

# UNEQUAL ERROR PROTECTION BASED ON FLEXIBLE MACROBLOCK ORDERING FOR ROBUST H.264/AVC VIDEO TRANSCODING<sup>1</sup>

Matteo Naccari<sup>\*</sup>, Giovanni Bressan<sup>\*</sup>, Marco Tagliasacchi<sup>\*</sup>, Fernando Pereira<sup>†</sup> and Stefano Tubaro<sup>\*</sup>

(<sup>\*</sup>)Dipartimento di Elettronica e Informazione, Politecnico di Milano  
({naccari,tagliasa,tubaro}@elet.polimi.it)

(<sup>†</sup>)Instituto Superior Técnico – Instituto de Telecomunicações  
(fp@lx.it.pt)

## ABSTRACT

This paper proposes an error resilient transcoding scheme to perform unequal error protection for H.264/AVC coded video sequences over error prone channels. In order to protect those macroblocks which impact mostly on the distortion introduced at the decoder, a slice partitioning algorithm is designed on the basis of information that is available as the output of the entropy decoder. Hence, the proposed error resilient transcoding algorithm does not require full decoding, which includes inverse transform and motion compensation, and it is indicated to work in those devices where the computational resources are scarce (routers and switching transmitter stations). The proposed method has been evaluated against a classic forward error correction scheme with equal error protection of the transmitted content. Experimental results on real video test sequences show gains of up to 3 dB with respect to equal error protection.

**Index Terms**— Flexible macroblock ordering, unequal error protection, slicing, H.264/AVC standard

## 1. INTRODUCTION

Nowadays, video coding technologies allow the delivery of digital video contents for a wide range of applications, networks and terminal devices. The most representative applications are broadcast television, video streaming, videotelephony, etc. Figure 1 depicts a general scenario for the delivery of the aforementioned applications: the original video sequence is encoded with a motion compensated predictive (MCP) encoder and the bitstream stored at a local node, or transmitted across a local network, which is characterized by a low packet loss rate (PLR). Due to the heterogeneity of receiving client networks, the node *transmitter base station* depicted in Figure 1 may perform error resilient transcoding in order to minimize the end-to-end distortion during the transmission of the video content over high packet loss rate networks such as wireless links ([1],[2]).

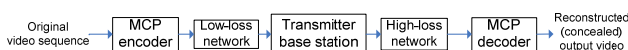


Figure 1 - Typical digital video delivery scenario

In the literature, there are several coding schemes that adopt transcoding to increase the robustness to channel loss. The error resilient transcoder proposed in [3] adds temporal and spatial resilience in the bitstream taking into account the statistics of the

video content and the channel bit error rate. Spatial resilience is achieved inserting synchronization markers whilst temporal resilience is obtained transcoding some Inter coded macroblocks into Intra coded ones. The overhead bitrate due to the added redundancy is compensated by applying a bitrate transcoder [1] in order to meet the bandwidth constraints. Also the work presented in [4] falls into the category of error resilience transcoding since it exploits channel feedback to add redundancy depending on the channel conditions. Two tools are used to provide resilience: Adaptive Intra Refresh (AIR) and Feedback Control Signaling (FCS). These tools can be used separately or combined to exploit the information coming from the feedback channel.

Although not precisely a transcoder, the work in [5] proposes a slicing partition for unequal error protection conceptually similar to our proposed algorithm. In that work, the Flexible Macroblock Ordering (FMO) error resilience tool in the H.264/AVC video coding standard [6] is exploited to group the macroblocks in one frame into different slice groups which have different degrees of importance (High, Medium and Low importance). Each slice group receives different redundancy protection depending on the assigned degree of importance. The slicing partition is performed based on the macroblock distortion. Finally, a dynamic programming algorithm is used to assign the percentage of redundancy protection to each slice group according to its degree of importance and the bandwidth constraints. The main difference between this scheme and the one proposed in this paper regards the central fact that here the original video sequence is not available to calculate the macroblock distortion, since the proposed algorithm shall work at the *Transmitter base station* node in Figure 1.

Since the computational resources at the transcoding node are typically scarce, a solution favoring low complexity has to be designed. Therefore, this paper proposes a fast homogeneous (i.e. H.264/AVC to H.264/AVC), transcoding algorithm that enhances the robustness of the output video stream by performing only entropy decoding-encoding and Forward Error Correction (FEC) based channel coding.

The remainder of this paper is organized as follows: Section 2 presents the proposed transcoding algorithm. In Section 3, the application scenarios and the experimental setup for the performance evaluation included in Section 4 are described. Finally, Section 5 concludes the paper and discusses the ongoing and future works to improve the proposed algorithm.

