

JOINT OPTIMIZATION OF SOURCE-CHANNEL VIDEO CODING USING THE H.264 ENCODER AND FEC CODES

Simone Milani[†], Gian Antonio Mian[†], and Luca Celetto[‡]

[†] Dept. of Information Engineering, University of Padova
via Gradenigo 6/B, 35012 Padova, Italy
email: {simone.milani, mian}@dei.unipd.it

[‡] STMicroelectronics,
via C. Olivetti 2, 20041 Agrate (MI), Italy.
email: luca.celetto@st.com

ABSTRACT

One of the most challenging drawbacks of video transmission over mobile lossy channels is the degradation of the reconstructed video sequence at the decoder. In fact, the high percentage of lost packets, as well as the intensive use of prediction to obtain a high compression ratio, greatly affects the visual quality of the reconstructed sequence. As a matter of fact, it is necessary to introduce some redundant data in order to increase the robustness of the coded bitstream. A possible solution is given by filling a matrix structure with RTP packets and applying a FEC code on its columns. However, the matrix size and the chosen type of FEC code affect the performance of the coding system. The paper proposes different optimization algorithms that adapt the channel coder configuration according to the input signal statistics in order to maximize the objective (and perceptual) quality of the decoded sequence.

1. INTRODUCTION

During the last years, the IT world has assisted the development of more and more efficient video coding standards. The requirements of video communications services over mobile channels have led the design of new coding schemes that are intended to obtain good perceptual quality at low bit rates.

The recent standard H.264/AVC [1] follows this trend and provides better performance than the previous hybrid motion-compensated video coding schemes. In fact, each macroblock is predicted using either a temporal motion compensation or a spatial interpolation from the borderline pixels. Then, the residual information is coded by a transform-based coding scheme. It is possible to control the compression gain and the introduced distortion by varying the quantization parameter QP.

Despite the good performance in terms of compression gain, the wide use of prediction constitutes a weak point whenever the coded stream is transmitted over an unreliable channel. The loss of a part of a frame implies the corruption of the displayed image due to the lack of a reference for motion compensation. In these cases, we need to implement some error concealment techniques at the decoder (see [2]) in order to estimate the lost state of the frame buffer. These techniques prove to be inefficient whenever the movement of the sequence is too complex or the number of packet losses is high. Therefore, we need to increase the performance of the error concealing decoder including some redundancy packets, retransmitting part of the lost information, or coding in Intra mode some macroblocks in order to refresh the frame buffer state.

In our work, we adopted the first solution implementing different FEC-based channel coding schemes. In case some

packets are lost because of channel errors, excessive delays or network congestions, the lost information can be recovered thanks to the additional bytes whenever the number of lost packets is lower than a given threshold. Over this limit, the channel decoder can not perform any reconstruction, and the final visual quality of the displayed sequence only depends on the adopted error concealment algorithm. This all-or-nothing behavior may result in a waste of bandwidth in case the correcting capability of the adopted code does not suit the loss probability.

Since the available bandwidth is limited, the channel transmission capability has to be shared between the source coder and the channel coder in order to maximize the perceptual quality of the reconstructed sequence. The paper presents an effective and efficient rate allocation strategy between a H.264 coder and a matrix-based FEC channel coder.

Section 2 presents different strategies to include the source RTP packets in the matrix of a matrix-based FEC channel coder. Section 3 discusses how to allocate the amount of FEC packets in order to maximize the video quality at the decoder. Section 4 presents different joint source-channel rate control algorithms that optimize matrix dimensions. Then, Section 5 compares the performances of the proposed algorithms.

2. SOME MATRIX-BASED FEC CODING SCHEMES

In order to increase the robustness of the coded bit stream, some FEC coding schemes create new redundancy packets from video source RTP packets (e.g. [3]). This solution allows the channel decoder to recover the lost information whenever the number of missing packets is lower than a given number.

An efficient coding scheme inserts the video RTP packets into a matrix, and computes some additional columns using different FEC codes, like the Reed-Solomon (RS) codes. The information in the matrix is sent across the network, and the redundancy bytes are packetized in a column-wise order according to the size of the Packet Data Unit (PDU). An extra payload is added to these packets in order to make them compliant with the RTP format (similarly to [3]). This scheme (denoted in the following as *RFC2733-like*) generalizes the RFC2733 scheme and provides an improved recovering efficiency since an $RS(N, K)$ code allows the channel decoder to recover up to $N - K$ lost columns. To reduce the percentage of FEC bytes sent across the network, it is possible to include more than one packet per column wrapping the exceeding bytes on the following columns (see [4] and Fig. 1).

