Error Robustness Evaluation of H.264/MPEG-4 AVC

Till Halbach and Steffen Olsen

Department of Telecommunications
Norwegian University of Science and Technology (NTNU)
Trondheim, Norway

ABSTRACT

The robustness of the recently ratified video compression standard H.264/MPEG-4 AVC against channel errors is evaluated with the focus on rate distortion matters. After a brief introduction of the standard and an explanation of its error-resistant features, it is investigated how the error resilience tools of H.264 can be deployed best for packet-wise transmission as in ATM, H.323, and IP-based services. Further, the performances of two error concealment strategies for use in an H.264-conform decoder are compared to each other.

Keywords: Error robustness, resilience, concealment, H.264, MPEG-4 AVC, packet loss

1. INTRODUCTION

During the last decade, mobile communication and the demand for multimedia content have experienced unequaled rapid growth. The recent development in standardization of video compression is witness to the convergence of both areas. In May 2003, Recommendation H.264 was published by the ITU-T.[1] As the standardization efforts have been carried out on a joint basis with ISO/IEC, the standard is identical with International Standard 14496-10, more commonly known under the acronym MPEG-4 Part 10/AVC (Advanced Video Coding), released in October 2003.[2] The subsequent reference to H.264 includes hence always MPEG-4 AVC. H.264 is the first third-generation video coding scheme after the first generation with H.120, H.261 and MPEG-1, and the second generation which consists of H.263, MPEG-2 (Part 2) and MPEG-4 (Part 2).

The article is organized as follows. The standard is introduced, and the basic framework is explained. The article then turns to the standard's error resilience features which are explained in detail. After that, it is investigated in how far some of the features of H.264 can be utilized in erroneous environments. Additionally, the performances of two error concealment techniques for use with H.264 are discussed at the end. Finally, conclusions are drawn, and some areas for further study are mentioned.

Throughout this work, H.264's reference software version JM-4.2 is used, extended by a rate control algorithm.[3] The progressive-scan CIF-size sequences with a frame rate of 30 fps *Foreman*, *Container*, *Paris*, and *Mobile* are coded at different rates, passed to the channel simulator, and decoded. The well known $PSNR$ and the image quality index $SSIM$[4] are employed as error metrics.

For a more elaborate version of this work, it is referred to the literature.[5]

2. CHANNEL MODEL

The channel considered transports binary data in a packet-wise fashion. Examples for such channels are ATM- and IP-type (combined with UDP and RTP) networks and those specified by ITU-T Rec. H.323. Often, real-time constraints do not allow the use of feed-back channels and ARQ mechanisms. This means that the decoder in these environments have to be able to cope with either residual bit errors in one packet, or packet losses if the packet is discarded before decoding due to detection of errors by means of some check sum algorithm. Packet loss may also occur due to network congestion and buffer overflow. Up to 20% have been reported in the literature.[6]

In order to limit simulation efforts, a simple channel model is used in this work, which is sufficiently described by the packet loss ratio (PLR). Erroneous packets are always removed from the bit stream such that all packets reaching the decoder may be assumed error-free.

The authors can be reached by e-mail, {*halbach,steffeo*}*@tele.ntnu.no.*

3. BASIC CODING FRAMEWORK

H.264 is based on the block-based hybrid coding scheme known from previous standards of the H.26x and MPEG-x series. A spatial (INTRA, I) or temporal (INTER, P/B) prediction of the current signal is computed on the encoder side and subtracted from the original. This prediction error is then transformed, quantized and entropy-encoded, generating a bit stream to be conveyed over a channel or to be stored on a suitable medium. To reduce complexity requirements, H.264 works on so-called macroblocks (MBs) which consist of one 16×16-sample luminance block and – assuming a YCbCr color space and 4:2:0 chrominance subsampling of the input signal – two blocks of 8×8 samples for the color components. MBs can further be split into smaller units, blocks, depending on the amount of detail and motion in the sequence. The blocking artifacts along MB and block edges are efficiently removed by an anti-blocking filter.

H.264 consists like its predecessors of several sets of algorithms, also called profiles, of which one has to be chosen for a specific application. The specification of a particular profile is complemented by a so-called level which defines limitations on parameter values like picture and buffer sizes, and processing rate. All profiles have Baseline compatibility.

For a comprehensive overview of H.264, it is referred to the literature.[7]

4. ERROR RESILIENCE FEATURES

Single bit and burst errors are likely to propagate through the bit stream in the two profiles of interest due to prediction and entropy coding. On the encoder side, INTRA and INTER prediction reduce redundancy of the source prior to the transform step. I coding makes use of the samples of neighboring blocks of the same picture. A potential error may thus be able to propagate throughout the spatial domain. Additionally, the parameter containing a particular coding mode is differently encoded. In P coding, a motion vector and corresponding reference frame index are determined for the current block. The motion vectors are then differentially encoded with regard to neighboring vectors. Other predictions exist as well, as e.g. the one of the quantization parameter which specifies the quantizer choice for a set of MBs.