<sup>1</sup> The work presented was developed within VISNET II, a European Network of Excellence (<http://www.visnet-noe.org>), funded under the European Commission IST FP6 programme.

## 2. UNEQUAL ERROR PROTECTION BASED ON FLEXIBLE MACROBLOCK ORDERING

The proposed transcoding scheme is based on an adaptive slicing group partitioning algorithm, which leverages the FMO tool supported by the H.264/AVC video coding standard [6]. FMO, combined with unequal error protection (UEP), can effectively combat channel noise.

In order to perform unequal error protection, the proposed transcoder has to establish the most relevant macroblocks in each frame. Here, relevance refers to the degree of distortion introduced at the decoder if that macroblock is lost. The proposed algorithm is independent from the actual concealment strategy performed at the decoder. It only assumes that the efficiency of inter-frame concealment depends on the local motion activity and prediction residuals, whereas that of intra-frame concealment is based on the local spatial texture. Nevertheless, the experiments illustrated in this paper adopt the concealment algorithms suggested in the H.264/AVC reference software [7] that work as explained in the following. For P slices, the concealment of a 16×16 MB consists of performing motion compensation using one of the motion vectors from the surrounding 8×8 blocks. In order to identify the best motion vector in the set of candidates  $v_i$ , the lost macroblock is replaced by the one pointed by  $v_i$  in the reference frame. The

chosen (concealed) motion vector,  $\bar{v}$ , is the one that minimizes the sum of absolute differences (SAD) along the boundaries of the lost macroblock. For I slices, the concealment is performed by a weighted averaging interpolation using the samples of the surrounding, neighboring macroblocks.

In the proposed error resilient transcoder the end-to-end distortion is estimated on the basis of some information that is obtained from the decoded stream at a low computational cost, this means without decoding up to the uncompressed pixel domain. Here the compressed information used for the relevance metrics are the motion vectors and quantized prediction residuals in the transformed domain; both can be extracted performing entropy decoding only and contain valuable information related to the coded video. Using this information, the transcoder determines a partitioning of the macroblocks into two slice groups: Group **A** contains those macroblocks that, when lost, are likely to introduce a larger distortion (thus they are the most relevant), while Group **B** contains the remaining macroblocks. After this step, entropy coding is performed and the slicing group partitioning determined by the proposed algorithm leads to an unequal error protection (UEP) solution, by adaptively tuning the amount of channel coding redundancy added to each slice partition with group **A** receiving more protection.

### 2.1. Algorithm Walkthrough

In this subsection, the proposed unequal protection group algorithm is presented in detail. In the following,  $\Gamma$  and  $\Omega$  denote the degree of relevance of macroblocks belonging to P and I slices, respectively. The steps involved in the proposed unequal protection transcoding algorithm can be summarized as follows:

#### For each frame **F** do

1. Entropy decoding of the coded bitstream to get motion vectors and prediction residuals in the quantized transformed domain for each macroblock.
2. Calculate the relevance degree of each macroblock MB belonging to frame **F**:
  - 2.1. If the MB belongs to a I slice then calculate relevance degree using the  $\Omega$  metric (see subsection 2.2).

- 2.2. Else if the MB belongs to a P slice then calculate the relevance degree using the  $\Gamma$  metric (see subsection 2.3).
3. Perform slice group partitioning based on the relevance degree determined in step 2) (see subsection 2.4).
4. Perform the entropy re-encoding according to the slicing group determined.
5. Apply channel coding redundancy over the coded bitstream (see subsection 2.5).

End do

### 2.2. Protection Metric for I Slices

Usually, concealment algorithms for intra coded macroblock try to recover lost data performing spatial interpolation between the neighbouring pixels, therefore producing a smooth version of the MB being concealed. Thus, if the lost macroblock contains detailed texture, the distortion introduced at the decoder is expected to be high. Based on this consideration, let  $\mathbf{M}$  denote the current macroblock and  $\Pi_i$  the  $i$ -th ( $i=1..16$ ) 4×4 partition belonging to  $\mathbf{M}$ . For each partition  $\Pi_i$  the proposed  $\Omega$  metric algorithm uses three frequency domain features based on the quantized prediction residuals in the transformed domain  $R_{j,k}$  ( $j=1:4, k=1:4$ ) belonging to  $\Pi_i$ , which are defined as:

$$f_1^i = \sum_{k=2:4} R_{1,k} \quad f_2^i = \sum_{j=2:4} R_{j,1} \quad f_3^i = \sum_{j=2:4} R_{j,j} \quad (1)$$

The  $\Omega$  degree of relevance for  $\mathbf{M}$  is then given as:

$$\Omega = \frac{1}{16} \sum_{i=1:16} (f_1^i + f_2^i + f_3^i) \quad (2)$$

The higher it is  $\Omega$  the richer is the MB in terms of frequency content and thus the higher its impact on the distortion if lost.

