



Error resilient H.264/AVC Video over Satellite for low Packet Loss Rates

Aghito, Shankar Manuel; Forchhammer, Søren; Andersen, Jakob Dahl

Published in:

Proceedings of IEEE Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2007

Link to article, DOI:

[10.1109/WIAMIS.2007.42](https://doi.org/10.1109/WIAMIS.2007.42)

Publication date:

2007

Document Version

Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):

Aghito, S. M., Forchhammer, S., & Andersen, J. D. (2007). Error resilient H.264/AVC Video over Satellite for low Packet Loss Rates. In *Proceedings of IEEE Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS) 2007* IEEE. <https://doi.org/10.1109/WIAMIS.2007.42>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Error resilient H.264/AVC video over satellite for low packet loss rates

Shankar Manuel Aghito, Søren Forchhammer and Jakob D. Andersen
COM•DTU Department of Communications, Optics and Materials
Ørsteds Plads 343, 2800 Kgs. Lyngby, Denmark
sma@com.dtu.dk sf@com.dtu.dk jda@com.dtu.dk

Abstract

The performance of video over satellite is simulated. The error resilience tools of intra macroblock refresh and slicing are optimized for live broadcast video over satellite. The improved performance using feedback, using a cross-layer approach, over the satellite link is also simulated. The new Inmarsat BGAN system at 256 kbit/s is used as test case. This system operates at low loss rates guaranteeing a packet loss rate of not more than 10^{-3} . For high-end applications as 'reporter-in-the-field' live broadcast, it is crucial to obtain high quality without increasing delay.

1 Introduction

Using satellite communication, video may be shot at the most remote areas and used e.g. for live broadcast. Considering such a 'reporter-in-the-field' application, the satellite communication will constitute the bottle-neck. Recently, Inmarsat has introduced the Broadband Global Area Network (BGAN) [2], which offers 256 kbit/s satellite communication using portable terminals. For the high-end live broadcast application, the video may be visualized on high-definition flat panel displays for nation-wide scrutiny of the effects of transmission errors on video signal. We have simulated the coding parts of the system, i.e. the basic Turbo coding used for forward error-correction (FEC) which uses FEC packets of 5, 10 and 20 ms at different signal-to-noise ratios. The resulting packet-loss rates are used to analyse the packet-loss effect on H.264 [4] video using the H.264 reference software. In this work we focus on the video coding side. Feedback over the satellite link based on geostationary satellites is possible, but with an inevitable delay. Using this delayed feed-back is also simulated. The error resilience tools of the H.264 reference software considered are intra macro block refresh and slicing, which are suitable for low delay transmission using portable devices, since they do not impose requirements on the complexity of the encoder and do not introduce additional delay. A recent review of error resilience tools for H.264 is given in [1].

This work was supported in part by the Danish Agency for Science Technology and Innovation.

2 Simulation Set-up

In H.264 encoded video, pictures are partitioned in one or more slices. Each coded slice is embedded into one Network Adaption Layer Units (NALU). We assume that each NALU is inserted into one RTP packet. The conventional RTP/UDP/IP protocol stack is utilized together with robust header compression (ROHC), which typically compresses header data into 3 bytes [6]. The whole set-up is simulated simply using the H.264 reference software with Annex B output format (4 bytes start code in each slice). The IP stream is fed to the BGAN, which divides the data in FEC packets (not synchronized with the NALUs). The FEC packets are divided into packet data units (PDU). Damaged FEC packets are detected by a CRC-check on the PDUs. Initial simulations indicated that discarding a whole FEC packet will be the dominating case even if this hold multiple PDUs. (The probability of some PDUs being correct when the FEC packet is damaged is small.) Thus the model chosen is to discard all NALUs which are hit by a discarded FEC packet. We consider an average (FEC) packet loss rate (PLR) of 10^{-3} , whereas most of the work available in the literature has focussed on considerably higher PLR according to the common test conditions defined in [5],[6]. For the satellite channel at this low level of loss it is reasonable to assume the loss of packets to be independent.

Even for a given coded video stream the effect of losing different NALUs may lead to large differences in the resulting decoded video after error concealment. Considering the loss of packets to be independent, we further assume that the individual (FEC) packets lost do not interact. (For one lost FEC packet leading to one or more lost NALUs we assume recovery prior to the next loss and/or the effect of these to be additive as in [3].) Hence the effect of single FEC packet losses on the decoded video is analysed. To capture the variations, the FEC packets of one coded sequence are lost in turn and statistics gathered on the (loss) performance.

