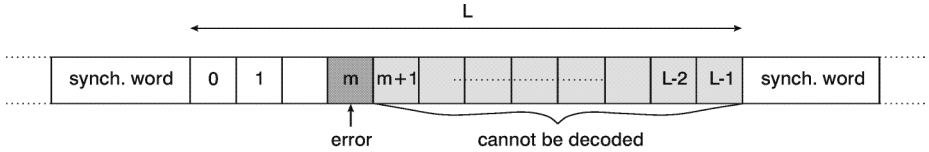

Fig. 2 *One-way decoding of variable length codes*

$$e_{eff1} = 1 - (1 - e_{ch})^{320} \quad (16)$$

The first partition effective error rate, with L as the packet length, is

$$e_{eff2} = \frac{L}{V}(1 - e_{ch})^{320}\left(1 - (1 - e_{ch})^{LA}\right) \quad (17)$$

For the second partition:

$$e_{eff3} = \frac{(1-e_{ch})^{320+LA}}{e_{ch}V}\left((1-e_{ch})^{X+1} + e_{ch}(X+1) - 1\right) \quad (18)$$

For the first partition CRC:

$$e_{eff4} = \frac{L}{V}(1 - e_{ch})^{L+320}\left(1 - (1 - e_{ch})^{8}\right) \quad (19)$$

For the second partition CRC:

$$e_{eff5} = \frac{X}{V}(1 - e_{ch})^{L+328}\left(1 - (1 - e_{ch})^{8}\right) \quad (20)$$

The effective error rate for a single video packet of length L is

$$e_{pk}(L) = e_{eff2}(L) + e_{eff3}(L) + e_{eff4}(L) + e_{eff5}(L) \quad (21)$$

From eqns. 15 and 21, the overall effective error rate for a single RTP packet is

$$e_{eff} = e_{eff1} + \left\lfloor \frac{V - 320}{L_P + 16} \right\rfloor e_{pk}(L = L_P) + e_{pk}(L = L_m) \quad (22)$$

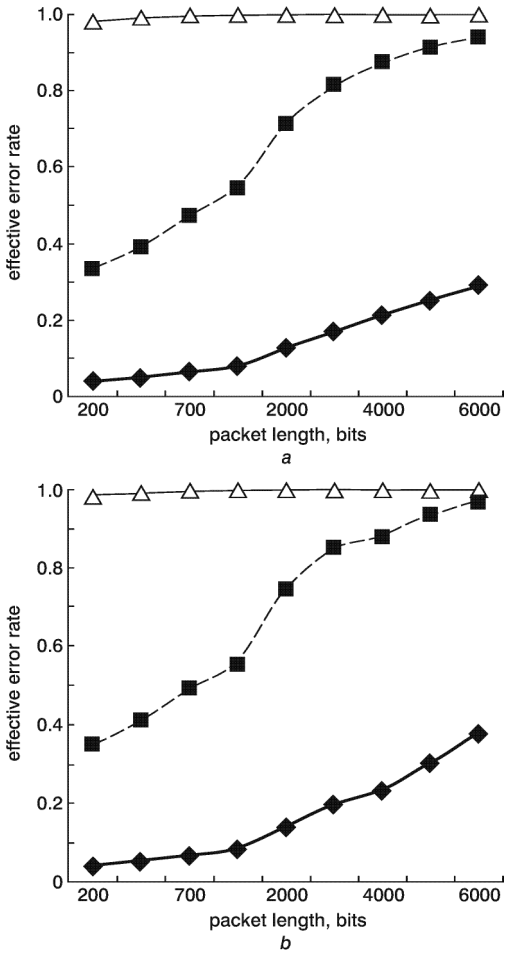

Fig. 4 *Effective error rates for different packet lengths with A = 0.3*
– ■ – BER = 10^{-3}
—◆— BER = 10^{-4}
—△— BER = 10^{-2}
a Scheme 1
b Scheme 2

2.3 Comparison of schemes

Fig. 4 shows graphs plotted using the effective error rate derivations outlined above. The results were obtained with VOP length V set to 6400 bits, and A equal to 0.3. This VOP length corresponds to video coded at 10 frames per second and 64 kbit/s, settings that could conceivably be employed for delivering video over UMTS. The value of A is typical for MPEG-4 bitstreams encoded with these settings. It is clear from Fig. 4 that with these settings, scheme 2 should be as robust to errors as scheme 1. However, as scheme 2 requires less overhead in terms of RTP headers, it can be considered to be the preferred scheme.

