Proc. of Int. Conf. on Systems, Signals and Image Processing (IWSSIP), Budapest, Hungary, Sept. 2006.

Robust Error Detection for H.264/AVC Using Relation Based Fragile Watermarking

Olivia Nemethova, Gonzalo Calvar Forte and Markus Rupp

Institute of Communications and Radio-Frequency Engineering
Vienna University of Technology
Gusshausstrasse 25/389, 1040 Vienna Austria
{onemeth, gcalvar, mrupp}@nt.tuwien.ac.at

Abstract - In this paper a robust mechanism for detecting errors in H.264/AVC video streams is proposed and evaluated in the context of transmission over mobile networks. The proposed algorithm utilizes fragile watermarking applied at macroblock level for the detection of errors in the decoded VLC stream. Watermarking based approaches are compared with a previously proposed alternative error detection mechanism using parity bits. Two watermarking schemes are investigated – forced even watermarking and a novel relation based watermarking. The obtained results show considerable improvement of the quality at the decoder for the proposed watermarking method, while preserving the data rate.

1. INTRODUCTION

For transmissions of video over packet networks (both fixed and mobile), usually low resolutions of 176×144 pixel (QCIF) or 320×240 pixel (QVGA) are used. In such small resolutions each pixel represents an essential part of a picture. To avoid the effect of a rather high overhead used by RTP/UDP/IP protocols, the encapsulated video part is usually large [1]. In nowadays systems typically the UDP protocol provides the parity check information. If an error is detected, the entire packet is discarded and an error concealment routine is called. The larger the degraded area, the more difficult it is to conceal an error. Thus, the packetloss inevitable in real-time communications in error-prone environments, leads to a considerable perceptual visual quality degradation.

The entire packet is discarded even if only a one bit error occurs. This is performed mainly due to variable length coding (VLC) used to maximize the entropy of the source. As soon as an error occurs the VLC will desynchronize – the boundaries of the codeword will be determined erroneously. Decoding after desynchronization is erroneously possible until the first invalid VLC entry is found. Nevertheless, the usage of the correctly received part of the packet up to the desynchronization point could improve the received quality substantially (depending on the position of the first error).

Several methods were proposed up to now trying to identify and utilize correctly received parts of the stream. In [2] knowledge about the video stream syntax (ranges of the information elements i.e. coefficients, motion vectors, etc.) is used to detect the errors. This method does not have any additional costs, but its detection performance is quite low. In [3] a fragile watermarking scheme was proposed, based on force even watermarking. An alternative was proposed in [4], utilizing the parity check over some smaller segments of the lost packets present in the lower layers. In [5] a parity check over a fixed sized picture area was introduced and moreover, the possibility of signalling the synchronization points to resynchronize the VLC *after* the error occurence was investigated. Another possibility is the sequential decoding of VLC that does not require any higher rate and not only detects, but also corrects parts of errors [6].

In this work we present the effect of the parity bits on the rate increase and investigate the error detection mechanism using fragile watermarking. In [3] forced even watermarking (FEW) was applied to H.263 discrete cosine transformation (DCT) coefficients. We adapted and implemented this scheme for the H.264 Joint Model [7]. To obtain better scalability, we further propose an alternative novel relation based watermarking (RBW) scheme and compare it to FEW. The results clearly confirm its suitability for the error detection purpose.

The paper is structured as follows: Section 2 analyzes performance of the error detection mechanism using parity bits. Section 3 presents the method using two different fragile watermarking schemes – forced even watermarking and the novel proposed relation based watermarking. Section 4 provides performance evaluation of the detection mechanism by means of the rate and the distortion at the encoder and decoder. Section 5 contains conclusions and some final remarks.

2. PARITY CHECK

In [5], we proposed a method based on adding of k parity bits over groups of M VLC encoded macroblocks to detect the errors. The advantage of such scheme is the detection within uniformly sized picture segments, rather than uniformly sized parts of a bitstream. In this case the rate is inverse proportionally to the rate of the original video stream. The rate increase $R_{inc} = \frac{R_{new}-R}{R} \cdot 100[\%]$ caused by adding 1 bit of parity check to each non-skipped macroblock for the "carphone" video sequence in QVGA resolution can be seen for different quantization parameter (QP) values in Figure 1. Adding more parity bits results in multiples of the rate increase presented for adding one parity check bit.