3 Simulation Results

The sequences utilized are *carphone*, *foreman*, and *mother and daughter (MD)* (30Hz, CIF resolution and 382, 400, and 400 frames, respectively). The H.264 reference

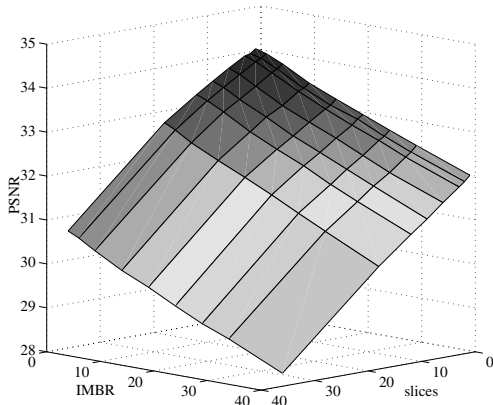


Figure 1. The relation between the average PSNR (error-free) and the number of IMBR and slices per picture is linear (*carphone*).

software (JM11) used the following settings: Main profile, NumberReferenceFrames = 2, SearchRange = 16, only P frames (except the first frame), CABAC is on, constrained intra prediction is on. The rate control is enabled, with target rate 256 kbit/s. The number of intra macroblock refresh and slices per picture are tuned by setting the parameters RandomIntraMBRefresh (IMBR) and SliceArgument (with SliceMode = 1, so that Slices = 396/ SliceArgument), respectively. In the decoder, error concealment of entire pictures is done in the ‘motion copy’ mode.

A first experiment was performed in order to evaluate the impact of the error resilience tools on the average PSNR, in the error-free case. All combinations of IMBR = {0, 2, 5, 10, 15, 20, 25, 40} and Slices = {1, 2, 3, 6, 9, 12, 18} are tested. The results for *carphone* are displayed in Fig. 1. These parameters has a linear relation to the average PSNR. For all sequences, increasing the number of slices by 1 unit produces reductions of the mean PSNR around 0.08 dB. The effect of IMBR depends on the amount of motion: for the fast motion sequences *carphone* and *foreman*, increasing the value of IMBR by 5 the PSNR is reduced about 0.3 dB, while for the static sequence *MD* the corresponding decrease is about 0.55 dB. The larger penalty in the latter case is expected, since static sequences are encoded very efficiently if inter prediction is fully utilized.

A second experiment was aimed at analyzing the impact of the loss of single packets. The experiment was carried out by removing FEC packets of 5, 10 and 20 ms (BGAN [2]) from the bitstream. One packet at the time is assumed to be corrupted, all the NALUs hit by the erroneous packet are removed, and the effect on the PSNR (i.e. the loss compared to the error-free case) is captured in an interval of 3 s from the occurrence of the error (i.e. 90 frames). The results for packets of 10 ms and settings IMBR=10 and Slices=9 for the sequence *carphone* are displayed in Fig. 2. As a first approximation, the average PSNR loss may be modelled as an exponentially decreasing function. The initial loss depends

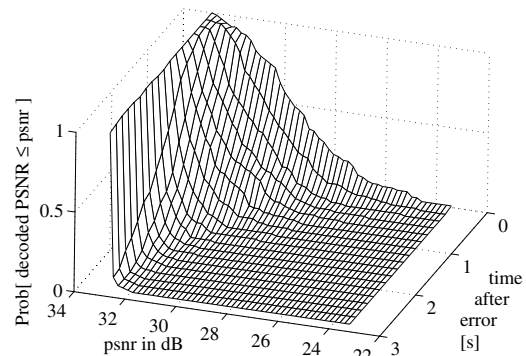
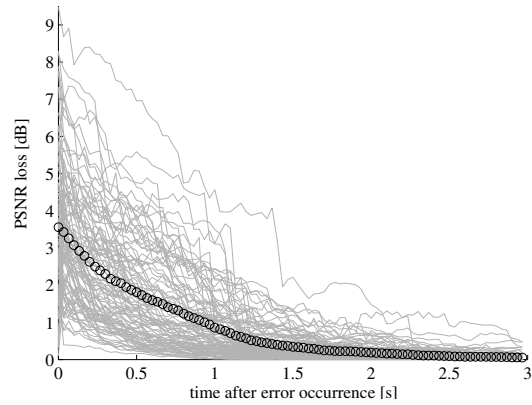