Although Fig. 4*b* indicates that the smallest possible packet sizes are preferable in terms of error resilience, the decoded quality will also be influenced by compression distortion. Video packets are generated by the MPEG-4 encoder, and are taken into account during compression. Thus, the extra overhead required by using an increased number of packets, reduces the compression efficiency, which increases compression distortion.

3 Simulations

Simulations have been performed to examine the relationship between the equations developed above and actual decoded quality after transmission over a noisy channel. Only scheme 2 was tested, given the poor performance of scheme 1.

Two standard ITU (International Telecommunication Union) sequences were used for testing, 'Suzie' and 'Foreman'. Both sequences are in quarter common interchange format (QCIF) with frame dimensions of 176 × 144. The sequences were compressed to 64 kbit/s at 10 frames per second. Eight adaptive intra refresh (AIR) macroblocks were encoded in each frame, the macroblock selection algorithm is as defined in an Annex of the MPEG-4 standard [2].

Tests were carried out using error patterns derived from a GPRS channel model. The simulated radio conditions were those for an interference-limited scenario, which is the most common operating scenario for mobile terminals. Propagation conditions were those specified in GSM 05.05 as TU50 ideal frequency hopping at 900 MHz. The TU50 model represents typical urban conditions with a terminal velocity of 50 km/h. A half rate convolutional code is included in the model, along with soft decision Viterbi decoding.

Simulations were performed with C/I ratios of 12 dB and 9 dB. Corruption with the 12 dB error pattern usually results in a BER (bit error rate) of around 10^{-4} while corruption with a C/I of 9 dB leads to BERs of approximately 3×10^{-3}. Please note that these are the BERs presented to the MPEG-4 decoder, after convolutional decoding. Each point of the graphs shown in Fig. 5 is the result of 24 GPRS channel simulations.

The simulation results are shown in Fig. 5. The trends of the graphs broadly follow the trends of the effective error rates shown in Fig. 4, in that the general trend is for the PSNR (peak signal-to-noise ratio) to fall as the effective error rate increases. Where the graphs appear to differ is that the PSNR in Fig. 5 does not drop so sharply as packet length increases past 1000 bits. For a C/I of 12 dB, this can be explained by the fact that the extra increase in effective error rate is small, and can easily be absorbed by the error-concealment algorithm. The error rates involved, with a C/I of 9 dB, fall in between 10^{-4} and 10^{-3}. Taking this into account the trends shown in Fig. 5 match the effective error rates more closely. The only exception is for

'Suzie', where, at a packet length of 200 bits, the distortion caused by extra overhead is greater than that caused by the greater effective error rate when 400 bit packets are used.

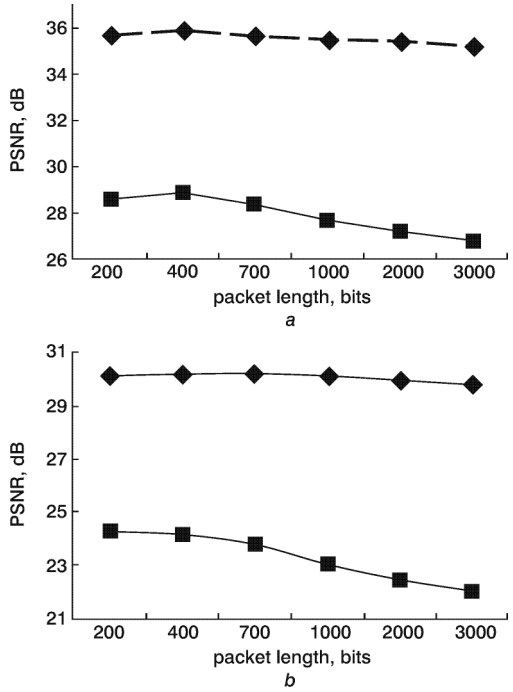

Fig. 5 *PSNR for 'Suzie' and 'Foreman' sequences after simulated transmission over GPRS*
◆ CS1, 12 dB
■ CS1, 9 dB
a Suzie
b Foreman

From the results it appears that the optimum packet sizes for the two sequences are different. The most obvious difference between the two sequences is the amount of motion. The proportion of each packet occupied by the first partition is related to the amount of motion in a scene. The variable A usually increases as the amount of motion increases. Therefore, the results suggest some relationship between the size of A and optimum video packet size.

