Slice Sorting for Unequal Loss Protection of Video Streams

Enrico Baccaglini, *Member, IEEE*, Tammam Tillo, *Member, IEEE*, and Gabriella Olmo, *Senior Member, IEEE*

***Abstract*—In this letter, we propose a novel unequal loss protection scheme, which allocates FEC codes to video slices according to their impact on the GOP distortion. This is evaluated taking the concealment procedure and the drift effect into account. Simulation results show that the proposed algorithm outperforms state-of-the-art approaches, reducing the gap with the error-free performance curve. Moreover, the complexity of the additional stage required to pilot the protection allocation stage is negligible with respect to traditional ULP schemes.**

***Index Terms*—H.264/AVC, unequal loss protection, video streaming.**

I. INTRODUCTION

THE increasing number of wideband home connections, jointly with the technological advances in video coding, are making peer-to-peer (P2P) video streaming more and more popular [1]. In P2P streaming, all peers provide resources in terms of upload bandwidth and storage space, and the distributed nature of the network allows one to increase robustness by replicating data over multiple peers. On the other hand, the overlay network should guarantee reliability of the connections and a constant flow of data, as well as low startup latency [2], crucial for streaming of live events. In this context, a main challenge to be coped with is the typically non-negligible packet loss rate, mostly due to the unreliability of peers. This may cause the loss of some macroblocks (MBs), slices, or even of whole frames at the decoder. Furthermore, distortion due to packet losses not only corrupts a single frame but also propagates to subsequent frames due to the predictive encoding mechanism. This effect is referred to as drift [3].

Retransmission mechanisms cannot be typically afforded due to the constraints of real-time content delivery. As a consequence, one should rely on source coding or joint source-channel coding mechanisms to insert a proper amount of redundant information able to cope with information losses. Error resilient coding schemes (such as flexible MB ordering and Intra MB refresh for the H.264/AVC standard [4]) may be useful to mitigate the effect of channel errors, but they usually increase computational complexity and reduce compression efficiency; moreover, they are seldom addressed as standalone methods but possibly associated to error correction mechanism to improve the overall performance. Error concealment is indeed applied at the decoder side, and it can mitigate the effect of lost information; sophisticated spatiotemporal interpolation of missing slices have been proposed, e.g., in [5].

Unequal loss protection (ULP) algorithms [6] recognize the fact that multimedia data are not equally important. They assign unequal amounts of forward error correcting (FEC) protection so as to obtain different protection strength according to the importance of various portions of the data. The main objective of ULP is to maximize the expected quality of the received video stream, and this is achieved by using different optimization techniques. In [7], ULP is applied to motion-compensated H.263 video by reordering the bitstream so as to make it embedded. In [8], stronger FEC codes are allocated to the first frames of a group of pictures (GOP), to account for their larger impact on the error propagation. In [9], the error resilience features of H.264/AVC are exploited together with Reed–Solomon (RS) codes to enhance the protection of the stream. The optimal classification of MBs into slice groups and the optimal rate allocation are achieved by iteration.

In this letter, we propose an alternative ULP algorithm named *slice sorting by relevance* (SSR). Video slices are first sorted, depending on their contribution to the GOP distortion. Then, slices are protected with different FEC codes, according to their importance and the packet loss rate. This letter is organized as follows. In Section II, we introduce the SSR allocation. In Section III, we provide some experimental results, and in Section IV, we draw some conclusions and discuss future developments.

II. SSR ALGORITHM: ENCODER OPERATION

Fig. 1 represents the block diagram of SSR, encompassing a standard H.264/AVC encoder as well as four additional blocks (dotted box). In the *decoder emulation block*, the decoding operations are emulated (including the adopted concealment strategy), and the contribution of each possibly lost slice to the GOP distortion is evaluated. In the *RRD function generator*, slices are ranked from the most to the least important, according to a suitable criterion, while in the *redundancy allocation* unequal FEC code allocation is performed, so that the most important slices are the most protected, hence likely to be successfully decoded.

In order to perform a *slice classification*, the impact of each slice on the GOP distortion must be evaluated. To this end, a measure of the distortion due to concealment, Δdr_i, is evaluated for each slice i. This represents the mean-squared error between the slice reconstructed at the decoder, and its concealed version in case a loss has occurred. We have deemed more effective to

---

Manuscript received February 8, 2008; revised May 12, 2008. This work was supported in part by the European Union under the SEA project. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Athanassios Skodras.

The authors are with the Dipartimento di Elettronica, Politecnico di Torino, 10129 Torino, Italy (e-mail: Enrico.Baccaglini@polito.it; Tammam.Tillo@polito.it; Gabriella.Olmo@polito.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2008.2001565

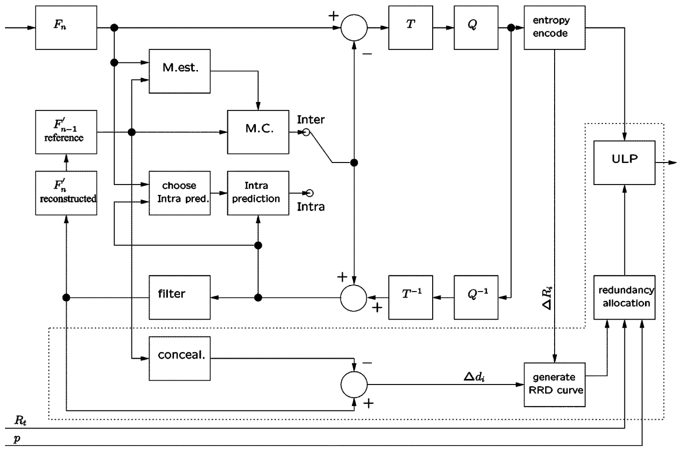

Fig. 1. Block diagram of a typical H.264/AVC encoder with the proposed stage (in dotted box). The box includes the concealment emulator stage, the RRD curve generator, the redundancy allocation procedure, and the ULP stage.

use the reconstructed slice than the original one, as this allows one to identify the residual error due to losses, keeping it well separated from the distortion due to compression.