Figure 2. PSNR loss within 3 s from error occurrence (*carphone*, IMBR=10, Slices=9, 10ms packets), mean at each instant denoted as \circ (top). The corresponding distribution of instantaneous PSNR values (bottom).

on the number of slices and the size of the packet, while the slope of the exponential depends on IMBR (both initial loss and slope depend of course also on the type of sequence). We note the leaky nature of the loss extending beyond the point where all macro-blocks have been intra updated. For the packet sizes of 5, 10, and 20 ms, given PLR = 10^{-3} , errors occur in average every 150, 300, and 600 frames. Results are reported in Table 1 comparing settings that provide roughly the same error-free PSNR, for each sequence (in two cases packet lengths are assumed equally likely, since in the BGAN these cannot be controlled by the user and the authors do not have better statistics). These were chosen in order to: maintain the average PSNR, with average losses compared to the non resilient setting (IMBR=0, Slices=1) between 1 and 2 dB; keep the residual average PSNR loss after 90 frames from the error well below 1 dB, in order to validate the assumption that the loss of one packet is recovered before the occurrence of the next error.

An initial evaluation, aimed at relating the PSNR loss and the perceived duration of error propagation to the overall perceived quality is shown in Fig. 3. The displayed cases refer to observable effects of the loss of data, i.e. errors generating peak PSNR losses smaller than 2 dB were not

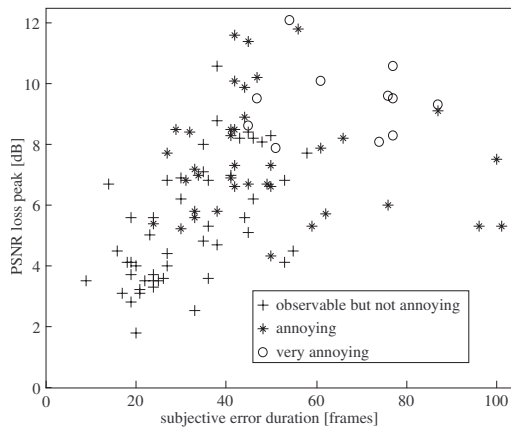


Figure 3. Subjective evaluation of observable effect of lost packets (*foreman*).

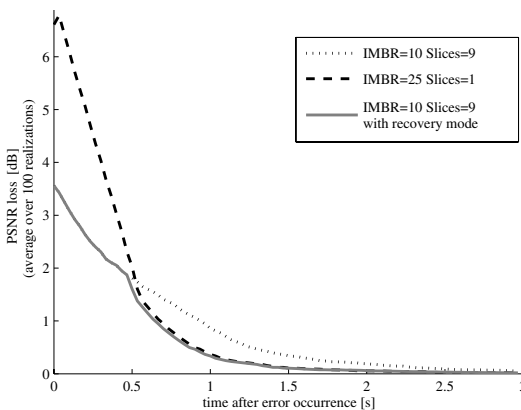


Figure 4. Recovery mode reduces PSNR loss after 500 ms (*carphone*, 10ms packets).

noticed, and errors that subjectively propagate for less than 500 ms are never considered annoying.

A last experiment was carried out for analyzing the improvement that can be obtained by using a feedback channel in order to signal to the encoder the loss of one packet. The feedback delay was assumed to be 500 ms. A simple solution was simulated, by allowing the encoder to switch to a new setting, named recovery mode, when the error is signaled. The recovery mode utilizes a higher value of IMBR in order to speed-up the recovery, and only 1 slice per picture such that the average PSNR (error-free) is maintained. An example is shown in Fig. 4, where the simulated loss profile is obtained by combining the profiles of the two settings. Results for the three sequences are reported in Table 1. By using the recovery mode the average PSNR is slightly improved, and a significant reduction of the PSNR loss after 500 ms is observed. This reduction should provide an important benefit in terms of visual quality evaluation (see also Fig. 3).

The setting IMBR/Slices = 10/9 is a robust solution over the different packet sizes, with good balance between error localization and recovery, providing the best average

PSNR results for *foreman* and *carphone* (Table 1). For the less critical sequence *MD*, slightly better average PSNR are obtained with setting 15/1, since in static sequences error concealment works well even when a whole picture is lost.