In case of losses, the channel decoder has to wait the matrix to be filled before recovering the lost information. There-

fore, the current frame is displayed after a time delay that depends on the dimension of the matrix.

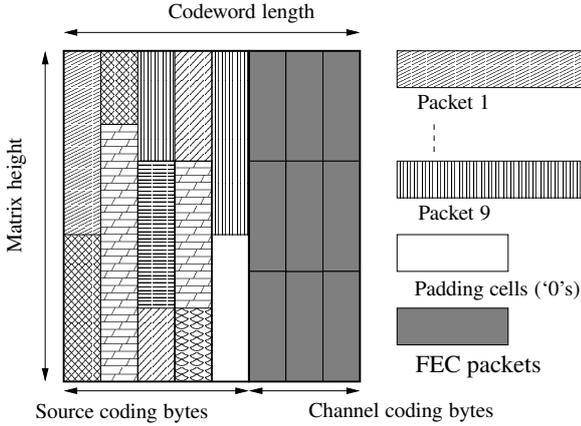


Figure 1: A matrix-based FEC coding scheme with packet wrapping.

Whenever the network is affected by bursts of losses, the video packets can be interleaved by adopting a higher matrix. As a drawback, we need to include more packets increasing the delay of the playout. In case of real-time applications, we may reduce the number of matrix columns in order to limit the delay. This approach allocates s columns for the video packets and C columns for the FEC packets, where C is computed according to the default channel code rate. The number of columns is set according to the length of RTP video units in order to avoid the wrapping of packets. In fact, data recovering can be jeopardized in case the longest packets get lost. This algorithm (denoted as $RS(C+s, s)$ with $C/s=const.$) provides better performances than the *RFC2733-like* scheme as it will be shown in Fig. 2 of Section 5 (circled line) for sequence *foreman*.

The choice of the matrix dimensions is critical. From one side, if we choose a great number of rows, packets will not be wrapped too many times and they will be efficiently protected by a smaller and less computationally intensive RS code, at the price of a higher number of padding bytes at the end of the matrix. From the other side, a low number of rows will require a minimal padding, but when a long packet is lost, several columns will be missing, requiring a longer RS code.

3. ADAPTATION OF THE CHANNEL CODER TO THE VIDEO SIGNAL

The previous section has shown how the effectiveness of the recovering operation is influenced by the relation between the size of the matrix and the characteristics of the RTP packet stream. The recovering performance can be increased considering the characteristics of the video signal.

Whenever the input video sequence presents some hardly-predictable information (such as a scene change or a non-translational motion), we need to increase the amount of redundancy bytes as the error concealing decoder can not estimate the missing elements from the previous frames. Moreover, since the current corrupted picture will be used as temporal reference for the future frames, the loss may affect more than one picture. In our simulations, we tried to adapt

the FEC code to each frame in order to increase the performance of the scheme. To this purpose, we adopted the activity of the frame as a measure of the correlation between adjacent frames assuming that a low-activity frame can be easily estimated from the previous ones by the error concealment algorithm.

While coding the video signal, the algorithm estimates the average value of the activity avg_act_i and its variance var_act_i , $i = I, P, B$, for each frame type i . After H.264/AVC has coded the n -th picture, the matrix-based channel coder adopts a $RS(K+C_n, K)$ code, where the parameter K is determined depending on the H.264/AVC RTP packets as Section 2 reports ($RS(s+C)$ with $s/C = const.$), and C_n is equal to

$$C_n = C_{default} + \begin{cases} 2 & \text{if } act > avg_act_i + \sqrt{var_act_i} \\ 1 & \text{if } act > avg_act_i + 0.5 \sqrt{var_act_i} \\ -1 & \text{if } act > avg_act_i - 0.5 \sqrt{var_act_i} \\ -2 & \text{if } act > avg_act_i - \sqrt{var_act_i} \end{cases} \quad (1)$$

provided that the final value of C_n is non-negative. In (1) $C_{default}$ is the default number of redundancy bytes for the coder while act is the activity value for the current frame. Fig. 2 of Section 5 demonstrates the effectiveness of the activity-based adaptation rule. The reported graph shows an increment of the quality of the reconstructed sequence since each frame is protected according to its importance.