The quantized and zig-zag-scanned residual is coded with context-adaptive variable-length coding. This technique codes parameters like the number of coefficients of a 4×4-sample block, the number of zero coefficients, their values, run's, and their signs with a number of code tables that vary depending on the value of the mentioned parameters in neighboring blocks.[7] All other data, e.g. header information, is mapped to an exponential Golomb code.

Because of prediction, variable-length and run length coding, H.264 does not attempt to limit the impact of one or several bit errors; instead, packet losses are of major concern. As already mentioned, two of the three profiles defined so far are of interest with regard to error robustness. The Main profile is without error resilience features.

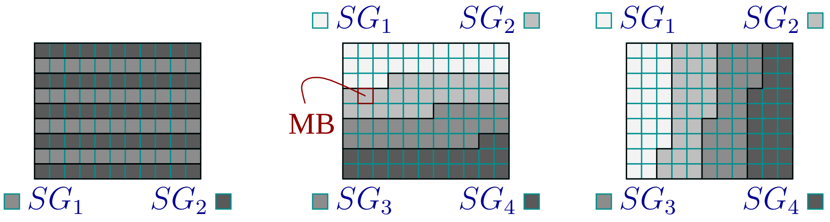
4.1. Baseline profile

The Baseline profile provides the standard's basic functionality including basic error resilience features.

Its main tool to resist transmission errors is the definition of slices and slice groups. One or several MBs are allocated to a slice group by means of an allocation map. A slice group is in turn composed of one or several slices. It is hereby possible to limit the size of slices according to the packet length requirements of a given network. Each MB belongs to exactly one slice which in turn may only consist of MBs of one picture. A slice is the smallest independently decodable unit in an encoded video. As such, any prediction referring to a spatial location beyond slice boundaries is not allowed, and e.g. the respective samples and motion vectors are marked as not available. The use of slices is aimed at stopping spatial error propagation, but simultaneously the coding efficiency is reduced as the prediction gain decreases.

H.264 knows effectively six different slice group (SG) allocation types, also known as flexible MB ordering (FMO). First, there is either a horizontal or vertical raster scan of MBs (Fig. 1(b) and Fig. 1(c), respectively), i.e. filling the slice groups row-/column-wise. This may be done in a forward or backward manner. A combination of both are rectangular slices which consist of contiguous areas of MBs. If carefully designed, rectangular slices

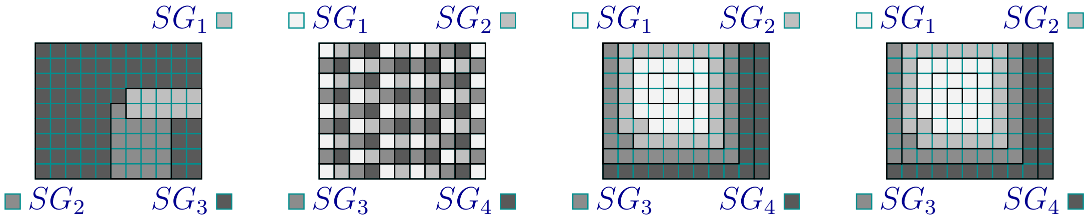
(Fig. 1(d)) do not reduce the coding efficiency so much but, nevertheless, bound the error impact to only a limited spatial area. Then, dispersive/scattered slices (Fig. 1(e)) are tailored for heavily interference-prone channels. Concentrated transmission errors are spread over the whole spatial plane and may efficiently be concealed by the decoder. However, this scheme reduces prediction gains significantly. The concept of interleaved slices follows the Group of Blocks structure in H.263, see Fig. 1(a). There is also the possibility of a clock-wise or a counter-clock-wise scan as depicted in Fig. 1(f) and Fig. 1(g), respectively. If one of these concepts should not suffice, MBs can be explicitly allocated to a slice group, one by one.



(a) Interlaced slice groups

(b) Horizontal raster scan

(c) Vertical raster scan



(d) Rectangular allocation

(e) Dispersive allocation

(f) Clock-wise spiral scan

(g) Counter-clock-wise spiral scan

Figure 1. Allocation types of MBs to SGs. For simplicity of illustration, the number of slices per slice group is set equal to one

According to the I and P MB principle, there are I and P slices. An I slice may contain only I MBs, and a P slice may be compound of both I and P MBs. Optionally, a *constrained* INTRA coding mechanism can be signaled to the decoder by means of a flag. If the flag is set, an I MB must not refer to samples that are generated by INTER coding, but may only rely on I-coded pixels.