Simulation of the video coding when losing a FEC packet of 5, 10 or 20 ms has shed light on the influence on the video performance. Based on the simulations of the Turbo FEC system it is possible to include the effect on the FEC efficiency. By introducing various puncturing schemes in the Turbo coding the FEC rate may be changed. Consider the performance for the 10 ms packets and fix the signal-to-noise (SNR) at the level corresponding to a packet loss-rate of 10^{-3} . At this SNR level change the block size and rate of the Turbo coding, increasing the rate to 263 kbit/s for 20 ms packets and decrease it to 248 kbit/s for the 5 ms packets. This leads to changes in the PSNR of the coded video. For *foreman* 20 ms FEC leads to a 0.15 dB increase and 5 ms FEC to a 0.15 dB decrease compared with 10 ms FEC. The PSNR values of Table 1 could be adjusted accordingly.

4 Conclusions

An analysis of the quality of H.264/AVC video at low packet loss rates over BGAN was presented. The effect of intra-macroblock refresh and slicing was analysed, in terms of average and instantaneous PSNR values, for different FEC packet sizes. A robust setting for all packet sizes was found. The benefit of a (500 ms) delayed feed-back channel was demonstrated.

Acknowledgements

We would like to thank Thrane & Thrane A/S for the fruitful discussions on the topic.

References

- [1] Kumar, S. et al., "Error Resiliency Schemes in H.264/AVC Standard," *Elsevier J. of Visual Communication and Image Representation*, 17(2):425-450, April 2006.
- [2] European Organisation For The Safety Of Air Navigation, *SwiftBroadband Capabilities to Support Aeronautical Safety Services WP1: Technical Description and Application to ATS*, http://www.eurocontrol.int/nexsat/gallery/content/public/Library/AeroBGAN_Study_WP1_0k.pdf.
- [3] Stuhlmuller, K. et al., "Analysis of video transmission over lossy channels," *IEEE Journal on Selected Areas in Communications*, 18(6):1012-1032, 2000.
- [4] ISO/IEC Int'l Standard 14496-10, "Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding," 2005.
- [5] S. Wenger, "Common conditions for wire-line, low delay IP/UDP/RTP packet loss resilient testing," ITU-T VCEG document VCEG-N79r1, Sep. 2001.
- [6] Roth, G. et al., 2001. Common Test Conditions for RTP/IP over 3GPP/3GPP2, ITU-T SG16 Doc, VCEG-M77.