### 2.3. Protection Metric for P Slices

Figure 2 shows the topology used to compute the  $\Gamma$  metric for the current macroblock  $\mathbf{M}$  in P slices. Let  $\bar{V}_M$  denote the average motion vector between all motion vectors belonging to  $\mathbf{M}$ ,  $e_M$  the energy of the quantized prediction residuals in the transformed domain, and, finally,  $V_M^{8 \times 8}(i)$  the motion vector of the  $i$ -th 8×8 block that surrounds  $\mathbf{M}$ ; if this block is further split into 4×4 pixel sub-blocks, then  $V_M^{8 \times 8}(i)$  is computed as the arithmetic average over all four motion vectors belonging to it. The  $\Gamma$  degree of relevance for  $\mathbf{M}$  is defined as:

$$\Gamma = \frac{\alpha}{8} \cdot \sum_{i=1}^8 |\bar{V}_M - V_M^{8 \times 8}(i)| + \beta \cdot e_M \quad (3)$$

where  $\alpha$  and  $\beta$  are appropriate normalizing constants, whose values are adapted on a frame-by-frame basis, in such a way that the ratio:

$$\frac{\beta}{\alpha} = \frac{\max_M \left[ \frac{1}{8} \cdot \sum_{i=1}^8 |\bar{V}_M - V_M^{8 \times 8}(i)| \right]}{\max_M [e_M]} \quad (4)$$

is kept constant. The first terms of right hand side (r.h.s) in equation (3) measures the degree of correlation between the average motion vector of  $\mathbf{M}$  and the motion vectors of the surrounding 8×8 blocks. The rationale behind this metric is that the lower the correlation, the higher will be the distortion after performing motion-compensated temporal concealment at the decoder. Furthermore, it can be observed that the error concealment algorithm cannot recover the prediction residuals

even if the motion vector  $\bar{V}_M$  is highly correlated with those of the neighboring blocks. One way to take this into account is to evaluate which is the expected distortion at the decoder if the prediction residuals are fully lost. The proposed slice grouping algorithm uses the energy of the prediction residuals in the transformed domain (the second terms in the r.h.s of equation (3)) to quantify the distortion at the decoder side due to the lack of these residuals.

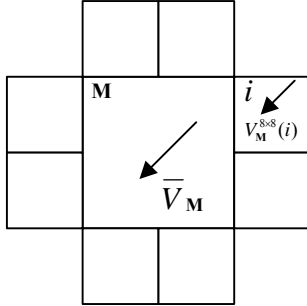


Figure 2: Chosen topology to calculate the  $\Gamma$  degree of significance for the macroblock  $M$

## 2.4 Slice Grouping

Once the significance degree  $\Gamma$  ( $\Omega$ ) of each macroblock  $M$ , belonging to each frame  $F$  in the coded sequence has been determined, the proposed unequal error protection transcoding algorithm sorts the macroblocks in descending order of  $\Gamma$  ( $\Omega$ ) and assigns the first  $m$  macroblocks to slice group **A**, the one that will get more protection in terms of channel coding.

The value of  $m$  has been determined empirically during the simulation test with the goal of minimizing the end-to-end distortion. In the conversational (broadcast) scenario, the value of  $m$  has been set equal to 20% (40%) of total number of macroblocks.

## 2.5 Channel Coding

Unequal error protection is achieved introducing more channel coding redundancy in the slices that belong to group **A**. In the proposed transcoder, the total amount of redundancy  $R$  is spread over the slices of group **A** and **B**; more precisely, group **A** received  $R_A$  redundancy while group **B** received  $R_B$  redundancy, with  $R_A + R_B \leq R$ . This allocation may be manually tuned or an algorithm, like the one proposed in [5], may automatically determine  $R_A$  and  $R_B$ .

The creation of slice groups introduces a rate overhead due to the MacroBlock Allocation Map (MBAmP) syntax element. Experimental results demonstrate that the overhead is about +16% with respect to the case where each frame fits in a single slice. This compares favourably with the disperse slicing partitioning, that introduces a +19% rate overhead on average.

## 3. TESTING SCENARIOS AND EXPERIMENTAL SETUP

The proposed unequal error protection transcoding algorithm has been tested for two typical application scenarios in the video coding field: broadcasting and conversational applications.