There are three other error resilience techniques specified in the Baseline profile. Redundant slices allow the insertion of primary and one or more secondary slices in the bit stream. If a primary slice is affected by errors, it can be replaced by an error-free redundant one, otherwise the redundant slices are discarded. This feature is especially useful for transmission of slice header data, but also in e.g. a simulcast environment where the primary slices are coded with a high and the other slices with a low bit rate. Also, in order to account for the packet arrival characteristics of e.g. IP-type networks, the decoder allows the slices to arrive in arbitrary order. All slices of one frame must, however, have arrived before the decoder can start decoding of the next frame. Finally, messages to the decoder interleaved in the code stream containing supplemental enhancement information may contain further information about the bit stream, which can be utilized e.g. by an error concealment scheme.

4.2. Extended profile

In addition to the tools of the Baseline profile, Extended provides – among other tools – further error robustness features.

Data partitioning (DP) allows to group the encoded slice data into three partitions according to the classification header, INTRA-specific, and INTER-specific data. Each partition is in turn transmitted as one packet over the network. DP eases the use of UEP schemes that account for the non-uniform importance of the data as a reference for subsequent prediction. Moreover, as the decoding process is very sensitive to transmission errors in the slice header, header-specific data can be protected by strong channel codes without the otherwise typical considerable decrease in coding efficiency.

Finally, Extended allows the use of so-called switching I and P slices, or short SI/SP slices. This technique aims at stream switching, stream skipping, error resilience, etc. The philosophy beyond S slices is that a reconstructed S slice is always identical with e.g. an I, P, or B slice of another stream, such that decoding of the stream can continue even though a different reference, i.e. the S slice, is used. This is illustrated in Fig. 2. SI slices are self-contained without any temporal prediction, and SP slices correspond to P slices. For error resilience purposes, switching slices are mainly of interest in combination with a feed-back channel, where the decoder signals the encoder that a certain slice is affected by errors and thus not recommended for use as prediction reference. The encoder can then send an S slice which is coded with a reference that is available error-free at the decoder.

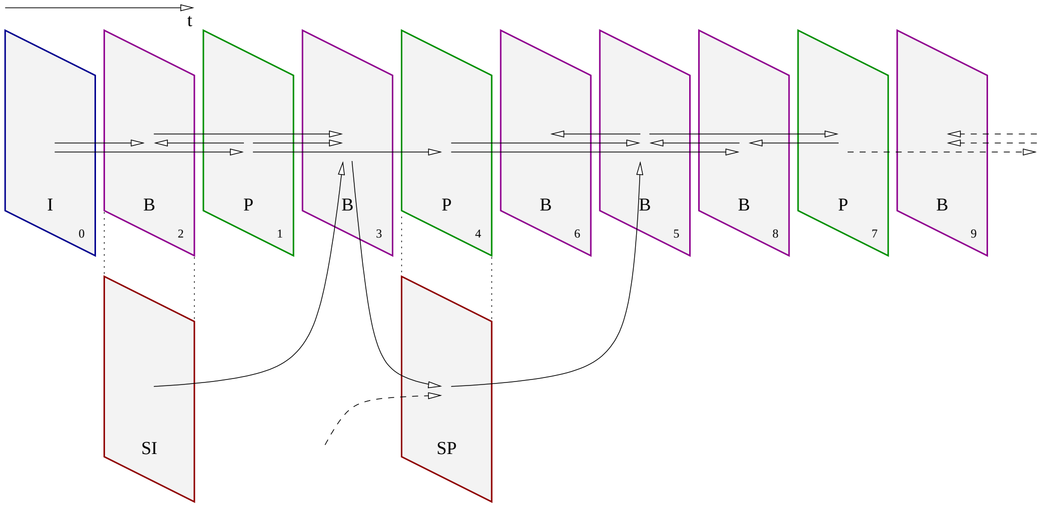


Figure 2. Example of several bit streams containing I, P, and B slices. B/bi-directional predictive slices are only known to the Main profile and hence not discussed here. For simplicity reasons, one slice is set equal to one frame.

4.3. Other resilience tools

A non-profile element to enhance the standard's error robustness is the split of the decoder, and hereby also the encoder, into two hierarchically grouped layers. The Video Coding Layer (VCL) is responsible for all coding matters as in conventional systems. It is the task of the lower-lying Network Abstraction Layer (NAL) to account for different applications like conversational communication types, video transmission by means of H.32x series packets, or considering RTP/UDP/IP-like communications. All the VCL needs to know is that the generated data is conveyed as a NAL unit. In such a unit, all encoded data, i.e. a complete slice or a partition, is encapsulated. This means, a unique marker precedes a NAL unit, and a decoder which has lost synchronization with the elements of the bit stream only needs to search the stream for that particular marker to synchronize again.