IMBR/ Slices per frame	packet size [ms]	mean PSNR		instantaneous PSNR loss after error occurrence				
		error free	PLR= 10 ⁻³	0	.5 s	1 s	2 s	3 s
0 / 18	5		31.85 (32.43)	2.1	1.4	1.2 (0.31)	0.9 (0.087)	0.7 (0.07)
	10	32.73	31.43 (32.46)	3.3	2.6	2.1 (0.42)	1.6 (0.110)	1.5 (0.10)
	20		30.75 (32.44)	5.3	3.9	3.3 (0.77)	2.9 (0.210)	2.7 (0.20)
2 / 18	5		32.01 (32.34)	1.9	1.2	0.9 (0.27)	0.56 (0.076)	0.4 (0.05)
	10	32.61	31.85 (32.38)	3.1	2.2	1.8 (0.37)	1.2 (0.110)	0.93 (0.08)
	20		32.00 (32.42)	4.9	3.7	3.1 (0.70)	2.0 (0.170)	1.5 (0.12)
10 / 12	5		32.30 (32.36)	2.3	1.1	0.50 (0.24)	0.13 (0.055)	0.046 (0.018)
	10	32.61	32.38 (32.44)	3.1	1.7	0.83 (0.33)	0.18 (0.063)	0.065 (0.021)
	20		32.44 (32.48)	4.6	2.7	1.20 (0.58)	0.22 (0.085)	0.062 (0.026)
10 / 9	5		32.52 (32.60)	2.5	1.3	0.64 (0.31)	0.14 (0.064)	0.056 (0.024)
	10	32.89	32.63 (32.70)	3.6	1.9	0.87 (0.33)	0.19 (0.060)	0.056 (0.018)
	20		32.66 (32.73)	5.6	3.3	1.80 (0.64)	0.35 (0.120)	0.16 (0.042)
15 / 6	5		32.50 (32.50)	2.9	1.3	0.34 (0.31)	0.054 (0.049)	0.016 (0.015)
	10	32.81	32.58 (32.61)	4.0	1.9	0.53 (0.33)	0.078 (0.045)	0.018 (0.010)
	20		32.65 (32.67)	5.4	2.8	0.75 (0.58)	0.100 (0.074)	0.026 (0.020)
20 / 3	5		32.45 (32.43)	4.3	1.5	0.21 (0.31)	0.017 (0.027)	0.005 (0.005)
	10	32.79	32.58 (32.57)	4.9	1.8	0.25 (0.33)	0.017 (0.032)	0.006 (0.008)
	20		32.65 (32.64)	6.0	2.4	0.41 (0.52)	0.043 (0.066)	0.019 (0.020)
25 / 2	5		32.24 (32.20)	5.1	1.4	0.19 (0.31)	0.045 (0.053)	0.01 (0.015)
	10	32.62	32.42 (32.40)	5.3	1.6	0.20 (0.30)	0.025 (0.038)	0.007 (0.012)
	20		32.49 (32.48)	6.4	2.0	0.27 (0.46)	0.043 (0.074)	0.011 (0.019)
25 / 1	5		32.17 (32.17)	6.8	2.1	0.39 (0.39)	0.038 (0.038)	0.007 (0.008)
	10	32.74	32.44 (32.44)	6.8	2.3	0.37 (0.37)	0.055 (0.055)	0.017 (0.017)
	20		32.55 (32.55)	8.1	3.0	0.58 (0.58)	0.087 (0.087)	0.03 (0.030)
0 / 18	5, 10, 20	31.86	31.10 (31.62)	3.7	2.7	2.03 (0.47)	1.44 (0.073)	1.09 (0.055)
2 / 18	5, 10, 20	31.73	31.20 (31.52)	3.5	2.4	1.80 (0.41)	0.97 (0.058)	0.68 (0.037)
10 / 9	5, 10, 20	32.06	31.68 (31.80)	4.5	2.7	1.28 (0.44)	0.33 (0.042)	0.102 (0.013)
10 / 12	5, 10, 20	31.77	31.46 (31.54)	4.1	2.4	1.14 (0.38)	0.28 (0.038)	0.103 (0.013)
15 / 6	5, 10, 20	31.99	31.69 (31.74)	4.7	2.3	0.78 (0.36)	0.11 (0.028)	0.0240 (0.0056)
20 / 3	5, 10, 20	31.92	31.64 (31.65)	5.3	2.3	0.48 (0.36)	0.028 (0.023)	0.0082 (0.0072)
25 / 2	5, 10, 20	31.73	31.49 (31.49)	5.4	1.7	0.23 (0.27)	0.011 (0.013)	0.0035 (0.0036)
25 / 1	5, 10, 20	31.88	31.62 (31.62)	5.6	2.1	0.30 (0.30)	0.012 (0.012)	0.0054 (0.0054)
0 / 36	5, 10, 20	36.95	36.37 (36.84)	1.1	0.90	0.85 (0.16)	0.83 (0.069)	0.81 (0.064)
5 / 18	5, 10, 20	37.96	37.78 (37.88)	1.3	1.01	0.77 (0.18)	0.41 (0.049)	0.19 (0.022)
10 / 9	5, 10, 20	38.04	37.91 (37.94)	1.5	0.95	0.45 (0.17)	0.078 (0.051)	0.047 (0.028)
15 / 3	5, 10, 20	37.96	37.84 (37.84)	1.9	0.94	0.16 (0.17)	0.031 (0.032)	0.028 (0.027)
15 / 2	5, 10, 20	38.06	37.92 (37.92)	2.1	1.11	0.18 (0.19)	0.039 (0.040)	0.034 (0.035)
15 / 1	5, 10, 20	38.18	38.01 (38.01)	2.5	1.27	0.22 (0.22)	0.050 (0.050)	0.040 (0.040)

Table 1. Objective quality evaluation on *carphone* (top), *foreman* (center) and *mother and daughter* (bottom), for different IMBR/Slices settings. The average error free PSNR values without using error resiliency, i.e. for the setting 0/1, are 34.05, 33.52 and 39.84, respectively. Mean PSNR and instantaneous PSNR loss (at {0, 0.5, 1, 2, 3} seconds after error occurrence) are reported. Values in (.) indicate the quality obtained using recovery mode. For *foreman* and *MD* results are reported assuming equally likely packet lengths (the relative performance between different packet lengths is similar as that provided for *carphone*).