In order to find a parameter that accurately characterizes both the image statistics and the produced bit rate, we adapted the FEC according to the variations of the percentage ρ of zero DCT coefficients, called “zeros” (see [5,6]). In fact, an increase in the complexity of the residual signal is characterized by a reduction of the percentage of zeros for a given QP. Therefore, whenever the percentage of zeros decreases, an increment of the recovering capability enhances the probability of restoring some of the most important information in the sequence. We ran different simulations adapting the code according to the rule

$$C_n = C_{default} + \begin{cases} 4 & \text{if } \rho > avg_p_i + 0.04 \\ 3 & \text{if } \rho > avg_p_i + 0.03 \\ 2 & \text{if } \rho > avg_p_i + 0.02 \\ 1 & \text{if } \rho > avg_p_i + 0.01 \\ -1 & \text{if } \rho > avg_p_i - 0.01 \\ -2 & \text{if } \rho > avg_p_i - 0.02 \\ -3 & \text{if } \rho > avg_p_i - 0.03 \\ -4 & \text{if } \rho > avg_p_i - 0.04. \end{cases} \quad (2)$$

where avg_p_i is the average percentage of zeros for i -type frames ($i = I, P, B$).

Fig. 2 reports the corresponding results. The graph shows that the ρ -based algorithm has approximately the same performance as the activity-based algorithm with the advantage that the percentage of “zeros” can be reused to model the bit rate.

4. JOINT SOURCE-CHANNEL RATE CONTROL

The experimental results reported in Fig. 2 show that the adoption of a matrix-based FEC coder is an efficient solution for protecting the data transmitted over an unreliable channel. However, the reported plots provide evidence for the need of optimizing the matrix size and the adopted code according

to the input video signal, the channel characteristics, and the available bandwidth.

In order to keep the coded bit stream within given bandwidth constraints, each frame is given a number of bits T_n that is computed according to

$$T_n = \frac{G_{n,i}}{K_{I,P} \cdot K_{P,B} \cdot N_I + K_{P,B} \cdot N_P + N_B} \quad (3)$$

where $G_{n,i}$ is the number of bits remaining in the i -th GOP after coding the n -th frame and N_t , $t = I, P, B$, is the number of not-yet-coded t -type frames that still remain in the current GOP. The quantity K_{t_1,t_2} is the complexity ratio

$$K_{t_1,t_2} = \frac{\hat{X}_{t_1}}{\hat{X}_{t_2}} \quad t_1, t_2 = I, B, P. \quad (4)$$

The average complexity \hat{X}_t (see [6]) is updated after coding each frame via the equation

$$\hat{X}_t \leftarrow \hat{X}_t \cdot 0.9 + 0.1 \cdot X_t \quad t = I, P, B. \quad (5)$$

where

$$X_t = (B^S + B^C) \cdot 2^{QP/6} \quad (6)$$

quantifies the complexity of the current frame with B^S the number of bits produced by the source coder and B^C the number of bits produced by the channel coder. The parameter QP is the average quantization parameter for the current frame.

The target number of bits T_n is partitioned into two amounts of bits such that

$$T_n = T_n^S + T_n^C \quad \text{with} \quad T_n^S = T_n \cdot (1 - r_n) \quad (7)$$

where r_n denotes the amount of bandwidth that is allocated to the channel coder. The target T_n^S is the number of bits available for the H.264 to code the current frame, while T_n^C is the number of bits that can be used to add redundancy information to protect the bit stream.

The rate allocation algorithm initially sets the percentage r_n to a default value \bar{r} , and modifies it according to the signal characteristics in order to improve the performance of the scheme. Let m be the index of the frame that precedes the current frame n and has the same coding type. In case $r_m > \bar{r}$, the coding rate for the current frame is modified as

$$r_n = \bar{r} - (r_m - \bar{r}). \quad (8)$$

In case $r_n \leq 0$, no FEC packets are sent. This adaptation avoids allocating too many bits of the available bandwidth to the channel coder reducing the number of available bits for the source coder and, consequently, the quality of the reconstructed sequence.