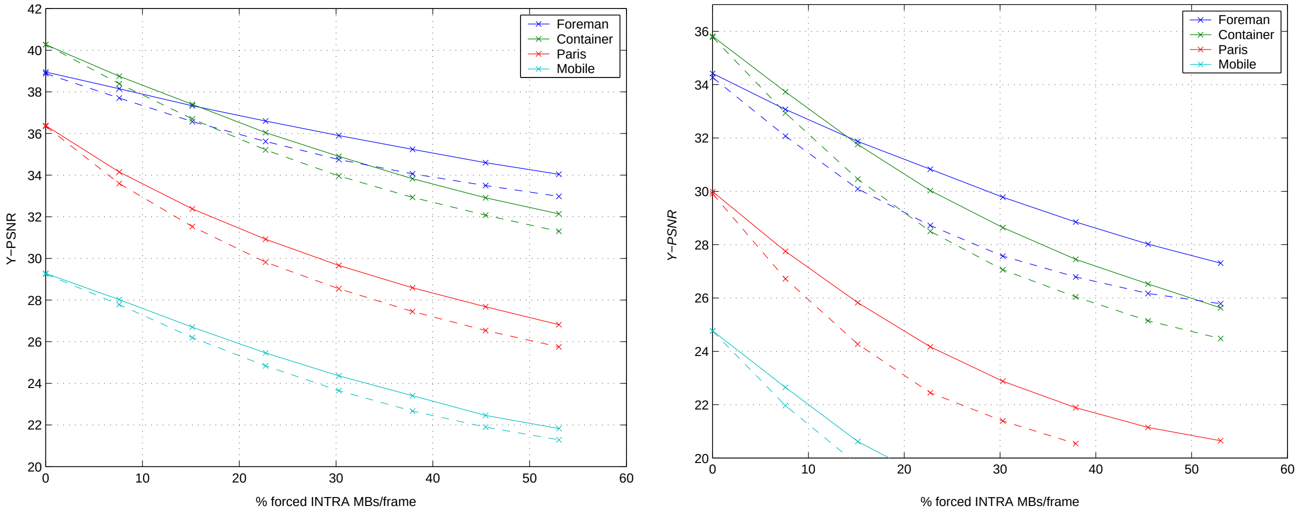
5. ERROR RESILIENCE CONSIDERATIONS

This section discusses several topics related to error resilience of H.264, especially with focus on the standard's rate distortion performance. Other literature[8, 9] exists with the focus on channel simulations.

5.1. INTRA update

Even though slice boundaries limit spatial error propagation within one frame, a potential transmission error may cross this boundary due to motion estimation/compensation over several frames. The well known so-called INTRA update is therefore employed to stop the error's spreading. In H.264, single MBs (and hereby arbitrary MB regions) and complete slices (and hereby a whole frame) can be coded in INTRA mode.

In constant bit rate (CBR) coding, frequent use of I MBs leads, however, to a significantly decreased image quality. The packet sizes of P slices are typical 10% of those of I slices.[7] Hence, slices are normally coded in INTER mode, and only single MBs are INTRA coded. MB INTRA update means, however, that propagation of a potential error may not be completely prevented at the decoder side. Fig. 3(a) shows how the $PSNR$ develops over a certain number of forced I MBs per frame at a CBR of 768 kbits/s, whereas in Fig. 3(b) 256 kbits/s are chosen.



(a) Bit rate of 768 Kbit/s

(b) Bit rate of 256 Kbit/s

Figure 3. Luminance $PSNR$ over the total number of I MBs per frame. The dashed lines denote the performance with constrained INTRA coding, the solid lines stand for the performance without constraints

It is observed that the $PSNR$ decrease is significant, ranging from roughly 5 dB for *Foreman* to 9 dB for *Paris*. *Foreman* has relatively few details and much motion, such that the degradation is not as strong as for *Paris* which has a very high degree of detailedness. The $PSNR$ gradient is higher at low rates due to the fact that then the decrease of coding gain due to use of I MBs is respectively higher than at high rates. Constrained INTRA prediction means that the curves are falling off stronger and show a more nonlinear behavior. There is at most only a 1 dB difference between the curves with and without constrained coding, such that its use can be recommended in error-prone environments without sacrificing the rate distortion performance too much.

5.2. Motion-sensitive INTRA update

There has been a lot of research on the topic of what is the optimum number of INTRA MBs per frame. E.g., a periodical INTRA refresh of video frames has been proposed,[10] or a periodical update of randomly chosen MBs.[11] Recent research has been concentrating on a rate distortion approach.[12–14] Most of the proposed algorithms cannot, however, be implemented in real-time, or only with a considerable amount of use of resources.