A single parity bit is capable of detecting any odd number of errors – bit inversions – within a b bits large macroblock. If P_b is the bit error probability (BEP) of the binary symmetrical channel (BSC), then the probability of an error within a macroblock is given by $P_{MB}^{(1)} = 1-(1-P_b)^b$. The probability of exactly l errors within a macroblock is $P_{MB}^{(l)} = \binom{b}{l} P_b^l (1-P_b)^{b-l}$. Therefore, the probability P_{res} of non-detected error within a macroblock with a single parity bit is

$$P_{res} = 1 - \sum_{k=0}^{(b+l-1)/2} \binom{b}{2k+1} P_b^{2k+1} (1-P_b)^{b-2k-1}. \qquad (1)$$

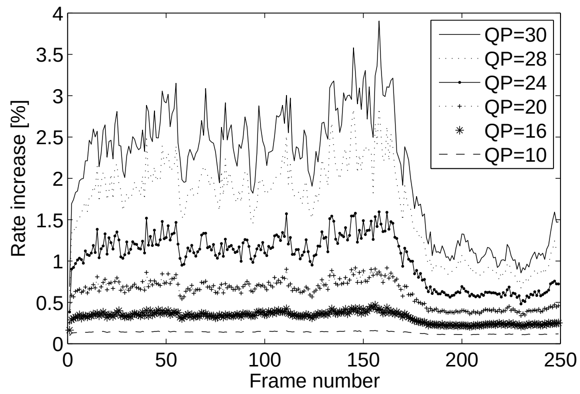

Figure 1: Rate increase caused by adding 1 bit of parity check to each non-skipped macroblock (”carphone” video sequence in QVGA resolution).

Note that the BEP in our case is not the channel BEP but rather the BEP caused by the VLC desynchronization. Thus the probability of the residual error for a single parity check lies around 50% (49% for the ”carphone” sequence and $\mathrm{BEP} = 1 \cdot 10^{-6}$).

3. WATERMARKING

Watermarking introduces distortion in the encoded stream to facilitate error detection. This can be equivalently understood as a rate increase since watermarking removes a part of the information from the stream. To transport the original data with the original quality and watermarking – a higher rate would be necessary.

To facilitate an efficient error detection, fragile watermarking is needed. A watermark is fragile if it changes with the smallest change of the watermarked picture. One of the well-known fragile watermarking schemes use changes in the least significant bit (LSB) of the coefficients or pixels. A forced even watermarking scheme was introduced in [3] to detect the errors in H.263 as summarized in the next section.

3.1 *Forced Even Watermarking*

According to [3], the FEW is inserted in each 8×8 pixels large H.263 block containing DCT coefficients a_i after quantization. Each a_i with $i = p \ldots N^2$ is transformed in a watermarked coefficient $a_i^{(w)}$ according to the following relation

$$a_i^{(w)} = \begin{cases} a_i & ; \quad |a_i| \bmod 2 = 0 \\ a_i - \mathrm{sign}(a_i) & ; \quad |a_i| \bmod 2 = 1, \end{cases} \tag{2}$$

where mod 2 denotes the modulo 2 operation and

$$\mathrm{sign}(a_i) = \begin{cases} 1 & ; \quad a_i \geq 0 \\ -1 & ; \quad a_i < 0. \end{cases} \tag{3}$$

In H.264/AVC the DCT transform is applied to blocks of size 4×4. We applied the FEW idea to these blocks. The probability of detection using FEW strongly depends on the start position p and on the content of the video sequence, since due to efficient prediction mechanism, most of the coefficients (especially the high frequency ones) in the blocks are zero. The probability density function of the coefficient values follows a Laplacian distribution $f(x) = \alpha \exp(-2\alpha|x|)$ with parameter α depending on QP [8] (for $QP = 30$ we obtained $\alpha = 0.5$ for H.264/AVC). Zeros are encoded in a run length manner [9] and thus no watermarking can be applied in such case. The errors in the codeword encoding the run can be in most cases successfully detected by a syntax check.

In [3] it was already shown that FEW brings considerable improvement over the simple syntax check mechanism. However, its detection possibilities are limited. If an error occurs resulting in an even coefficient, it remains undetected. Moreover, FEW allows to control the distortion only by means of changing p. From the point of view of the error detection probability, however, inserting the watermark in the significant regions (low frequencies) is beneficial. To improve the error detection probability, in the following section we propose a novel scalable fragile watermarking approach, allowing for better control of the distortion.