According to the target value T_n^S and the signal characteristic, it is possible to estimate a target percentage ρ_n of null quantized DCT coefficients for the current frame [5, 6]. The parameter ρ_n characterizes the amount of information brought by the coded coefficients and, as a consequence, denotes which frames or slices can be hardly predicted from the neighboring ones in case they get lost. The lower the percentage of “zeros”, the more important is the information brought by coefficients. Therefore, it is possible to improve the performance of the coding scheme modifying the adopted

channel code by increasing the number of columns C_n whenever the parameter ρ_n for the current frame is lower than its average value. First, the algorithm computes a default number of FEC columns C

$$C_{default} = \lfloor r_n \cdot C_{tot} + 0.5 \rfloor \quad (9)$$

where C_{tot} is the total number of columns of the matrix and r_n is the computed channel code rate. Then, the number of FEC columns is adjusted to the final value C_n according to equation (2).

In this way it is possible to increase the protection of those part that present some important changes in the video sequence.

The available number of bits for the current GOP is updated through the equation

$$G_{n+1,i} = G_{n,i} - B_n^S - B_n^C. \quad (10)$$

5. EXPERIMENTAL RESULTS

In order to evaluate the efficiency of the different coding solutions, we varied the length of channel code and the error generation process in order to test the solutions under different conditions. In case some RTP packet is still missing, the adopted decoder resorts to error concealment interpolating the lost part of the image from the neighboring pixels [2]. In case of an Inter macroblock, the motion information is estimated from the neighboring motion vectors.

In our investigation, we considered a packet-switched transmission on an AWGN radio channel with $E_b/N_r = 4\text{dB}$. The length of the frame was 200 bytes and the adopted transmission scheme was a QPSK modulation with a convolutional code ($rate = 1/2$ and $memory = 5$). The measured BER is $0.2 \cdot 10^{-3}$, which results in a loss probability $P_{SDU\ loss}$ for a Single Data Unit (SDU) equal to 0.10%.

In our first investigation, we analyzed the performance of different matrix sizing algorithms for video packet stream coded at constant visual quality (fixed Quantization Parameter QP) in slices of 11 macroblocks. The GOP is 15 frames long with structure IPBB. Figure 2 reports the performance in terms of objective quality of the reconstructed sequences for the different algorithms and sequence `foreman`. It is possible to notice that the adaptive approach (RS(x+y,x) with $x/y=\text{cost}$.) requires a lower amount of bandwidth with respect to the fixed size approach (RFC2733-like).

We tried different matrix configurations by increasing the number of redundancy bytes and the length of the matrix. Then, we optimized the adopted code by varying the number of redundancy columns in the matrix according to the activity of the coded frame. The corresponding graph (marked with crosses) in Fig. 2 shows an increment of the visual quality of the reconstructed sequence since each frame is protected according to its importance in the decoding process. As a matter of fact, the algorithm avoids using a lot of useless FEC packets whenever the sequence characteristics can be easily estimated by the error concealer. In addition, it is possible to notice that the performance of the ρ -adaptive algorithm (plot marked with squares) is similar to the one of the activity-adaptive approach proving that the activity parameter can be substituted by ρ without affecting the correcting performance of the code. The percentage of “zeros” allows

Target Bit Rate (kbit/s)	Actual Bit Rate (kbit/s)	Default Channel Code Rate \bar{r}	Effective Channel Code Rate	Lost RTP packets (%)	Lost RTP packet after FEC correction (%)	Average PSNR (dB)
356	355.67/357.29	0.22	0.40/0.57	13.52/13.68	5.93/2.62	29.19/32.00
400	400.44/400.24	0.22	0.60/0.57	13.51/13.53	3.07/2.48	28.87/32.64
450	471.35/450.19	0.22	0.41/0.58	13.44/13.95	6.88/3.36	29.04/33.12
500	511.85/507.89	0.22	0.54/0.56	14.34/14.19	3.48/3.73	29.60/33.41
550	552.26/562.14	0.22	0.62/0.57	14.29/14.76	3.33/4.85	28.98/32.22

Table 1: Comparison of fixed rate (left) and ρ -adaptive (right) algorithms for the sequence `news`