A simple solution that also implementers would accept is therefore to increase the number of I MBs per frame by a factor that influences the coding mode decision process on the encoder side. An often used low-cost prediction error metric is the summed absolute transformed difference ($SATD$). The coding mode decision can e.g. be done by Lagrange functional minimization $J = SATD + \lambda \cdot R$ with a rate constraint R and the optimization parameter λ. $SATD$ for a 4×4-sample block – the transform size in H.264 – is defined as $SATD = \frac{1}{2}\sum_{i,j=0}^{3} |T_H\{D(i,j)\}|$. $T_H\{\cdot\}$ is the 2-D Hadarmard transform, and the definition of the prediction error is $D(i,j) = X_{\text{org}}(i,j) - X_{\text{pred}}(i,j)$, with the original and predicted samples X, at the position column i in line j. The minimization is done for all INTRA- and INTER-frame MB coding modes (the latter one implying all reference frames).

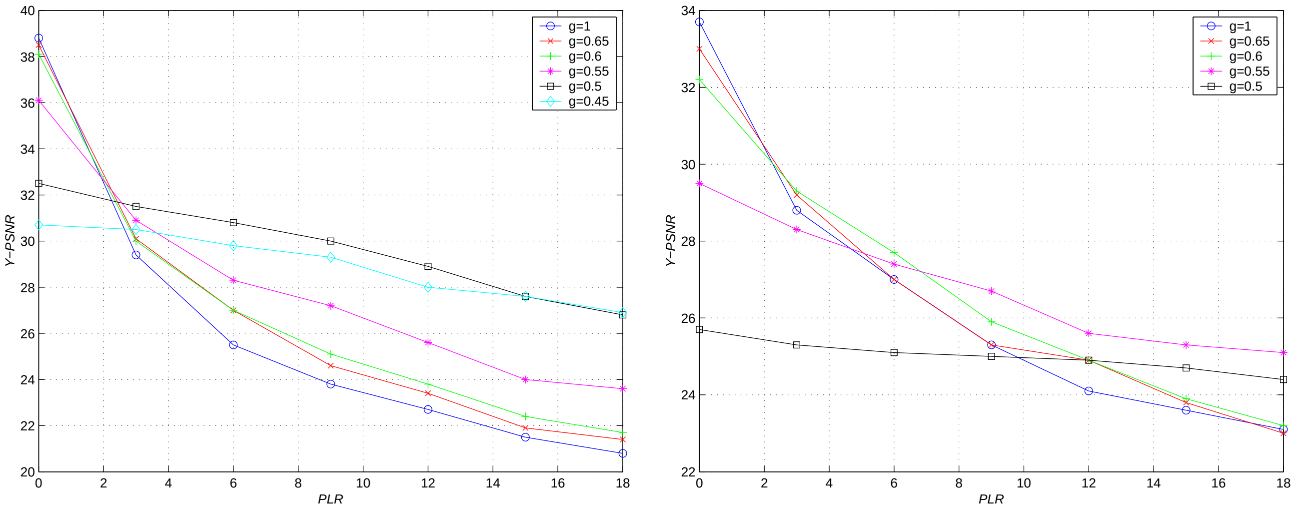
INTRA coding of a MB is chosen if the respective $SATD$ is less then those of all possible INTER coding modes, $SATD_{\text{INTER}} > SATD_{\text{INTRA}}$. With $SATD_{\text{INTER}} > g \cdot SATD_{\text{INTRA}}$, $g = 0, \ldots, 1$, it is possible to control the ratio of I MB in a P slice. This approach has besides simplicity the advantage that areas with much motion, which are often the regions of interest in a video, are likely to be refreshed quickly, whereas regions which are less interesting are INTRA-updated later. This is illustrated in Fig. 4. That is, $SATD_{\text{INTER}}$ is for low-motion material so small that $SATD_{\text{INTRA}}$ is almost always greater, and no INTRA update is made.