### 3.1 Source Coding Conditions

This subsection illustrates the video sequences and the coding conditions used for the aforementioned testing scenarios. Both sequences involved in the experimental scenarios were coded

with the H.264/AVC encoder software [8] in the Baseline profile.

In the broadcasting scenario, the Football video sequence has been coded at 525 SIF spatial resolution, 30 frames/s, constant quality QP = 24 (average bitrate  $R = 1,8$  Mbit/s), average PSNR of 38 dB and IPPP GOP type with an intra frame every 12 frames. For the conversational scenario, the Mother and Daughter video sequence has been coded at QCIF spatial resolution, 15 frames/sec, constant quality QP = 30 (average bitrate  $R = 28$  kbit/s), average PSNR of 35 dB and GOP type IPP with length of 300.

### 3.2 Channel Coding Conditions

This subsection details the channel conditions and the allocated redundancies to test the proposed algorithm. In both the scenarios, the packetization adopted is the Real-time Transfer Protocol (RTP) [9] with each packet corresponding to a coded slice. Following the guidelines in [9], the size of each packet is 64 bytes for the conversational scenario and 256 bytes for the broadcasting one; the packet loss rates (PLR) studied were 3,5,10 and 20 (%) and the respective loss patterns were generated according to [10].

Forward error correction is performed by means of  $(n, k)$  Reed-Solomon codes. The number of data slices, i.e.  $k$ , varies on a frame-by-frame basis, while the number of redundant slices  $n - k$  is tuned to match the desired channel rate  $k/n$ , which is set to be approximately equal to 0.86 (i.e. 15% redundancy) and 0.77 (i.e. 30% redundancy) for the conversational and broadcasting scenarios, respectively.

For the proposed slice group partitioning, the redundancy rates  $R_A$  and  $R_B$  (in percentage) were  $R_A = 10\%$ ,  $R_B = 5\%$  for the conversational scenario and  $R_A = 20\%$ ,  $R_B = 10\%$  for the broadcasting scenario.

## 4. PERFORMANCE EVALUATION

To evaluate the proposed unequal error protection transcoding, comparisons in terms of average PSNR were made with an Equal Error Protection (EEP) scheme, where the allocated redundancy is equally distributed over all the coded video sequence. The slicing partitioning for the EEP scheme is dispersed [6] with two slice groups. Both the UEP and the EEP schemes use the H.264/AVC JM11 reference decoder software [8] with the concealment of the I slices modified as follows: the MBs of the I slices which belong to the first frame are concealed as described in [7], whilst those MBs which belong to I slices in further frames are concealed with the spatially corresponding macroblock from the previous frame, i.e. zero-motion temporal concealment. This modification has been introduced because of the poor performance in terms of PSNR of the spatial concealment algorithm in [7], which is often outperformed by a simple zero-motion temporal concealment strategy.

Figure 3 and Figure 4 show the average PSNR for the broadcasting and conversational (respectively) scenarios. For low PLRs, the EEP technique performs better than the proposed UEP technique; however, as the PLR increases to percentages typical of wireless links, the proposed UEP algorithm performs better achieving a gain (with PLR = 20%) of 3,85 dB for the broadcasting scenario and, roughly, 3 dB for the conversational one.

The better performance achieved by the EEP scheme for low PLRs can be explained by considering the increased correcting power due to larger block lengths, which can be obtained by channel coding each frame as a whole, instead of a subset of it.

Figure 5 shows the average PSNR achieved by the proposed UEP algorithm in the conversational scenario with different

bitrates and PLR = 10%, again the proposed transcoding algorithm performs better than the EEP one; in particular, as the bitrate increases the proposed slicing partition works better due to presence of a larger number of non-zero prediction residuals that are retained after the quantization process at the encoder side. Finally, Figure 6 compares the proposed UEP algorithm with another UEP algorithm using a slicing partitioning where the macroblocks are ranked on the basis of their real distortion (assuming that the original video sequence is available to compute it); this type of slicing partitioning represents an oracle-based optimum solution that can only be approached with the proposed approach. Figure 6 shows that the proposed algorithm achieves a PSNR performance that is nearly the same as the optimum partitioning (also used in [5]).