3.2 *Relation Based Watermarking*

Relation based watermarking modifies n chosen DCT coefficients $\underline{c} = (c_1, c_2, \ldots, c_n)$ to $\underline{\tilde{c}} = (\tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n)$ so that $\underline{\tilde{c}}$ fulfills the predefined equation of the extracting function $f(.)$:

$$f(c_0, \tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n) = K. \tag{4}$$

Note that c_0 is the DC component that is used in the extracting function. It remains unchanged to avoid the high distortion its change could cause. From the possible solutions for $\underline{\tilde{c}}$ the one that minimizes the distortion is chosen

$$\underline{\tilde{c}} = \arg\min_{\underline{\tilde{c}}} \|\underline{\tilde{c}} - \underline{c}\|. \tag{5}$$

The even better can be advantageously replaced by minimization of the distortion in the pixel domain, for example:

$$\underline{\tilde{c}} = \arg\min_{\underline{\tilde{c}}} \|\mathrm{IDCT}(\underline{\tilde{c}}) - \mathrm{IDCT}(\underline{c})\|, \tag{6}$$

where IDCT denotes the operation of re-scaling and inverse DCT. Selection of the extracting function $f(.)$ determines the distortion and detection probability. To minimize the distortion a function is required that provides many solutions near to the original coefficients. Thus, the number n of the coefficients that can be modified to match the desired extracting function influences the minimum distortion. For the probability detection, fragileness is required – functions that change by single error are necessary, robust against multiple errors. To meet these criterions we chose a modulo M operation over the sum of the selected coefficients:

$$f(c_0, \tilde{c}_1, \tilde{c}_2, \ldots, \tilde{c}_n) = \left(c_0 + \sum_{i=1}^{n} \tilde{c}_i \right) \bmod M = K. \tag{7}$$

Increasing of M will improve the fragileness of the scheme against multiple errors but on the other hand it will increase the distortion.

4. PERFORMANCE EVALUATION

All our experiments we performed with the test video sequence ”carphone” (like in [3]) in QVGA (320×240 pixels) resolution, 250 frames and full frame rate of 30 fps. We used slicing mode 2 with maximum 1200 bytes per slice. We encoded the sequence with different quantization parameters

(10,16,20,24,28,30) and disabled rate-distortion optimization. Only the first frame was intra coded (IDR), the rest of the sequence were P frames (we avoided usage of B frames to align with the baseline profile of H.264) to show the effect of error propagation. No data partitioning has been used and context adaptive VLC (CAVLC) was chosen as an entropy coding. We used the Joint Model software [7] with temporal error concealment that conceals the lost part of the frames by the corresponding area from the previous frames. We implemented FEW for every DCT block in a macroblock and RWB only for the first non-zero block in a macroblock.

4.1 *Impact on the rate*

After applying FEW the compression becomes more efficient. This is mainly because of the high probability of coefficients with absolute value one in the stream, becoming zero in the watermarked stream. The reduction $R_{red} = \frac{R-R_{new}}{R} \cdot 100$ [%] of the bitrate R (compressed) to rate R_{new} (compressed and watermarked) is shown in Table 1. Unlike

QP	p=2	p=4	p=7	p=11	p=14
10	30.13	27.26	22.08	14.40	7.49
16	42.45	37.32	20.82	16.99	8.01
20	65.92	71.78	20.82	11.40	4.85
24	30.75	24.54	17.17	9.02	3.49
28	31.97	24.10	16.41	7.69	2.73
30	31.48	23.15	15.05	6.94	2.01

Table 1: Average bitrate reduction in % after applying FEW to the ”carphone” video.

FEW, RBW causes a rate increase. The rate increase R_{inc} is presented in Table 2. It is caused by forcing different and sometimes higher values to the DCT coefficients.

QP	M = 4			M = 6		
	n=4	n=5	n=10	n=4	n=5	n=10
10	1.22	2.01	3.09	1.87	1.73	4.38
16	9.30	10.95	5.79	14.44	12.12	17.74
20	25.03	27.85	36.05	44.04	37.70	43.02
24	32.78	34.34	46.57	72.94	63.04	72.07
28	39.66	41.54	55.38	111.62	95.56	107.35
30	47.69	50.23	64.58	151.85	131.25	147.69

Table 2: Average bitrate increase in % after applying RBW to the ”carphone” video.

4.2 *Distortion at the encoder*

To measure the distortion in the mth frame caused by watermarking, we use the peak signal to noise ratio (PSNR) of the luminance as we only insert watermarking into the luminance:

$$\text{Y-PSNR}[m] = 10 \cdot \log_{10} \frac{(2^q - 1)^2}{\text{MSE}[m]} [\text{dB}], \qquad (8)$$

where q represents the number of bits used to express the luminance values and the mean square error (MSE) is given by

$$\text{Y-MSE}[m] = \frac{1}{N_1 \cdot N_2} \sum_{i=1}^{N_1} \sum_{j=1}^{N_1} [\mathbf{F}_m(i,j) - \mathbf{R}_m(i,j)]^2, \qquad (9)$$

where $\mathbf{F}_m$ is the degraded frame (compressed and/or watermarked), $\mathbf{R}_m$ is the original frame and $N_1 \times N_2$ is their size.