(a) Spatial and temporal concealment coupled with random constrained INTRA updates. 2 slices per picture. Fourth frame after error occurrence

(b) Motion-sensitive constrained INTRA update without error concealment. 6 slices per picture. First frame after error occurrence. The green color is due to the transform of zero sample values from the YCbCr to the RGB domain

Figure 4. Comparison of rate-distortion-combined INTRA update and random INTRA update

The proposed approach has the disadvantage that the degree of increase of error robustness depends much on the source statistics. The optimal values of g have therefore to be determined for typical video characteristics. As the curves in Fig. 5 show, the most reasonable values for g are in the range $0.4 \leq g \leq 0.6$. The use of lower values cannot be justified since they do not yield the highest $PSNR$ at high PLRs. For $g > 0.6$, the curves show no remarkable difference in behavior. It is further observed that the higher the rate, the more gain is achieved by the new approach at high PLRs. Another observation is that the $PSNR$ is the higher the more motion the image sequence contains. It should also be mentioned that the new approach do not work with image material with a low degree of detailedness and few motion as e.g. *Container*, provided that temporal error concealment is applied to lost MBs. Here, only a degradation in $PSNR$ is achieved for the PLR range of interest. The reason for this is the good performance of the concealment technique. The samples of a lost MBs are replaced by the samples at the same location from the adjacent previous frame, also referred to as CPYMB. In Fig. 4 on the right side, the area in the middle is repeatedly refreshed, and areas with few or no motion less often. However, the green blocks are usually efficiently concealed, and hence the motion-sensitive update takes into account that regions with a lot of motion are more important as they contain much new information.



(a) *Foreman* at 768 kbits/s

(b) *Paris* at 512 kbits/s

Figure 5. $PSNR$ over PLR with g as a parameter.

As the focus of this work is mainly on video conferencing, the optimal values of g are determined for *Paris* and shown in Tab. 1. As table data and curves of Fig. 5 show, unfortunately no general rule can be set up, and it is thus recommended to tabulated g as a function of rate and packet loss ratio. It is noted that the lower bound for g rise with decreasing rate. The resolution interval of 3 for the PLR is a good compromise between the requirement for a high resolution at high rates and a low resolution at low rates.

Rate	PLR [%]					
[kbit/s]	[0–3)	[3–6)	[6–9)	[9–12)	[12–15)	[15–18)
768	$g > 0.66$	0.52	0.48	0.44	0.44	0.44
512	$g > 0.66$	0.56	0.56	0.56	0.52	0.52
256	$g > 0.66$	0.66	0.64	0.64	0.62	0.62

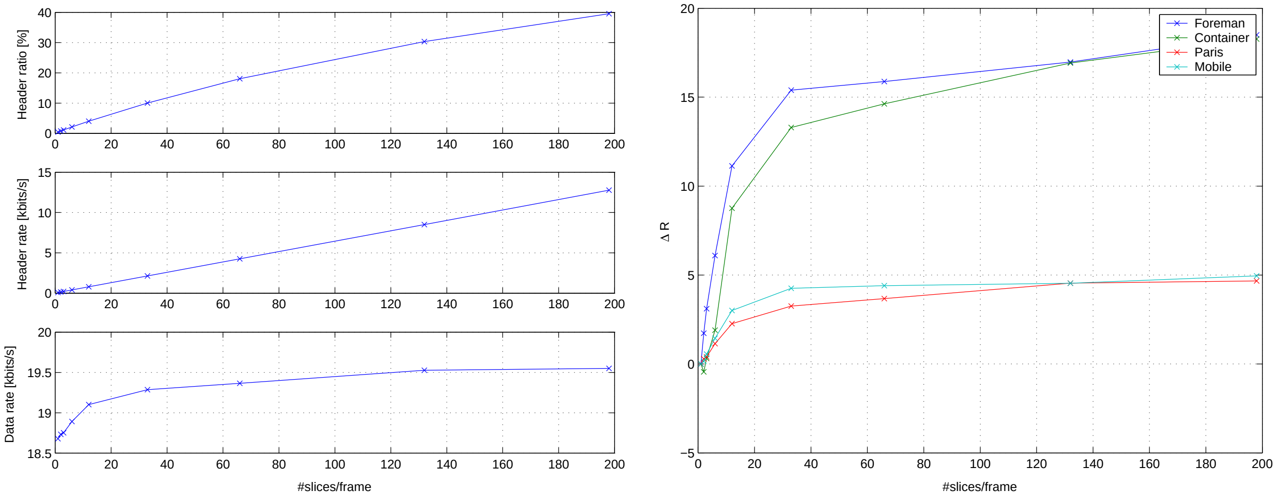
Table 1. Tabulated optimal values of g for the sequence *Paris*. Otherwise mentioned g equals the value given

5.3. Slice sizes

Frequent use of slices leads either to a significant rate increase if encoding is quality-controlled or to a degradation in $PSNR$ if rate control is enabled. This is partly due to limited prediction and partly due to a larger rate consumption by slice header data with regard to the total rate use.

As plotted in Fig. 6(a), the slice header rate increases linearly, the larger the number of slices (plot in the middle). Here, the quality ($PSNR$) is kept constant with a quantization parameter of 28. For the often used horizontal-raster-scan slice group allocation as depicted in Fig. 1(b), the data rate increases in a strongly non-linear manner (bottom curve) with decreasing slice size. However, the curve flattens out when the point is reached where a slice is set to be a complete line of MBs, for CIF-size material 22 MBs. Then, there is only horizontal inter-slice prediction, and the $PSNR$ decrease is small for even shorter slices. The upper curve shows the increase of header ratio which is defined as the ratio between header rate and total rate.