Target Bit Rate (kbit/s)	Actual Bit Rate (kbit/s)	Default Channel Code Rate \bar{r}	Effective Channel Code Rate	Lost RTP packets (%)	Lost RTP packet after FEC correction (%)	Average PSNR (dB)
356	359.03/360.94	0.44	0.29/0.77	12.91/13.10	7.04/0.34	22.29/28.44
400	404.24/404.54	0.44	0.60/0.77	13.59/13.09	5.11/0.45	25.56/29.22
450	453.07/453.92	0.44	0.61/0.77	13.51/13.19	3.07/0.29	27.05/30.51
500	502.13/503.57	0.44	0.62/0.77	13.59/13.19	2.48/0.42	27.24/32.04
550	551.93/553.35	0.44	0.62/0.78	13.77/13.43	2.70/0.50	27.20/30.78

Table 2: Comparison of fixed rate (left) and ρ -adaptive (right) algorithms for the sequence `foreman`

the channel coding scheme to obtain the same results of the activity-based algorithm with the advantage that the parameter ρ can be reused to model the bit rate, as reported in [5, 6].

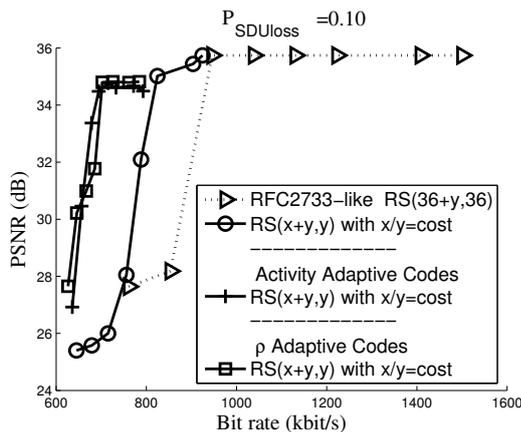


Figure 2: A comparison between different matrix sizing algorithms (coding results for the sequence `foreman` with QP=30)

As a consequence, we analyzed the performance of the joint source-channel rate control reported in Section 4 with respect to applying a fixed dimension code. Tables 1 and 2 report the experimental results obtained for the sequences `news` and `foreman` coded with GOP IPBB at 30 frame/s.

It can be appreciated that the ρ -adaptive algorithm is able to change the coding rate according to the input sequence, providing a higher perceptual quality with respect to the fixed rate approach. In fact, the algorithm is able to partition the available bandwidth in appropriate manner increasing the code rate whenever the input signal is not correlated with the previous data.

6. CONCLUSIONS

In the paper we presented different FEC-based techniques for the robust transmission of coded video sequences. By including the video RTP packets produced by the H.264/MPEG-4 AVC coder in a matrix, it is possible to optimize the adopted

FEC code reducing the redundancy bit rate. Our experimental results show that a further improvement can be achieved by adapting the code to the input signal. It was shown that such an adaptation can be performed using the activity parameter to characterize the statistics of the video signal. In alternative, the adaptation can be based on the percentage of null DCT coefficients: this approach permits to model both the signal statistics and the bit rate produced by the H.264 coder and results in an effective joint source-channel rate control. Simulations provide experimental evidence for the noticeable improvements in terms of objective visual quality of the reconstructed sequence.

REFERENCES

- [1] Ian E. G. Richardson, *H.264 and MPEG-4 video compression: video coding for the next-generation multimedia*, Wiley, Chichester, 2003.
- [2] G. Gennari, G. A. Mian, and L. Celetto, "A robust H.264 decoder with error concealment capabilities," in *Proc. of the 12th European Signal Processing Conference (EUSIPCO 2004)*, Wien, Austria, Sept. 6–10 2004, pp. 649–652.
- [3] J. Rosenberg and H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction (RFC2733)," in *Network Working Group*, Dec. 1999.
- [4] 3GPP TSG-SA4 Siemens Group, "Matrix approach vs. packet approach for MBMS application layer FEC," in *3GPP TSG-SA4 Meeting TSG-SA4 # 30*, Malaga, Spain, Feb. 23–27 2004.
- [5] Z. He and S. K. Mitra, "A unified rate-distortion analysis framework for transform coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 12, pp. 1221–1236, Dec. 2001.
- [6] S. Milani, L. Celetto, and G.A. Mian, "A rate control algorithm for the H.264 encoder," in *Proc. of the Sixth Baiona Workshop on Signal Processing in Communications*, Baiona, Spain, Sept. 8–10, 2003.