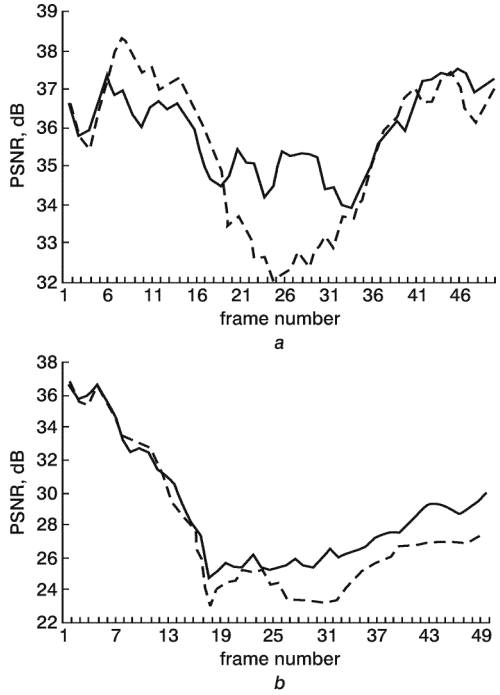

Fig. 6 *Frame-by-frame PSNR analysis of 'Suzie' with different packet lengths and C/I ratios*
——— 400 bit packets
– – – – 1000 bit packets
a CS1, *C/I* = 12 dB
b CS1, *C/I* = 9 dB

Frame-by-frame PSNR analysis of 'Suzie' is displayed in Fig. 6, while A is plotted against frame number in Fig. 7. Fig. 6*a* demonstrates that when A is small at the beginning of the sequence, a larger packet size produces better results. However, when A increases, a smaller packet size becomes more appropriate. Although both packet sizes produce similar results at the end of the sequence, it should be noted that the 1000 bit packets case starts from a lower PSNR after A drops off. The PSNR for 1000 bit packets has climbed faster when A has fallen back to a low level. These results indicate that a scheme that varies packet size with first partition size may provide quality improvements for sequences featuring sudden bursts of motion. The next Section describes how such a scheme was implemented, and demonstrates its benefits.

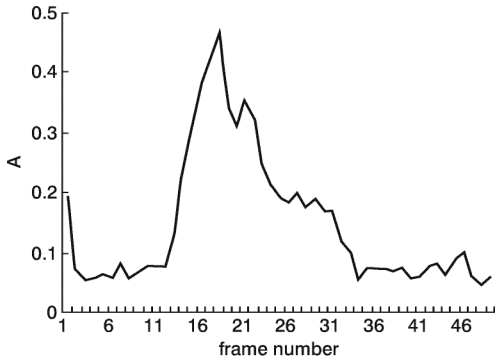

Fig. 7 *Proportion of packet occupied by the first partition, A, against frame number for 'Suzie'*

Table 1: Average PSNR of 'Suzie' after transmission over a GPRS channel with a C/I of 12 dB

Packetisation, bits	200	400	700	1000	Adaptive
Av. PSNR, dB	35.65	35.90	35.64	35.45	36.41

4 Motion adaptive packetisation

Implementation of the motion adaptive scheme was achieved through alteration of the encoder, facilitating linear variation of video packet length with respect to A. From the results shown in Table 1, it was decided that for a C/I of 12 dB, packet lengths would be decided from the following:

$$L = 1200 - 2000A \text{ for } 0.1 < A < 0.4$$
$$L = 400 \text{ for } A > 0.4$$
$$L = 1000 \text{ for } A < 0.1 \quad (23)$$

For C/I ratios of 9 dB it was observed that the following should be used:

$$L = 1267 - 2667A \text{ for } 0.1 < A < 0.4$$
$$L = 200 \text{ for } A > 0.4$$
$$L = 1000 \text{ for } A < 0.1 \quad (24)$$

Simulations for the adaptive method were performed as described in Section 3. Fig. 8 demonstrates that the proposed algorithm approximately identifies the optimum packet length throughout the sequence, resulting in a PSNR that is close to or greater than the optimum. Table 1 demonstrates that average PSNR with a C/I of 12 dB is higher using the adaptive technique. In Table 2 the adaptive scheme is shown to have similar performance to the optimum fixed length case.

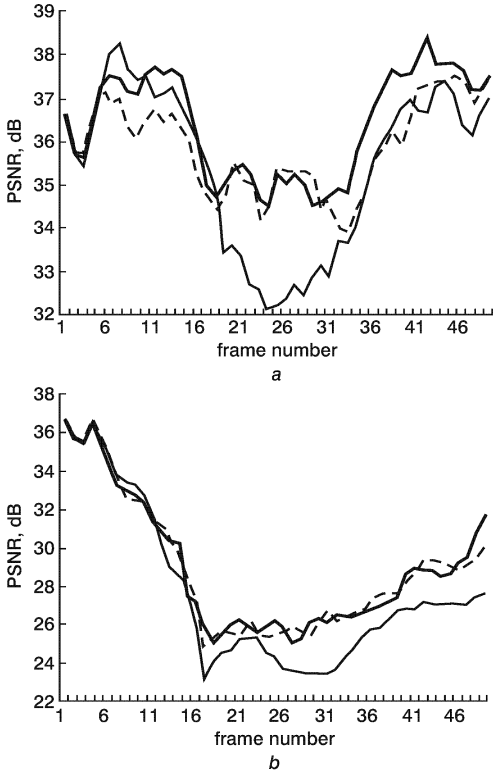

Fig. 8 *Frame-by-frame analysis of 'Suzie', adaptive packetisation against fixed packetisation*
adaptive
1000 bit packets
400 bit packets
a C/I = 12 dB
b C/I = 9 dB