The data rate increase ΔR values of all sequences are plotted in Fig. 6(b) for up to 198 slices per frame, which corresponds to 2 MBs per slice. It is observed that the increase is strongest for *Foreman* and *Container*



(a) Header rate increase

(b) Data rate increase with respect to 1 slice per frame

Figure 6. Rate increase of slice header and data, assuming a horizontal raster allocation. All numbers are averaged over the first 100 frames of the sequence. *Paris*, P slice, constant $PSNR$

which both rely on efficient coding (prediction) of their motion vectors which describe quite uniform motion. The maximum rate increase is roughly 5% for *Mobile* and *Paris*, and approximately 18% with *Foreman* and *Container*. For INTER coding, the data rate increase of sequences with much motion will generally be high, as the prediction gain of motion vectors deteriorates. The same applies to INTRA coding, but here due to the fact that much motion means an increased ratio of I-coded MBs of which the prediction will become worse with smaller slices. It is, however, better to accept a rate increase due to the use of many slices than to accept the increase due to frequent use of I MB updates since, in the latter case, a much larger image region is covered by that respective slice, and if it is lost, it is more difficult to conceal than a small region.

6. ERROR CONCEALMENT STRATEGIES

Decoder issues like error concealment (EC) are mentioned as informal supplements and are not an integral part of the standard. However, features like the aforementioned dispersive MB allocation have obviously been developed with concealment in mind. The area of lost slices is hereby spread over the whole picture, and lost MB samples can be estimated based on neighboring MBs (spatial EC) or based on previous or later pictures (temporal EC).

A reasonable extension of the standard is therefore to log for each MB if it could be correctly decoded, and invoke a concealment algorithm if not. It is advisable to assume all MBs as lost before the beginning of decoding of a new frame. Each MB of that frame, which can be decoded without error, is marked as being correct, and before the decoder turns to the next frame, all other MBs are concealed. As H.264 differentiates between I and P slices, spatial or temporal concealment is applied to lost MBs. Spatial concealment can be done as e.g. weighted pixel averaging.[15] For temporal concealment, a most likely motion vector of a lost MB is computed as the motion vector average which gives the lowest pixel difference along the MB edge.[16]

The performances of two schemes for temporal concealment are investigated in this work. The first scheme, called zero-vector concealment, copies the corresponding MB at the same location from the temporally adjacent previous frame. This technique implies almost no computational resources and works fine at high frame rates and a low to moderate amount of motion in the sequence. A visually more pleasing impression is accomplished if motion vector averaging is employed. However, vector averaging comes at the cost of a somewhat increased computational complexity. Both schemes allow the re-use of concealed regions as reference to other areas.

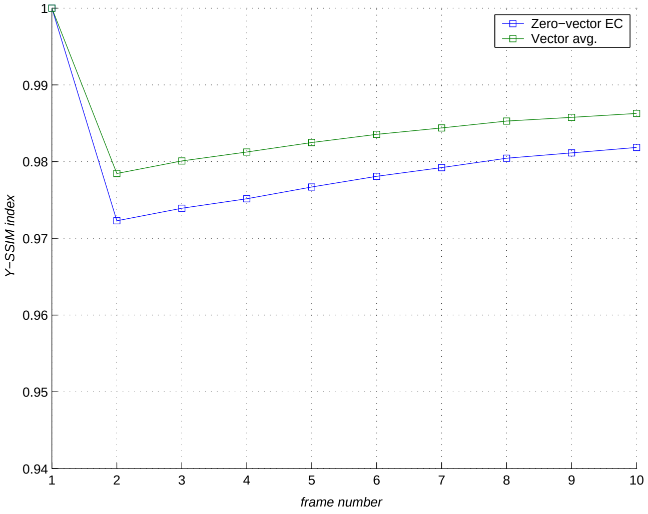


Figure 7. A comparison of EC method in terms of visual quality, measured by the $SSIM$ index[4]

In order to compare both EC methods, the test sequences are coded with 3, 4, 6, 9, and 12 slices per frame. For each of these horizontal-raster-scan MB allocations, the $SSIM$ index values (per frame) are averaged over all possible single-slice losses. In Fig. 7, the slice loss is in frame 2, and the development of $SSIM$ over 10 frames is shown. Vector averaging performs at least as well as zero-vector concealment for videos with few motion and outperforms the latter scheme considerably with high-motion sequences like *Foreman*. A visual comparison is given in Fig. 8, where a frame consists of 6 slices, and where the MBs are allocated according to the scheme of horizontal raster scan. Two slices are lost. It is easily verified that also here vector averaging perform superior to the zero-vector scheme.