In Figures 2 and 3 the distortion caused by watermarking at the encoder is shown for FEW and RBW respectively.

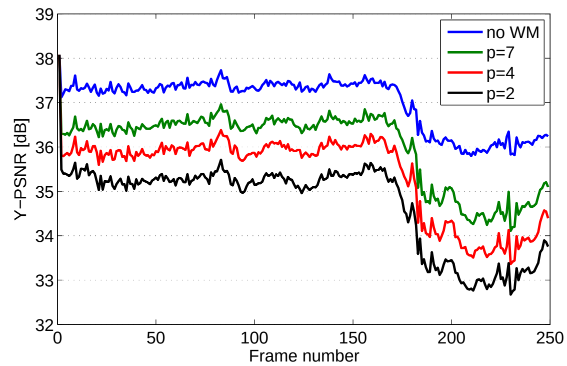

Figure 2: Y-PSNR per frame at the encoder for ”carphone” sequence compressed with QP=28 and inserted FEW for $p = 2,4,7$ and without watermarking.

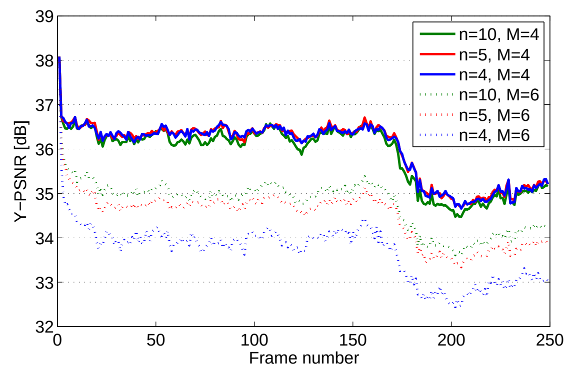

Figure 3: Y-PSNR per frame at the encoder for ”carphone” sequence compressed with QP=28 and inserted RBW for $M = 4,6$ and different n.

The distortion caused by FEW with $p = 7$ is similar to the distortion caused by the RBW with $M = 4$ and $n = 4,5,10$. It decreases the quality on average of about 1-1.5dB, that grossly corresponds to less than one level of QP.

4.3 *Distortion at the decoder*

The most important performance parameter is the quality at the receiver. To test the robustness of the investigated methods we simulated transmissions of the video stream over an error-prone environment by binary symmetrical channel (BSC) with bit error probability of BEP $= \{10^{-4}, 10^{-5}, 10^{-6}\}$. If an error is detected, error concealment is applied from that position until the end of the slice. Our absolute results for FEW are worse than those presented in [3]. This is caused by the difference between the used codecs and by the fact that for our experiments we did not implemented the syntax check algorithm yet, that would improve the performance of FEW and RBW when combined. However, the difference in performance of the investigated methods can be clearly seen, not only from the decoder quality results, but also from the probabilities of the error detection. In Figure 4 and 5 the Y-PSNR for the BEP $= 10^{-6}$ is shown using FEW and RBW respectively. RBW per-

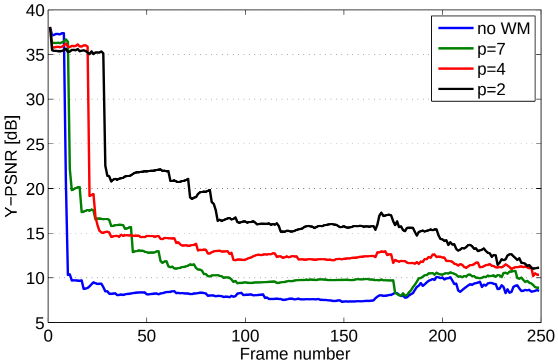

Figure 4: Y-PSNR per frame at the decoder for ”carphone” sequence compressed with QP=28 and inserted FEW for $p = 2, 4, 7$ and without watermarking. BEP $= 1 \cdot 10^{-6}$.