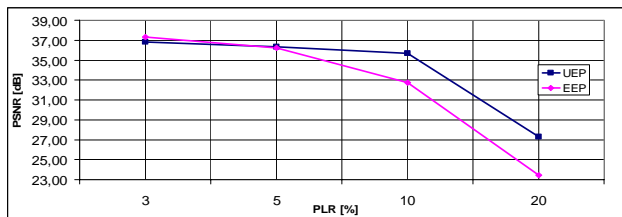


Figure 3: Average PSNR for the broadcasting scenario

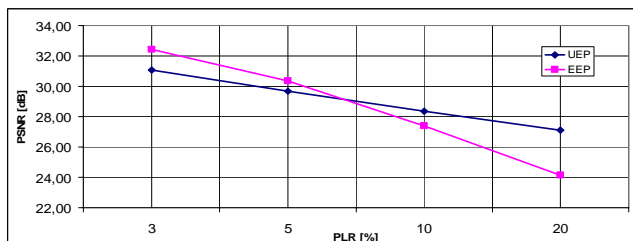


Figure 4: Average PSNR for the conversational scenario

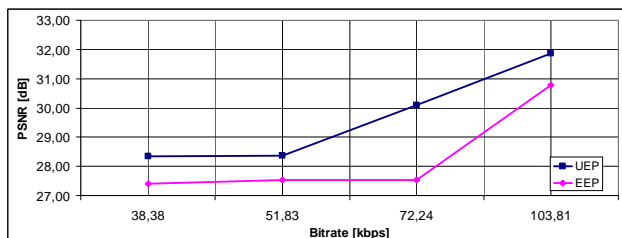


Figure 5: Average PSNR for the conversational scenario with varying bitrate and PLR = 10%

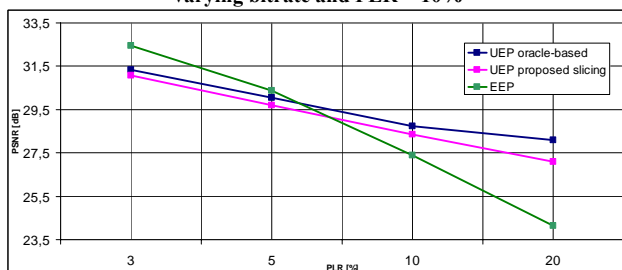


Figure 6: Average PSNR for the conversational scenario with varying PLR

## 5. CONCLUSIONS AND FUTURE WORK

In this paper an error resilient transcoder has been proposed. The slice partitioning algorithm establishes the relevance degree of each macroblock based on information available after entropy decoding. Experimental results have shown not only that the proposed transcoder outperforms a classical EEP scheme, especially at high packet loss rates, but also that the outlined

slice partitioning performs nearly as an oracle-based one, which assumes the knowledge of the original frames.

Currently the redundancy allocation for the slice group **A** and **B** is tuned manually. Future research will address the problem of allocating automatically  $R_A$  and  $R_B$  in order to minimize the end to end distortion. Since the original frames are unavailable in a transcoding scenario, the latter needs to be somehow estimated based on available data.

## REFERENCES

- [1] A. Vetro, C. Christopoulos and H. Sun, "An overview of video transcoding architectures and techniques", *IEEE Signal Processing Magazine*, vol. 20, n° 2, pp 18-29, March 2003.
- [2] A. Vetro, J. Xin and H. Sun, "Error resilience video transcoding for wireless communications", Technical Report MERL Laboratories, August 2005.
- [3] G. de los Reyes, A.R. Reibman, S.-F. Chang, and J.C.-I. Chuang, "Error-resilience transcoding for video over wireless channels", *IEEE J. of Selected Areas Commun.*, vol. 18, n° 6, pp. 1063-1074, June 2000.
- [4] S. Dogan, A. Cellatoglu, M. Uyguroglu, A.H. Sadka, and A.M. Kondo, "Error-resilience video transcoding for robust inter-network communications using GPRS", *IEEE Trans. On Circuits and System for Video Technologies*, vol. 12, pp. 453-464, June 2002.
- [5] N. Thomos, S. Argyropoulos, N.V. Buolgouris and M. G. Strintzis, "Robust transmission of H.264/AVC video using adaptive slice grouping and unequal error protection", ICME 2006, Toronto, Canada, July 2006.
- [6] Information Technology – *Coding of audio-visual objects – Part 10: advanced video coding*. Final Draft International Standard, ISO-IEC FDIS 14 496-10, 2003.
- [7] G. Sullivan, T. Wiegand and K. P. Lim, "Joint model reference encoding methods and decoding concealment methods", JVT-I049, September 2003.
- [8] H.264/AVC Reference software version JM11 downloadable at [http://iphome.hhi.de/suehring/tml/download/old\\_jm/](http://iphome.hhi.de/suehring/tml/download/old_jm/).
- [9] S. Wenger, "H.264/AVC over IP", *IEEE Transactions on Circuits and System for Video Technologies*, vol. 13, n° 7, pp. 645-656, July 2003.
- [10] S. Wenger, "Error patterns for Internet experiments", JVT-Q15-I-16r1, October 1999.