Table 2: Average PSNR of 'Suzie' after transmission over GPRS with a C/I of 9 dB

Packetisation, bits	200	400	700	1000	Adaptive
Av. PSNR, dB	28.63	28.90	28.39	27.68	28.92

5 Conclusions

In this paper, mathematical analyses of two RTP packetisation schemes for low bit-rate MPEG-4 have been presented. For both schemes the theoretical effective error rates were derived, which clearly identified a preferred scheme. This preferred scheme requires encapsulation of a single video frame within a single RTP packet.

Simulations of transmission over a GPRS channel demonstrated a correlation between the theoretical effective error rates and average PSNR. Differences between the simulations and the mathematical analysis were apportioned to the error concealment mechanism, and extra compression distortion that is introduced with smaller packet sizes.

A potential link between optimum video packet length and the proportion of the packet occupied by the first partition was examined. This led to the formulation of an adaptive scheme that varied MPEG-4 video packet size with first partition size. Simulation results established the benefits of the technique in terms of PSNR gains.

6 Acknowledgments

This work has been supported by British Telecom. In particular the authors would like to thank Richard Jacobs and Ben Strulo from BT for their invaluable advice.

7 References

1 WANG, Y., and ZHU, Q.-F.: 'Error control and concealment for video communication: a review', *Proc. IEEE*, May 1998, 86, (5), pp. 974-997
2 ISO/IEC JTC 1/SC 29/WG 11, Doc. N28022. 'Information technology - Generic coding of audio-visual objects - Part 2: Visual, ISO/IEC 14496-2'. Proceedings of MPEG Vancouver meeting, July 1999
3 PURI, A., and ELEFTHERIADIS, A.: 'MPEG-4: An object-based multimedia coding standard supporting mobile applications', *Mobile Netw. Appl.*, 1998, 3, (1), pp. 5–31
4 TALLURI, R.: 'Error-resilient video coding in the ISO MPEG-4 standard', *IEEE Commun. Mag.*, 1998, 36, (5), pp. 112–119
5 SCHULZRINNE, H., CASNER, S., FREDERICK, R., JACOBSON, V.: 'RTP: a transport protocol for real-time applications'. Audio-Video Transport Working Group, RFC 1889, January 1996
6 BASSO, A., VARAKLIOTIS, S., and CASTAGNO, R.: 'Transport of MPEG-4 over IP/RTP'. Proceedings of Packet Video Workshop, Sardinia, Italy, May 2000
7 SALAMATIAN, M.R.: 'Joint source-channel coding applied to multimedia transmission over lossy packet network'. Proceedings of 9th international workshop on *Packet Video*, New York City, USA, April 1999
8 LU, J., NOSRATINIA, A., and AAZHANG, B.: 'Progressive source-channel coding of images over bursty error channels'. Proceedings of IEEE international conference on *Image Processing*, ICIP 1998, Kobe, Japan, pp. 127–131
9 CÔTÉ, G., and KOSSENTINI, F.: 'Optimal intra coding of blocks for robust video communication over the internet', *Signal Process., Image Commun.*, 1999, 15, (1–2), pp. 25–34
10 LE LÉANNEC, F., TOUTAIN, F., and GUILLEMOT, C.: 'Packet loss resilient MPEG-4 compliant video coding for the Internet', *Signal Process., Image Commun.*, 1999, 15, (1-2), pp. 35-56
11 LIU, H., and EL ZARKI, M.: 'Adaptive source rate control for real-time wireless video transmission', *Mobile Netw. Appl.*, 1998, 3, (1), pp. 49–60
12 YOO, K.-Y.: 'Adaptive resynchronisation marker positioning method for error resilient video transmission', *Electron. Lett*, 1998, 34, (22), pp. 2084–2085
13 WORRALL, S.T., SADKA, A.H., KONDOZ, A.M., and SWEENEY, P.: 'Backward compatible insertion of user-defined data into MPEG-4', *Electron. Lett.*, 2000, 36, (12), pp. 1036–1037