(a) Vector average EC

(b) Zero-vector EC

Figure 8. Image quality three frames after error occurrence. The originally two lost slices are marked by rectangles

7. CONCLUSIONS AND OUTLOOK

Compared to previous standards, I coding has been considerably improved in H.264. The same applies to P coding, and as a consequence the rate increase from INTER to INTRA coding is significant. Coding of a complete frame as I frame to stop potential error propagation can therefore not be recommended. Moreover, the penalty in terms of quality degradation when using constrained INTRA coding in the error-free case is only small, and this feature should hence always be enabled in erroneous environments. It has also been shown that the visual performance of a constrained INTRA MB update is further improved by making the mode decision algorithm sensitive to the video's motion. Important areas (those with much motion, which the human eye is usually attracted to) are coded in constrained INTRA mode and can hereby be preserved in highly error-prone environments.

As already mentioned, the proposed INTRA update refreshes regions with much motion repeatedly. However, it is not of advantage to choose I coding for many temporally adjacent MBs at the same spatial location as then the coding gain will be considerably reduced. A topic for future research is hence to take into account the rate consumption over several frames and the hereby associated distortion in the presence of errors by a look-back approach.

The slice structure offers various possibilities for impact limitation of channel errors. The rate increase by frequent use of (small) slices should be prefered to the rate increase introduced by INTRA updates. Slice structures like rectangular slices, which do not impair the various prediction processes of the standard are of advantage.

Concerning error concealment, motion vector averaging can be recommended as its use means only a minor increase in computational complexity, while the visual evaluation reveals a considerable performance improvement compared to zero-vector concealment.

REFERENCES

1. ITU-T, *Advanced video coding for generic audiovisual services.* Recommendation H.264, May 2003.
2. ISO/IEC, *Information Technology – Coding of Audio-Visual Objects – Part 10: Advanced Video Coding.* International Standard 14496-10 (MPEG-4 Part10/AVC), Oct. 2003.
3. M. Siwei, W. Gao, Y. Lu, and H. Lu, "Proposed draft description of rate control on JVT standard," Tech. Rep. F086, ITU-T VCEG | ISO/IEC MPEG (JVT), Dec. 2002.
4. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity." Submitted to IEEE Transactions on Image Processing (Jan. 2003).
5. S. Olsen, "Error robustness evaluation of H.264/MPEG-4 AVC," diploma thesis, Norwegian University of Science and Technology (NTNU), Trondheim (Norway), June 2003.
6. Q. F. Zhu and L. Kerofsky, "Joint source coding, transport processing, and error concealment for H.323-based video," in *Proc. SPIE's Visual Communications and Image Processing (VCIP)*, **3653**, pp. 52–62, (San Jose (CA, USA)), Jan. 1999.
7. T. Halbach, "The H.264 video compression standard," in *Proc. Norwegian Signal Processing Symposium (NORSIG)*, (Bergen (Norway)), Oct. 2003.
8. B. Jung, Y. H. Hwang, B. Jeon, M. D. Kim, and S.-I. Choi, "Error resilient performance evaluation of MPEG-4 and H.264," *Proc. of SPIE* **5150**, pp. 1050–1061, July 2003.
9. C. Calafate and M. Malumbres, "Testing the h.264 error-resilience on wireless ad-hoc networks," in *Proc. EURASIP Video/Image Processing and Multimedia Communications Conference (VIPMCC)*, pp. 789–796, 2003.
10. T. Turletti and C. Huitema, "Videoconferencing on the internet," *IEEE/ACM Transactions on Networking* **3**, pp. 340–351, June 1996.
11. G. Côté and F. Kossentini, "Optimal intra coding for blocks for robust videocommunications over the internet," *Image Communication* **1**, pp. 25–34, Sept. 1999.
12. R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *J. Special Areas Comm.* **18**, June 2000.
13. K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Select. Areas Commun.* **18**, pp. 1012–1032, June 2000.

14. G. Côté, S. Shirani, and F. Kossentini, “Optimal mode selection and synchronization for robust video communications over error-prone networks,” *J. Special Areas Comm.* **18**, pp. 952–964, June 2000.
15. P. Salama, N. Shroff, and E. Delp, *Error concealment in encoded video streams*, ch. 7. Kluwer Academic Publishers, 1998.
16. W.-M. Lam, A. R. Reibman, and B.Liu, “Recovery of lost or erroneously received motion vectors,” in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, **1**, pp. 417–420, (Minneapolis (USA)), Apr. 1993.