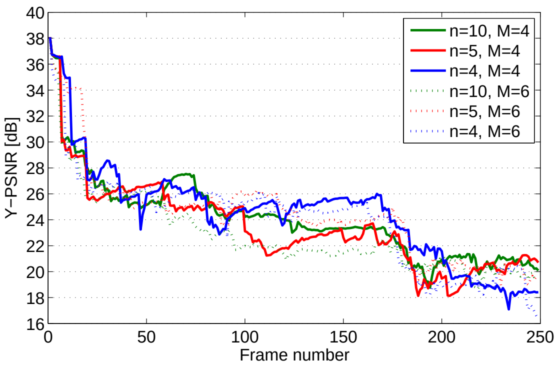

Figure 5: Y-PSNR per frame at the decoder for ”carphone” sequence compressed with QP=28 and inserted RBW for $M = 4, 6$, different n. BEP $= 1 \cdot 10^{-6}$.

forms considerably better. To compare fairly and to consider the rate increase caused by using RBW resp. rate decrease using FEW, Figure 6 shows the dependency between the size of the compressed and watermarked stream and the Y-PSNR at the decoder for BEP $= 10^{-6}$. Still, RBW brings between 2 and 15 dB gain against FEW. The gain is slightly higher for the higher error probabilities and would decrease for decreasing BEP. This gain is given by higher error detection probability of RBW. For example when encoding with QP $= 20$ we obtained error detection probability of 98.2% for BEP $= 1 \cdot 10^{-6}$, 98.3% for BEP $= 1 \cdot 10^{-5}$ and 98.6% for BEP $= 1 \cdot 10^{-4}$ with RBW ($M = 4, n = 4$). With FEW ($p = 4$) the probabilities of error detection were about 90%, similar for all three BEP values. Single parity check only results in about 49% of error detections.

5. CONCLUSIONS

In this work we investigated a fragile forced even watermarking scheme used to detect the errors in the VLC stream in H.264. We proposed an alternative approach to FEW that turned out to provide considerably better results in terms of

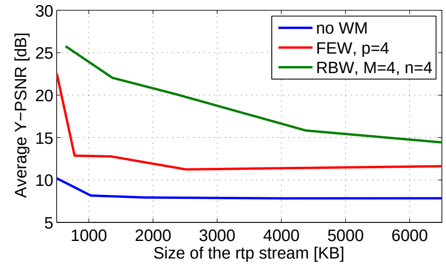

Figure 6: Comparison of FEW, RBW and packetloss concealment: Y-PSNR at the decoder over the size of the encoded and watermarked RTP stream and BEP $= 1 \cdot 10^{-6}$.

the quality at the decoder for the same rate at the encoder after applying the watermark. The proposed approach is based on the forced relation between the chosen transform domain coefficients. The solution is not unique and as the watermark finally applied, the modification of coefficients is chosen that minimizes the distortion. The choice of the extracting function and the number of coefficients to be watermarked determine the tradeoff between the probability of the detection and the distortion at the encoder. They should be chosen carefully according to the bit error probability of the transmission environment. As further work we would like to perform a more detailed investigation of the relation between the bit error probability and the optimal choice of the extracting function parameters.

REFERENCES

[1] 3GPP TSG TR 26.937 ”Transparent end-to-end packet switched streaming service (PSS); Real-time Transport Protocol (RTP) usage model,” ver 5.0.0, 2003.

[2] M. Barni, F. Bartolini, P. Bianco, ”On the performance of syntax-based error detection in H.263 video coding: a quantitative analysis,” Electronic Imaging 2000 – SPIE Conf. on Image and Video Communications, San Jose, CA, Jan. 2000.

[3] M. Chen, Y. He, R.L. Lagendijk, ”A Fragile Watermark Error Detection Scheme fot Wireless Video Communications,” IEEE Trans. on Multimedia, vol. 7, no. 2, pp. 201-211, April 2005.

[4] O. Nemethova, W. Karner, A. Al-Moghrabi, M. Rupp, ”Cross-Layer Error Detection for H.264 Video over UMTS,”, in Proc. of Wireless Pers. Mob. Comm., Aalborg, Denmark, Sep. 2005.

[5] O. Nemethova, J. Canadas, M. Rupp, ”Improved Detection for H.264 Encoded Video Sequences over Mobile Networks,” Proc. of Int. Symp. on Communication Theory and Applications, Ambleside, UK, July 2005.

[6] C. Weidmann, P. Kadlec, O. Nemethova, A. Al-Moghrabi, ”Combined Sequential Decoding and Error Concealment of H.264 Video,” Proc. of MMSP, Oct. 2004, Siena, Italy.

[7] H.264/AVC Software Coordination, ”Joint Model Software”, ver. 7.3, available in `http://iphome.hhi.de/suehring/tml/`.

[8] F. Bellifemine et all., ”Statistics analysis of the 2D-DCT coefficients of the differential signal of images,” Signal Processing: Image Communic., vol. 4, 1992.

[9] JVT, ”Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T rec. H.254 – ISO/IEC 14496-10 AVC),” March 2003, JVT-G050 available on `http://bs.hhi.de/~wiegand/JVT.html`.