Then, the drift caused by a lost slice i is accounted for by means of a proper weight parameter w_i. This takes into account the position of each slice in the current GOP, and the leakage in the prediction loop, by which the propagated distortion decays over time. According to the analytical model in [3], the leakage is determined by the fact that some MBs are encoded in I mode, by the presence of the de-blocking loop filter and because of the spatial filtering used for sub-pixel motion estimation. The parameter w_i is evaluated as [3]

$$w_i = \sum_{n=f(i)}^{N} \frac{1}{1+\gamma[n-f(i)]} \tag{1}$$

with N being the GOP length, and $f(i)$ a many-to-one mapping function, which receives as input a slice index and outputs the corresponding frame index, referred to the current GOP. The parameter γ depends on the video sequence characteristics and the encoding rate. In general, $\gamma > 0$ and the lower γ, the more the residual distortion propagates in the following frames. Therefore, its value usually increases when more spatial filtering is applied in the prediction loop, or when the introduced error includes high spatial frequencies that are easily removed by the loop filter. However, as discussed in Section III, we have verified that the value of γ has little impact on the system performance. Given w_i and Δdr_i, the encoder finally evaluates the metric $\Delta d_i = w_i \times \Delta dr_i$ for each slice of a GOP. Clearly, this semi-analytical approach to account for the drift is by far less computationally intensive than directly evaluating the impact of each lost slice on the overall GOP distortion; in fact, this would require the decoding of the entire GOP for each slice.

In addition to Δd_i, the SSR algorithm also requires the rate Δr_i devoted to encode each slice. This is available in the video entropy coding stage of the encoder [4], as shown in Fig. 1; therefore, it does not increase the system complexity. The pairs $(\Delta d_i,\, \Delta r_i)$ are then used to sort all the slices in a GOP. To this end, the *normalized residual distortion* (NRD) $\Delta d_i / \Delta r_i$ is employed; the normalization term Δr_i allows one to compare the distortion contributions of slices that are not necessarily encoded at the same rate.

The ordered list of pairs $(\Delta d_i,\, \Delta r_i)$ allows the encoder to generate a particular rate-distortion function, called *rate-residual-distortion function* (RRD). This is made of discrete points $[\hat{R}_j, D(\hat{R}_j)]$, where $\hat{R}_j = \sum_{i=0}^{j} \Delta r_i$ represents the cumulative rate of the j most important slices, and $D(\hat{R}_j) = \sigma^2 - \sum_{i=0}^{j} \Delta d_i$, $\sigma^2 = D(R_0) = D(0)$ being the variance of the video sequence; this term measures the distortion decrease achieved when the j most important slices are correctly received.

The RRD is used to pilot the ULP allocation stage as described in the following. It is worth noticing that it does not represent the source rate-distortion curve; instead, it classifies the slices in terms of their relative importance and it is evaluated on a GOP basis.

The second significant operation of the SSR encoder is the *unequal FEC allocation*. Given the different impact of lost slices on the GOP distortion, the level of FEC at the encoder should be dependent on the slice relevance and the probability of packet loss, p. In [10], a Lagrangian rate-allocation procedure has been proposed to evaluate the suitable level of protection for different quality layers, using RS erasure codes and assuming independent losses. A similar approach is adopted also in this letter, using the RRD function as input. From now on, and without loss of generality, we assume to use RS codes. Raptor codes or Fountain codes [11] can be employed as well. The value of p used to pilot the ULP allocation is assumed to be known at encoding time, and it can be determined by a feedback channel or by estimating the transmission channel characteristics.

To create M packets of R bits each, the Lagrangian optimization procedure partitions the sequence of ordered slices into groups (*chunks*). A chunk is made of a variable number of slices, which are protected with the same (M, k) RS code. Thus, the chunk can be decoded if at least k packets out of M are received. The Lagrangian optimization marks the RRD function at M different rates; chunk c is bounded by rates R_{c-1} and R_c $(1 \leq c \leq M)$. An example of RRD function and the corresponding ULP allocation scheme is reported in Fig. 2.

A chunk protected with the (M, k) code is divided into k parts of equal length, and it is accommodated into k different packets. The remaining $(M - k)$ packets host the RS parity bits required to protect the information according to the (M, k) RS code. Thus, each packet is composed by data (slices) and/or FEC of all the chunks. The resulting total output rate [10] is $R_t = \sum_{j=1}^{M} \alpha_j R_j$ with $\alpha_j = M/j(j+1)$ for $j = 1, \ldots M-1$, $\alpha_M = 1$ and $R_1, \ldots R_M$ being the M rates at which the RRD function is marked. The term R_t represents the Lagrangian constraint on the ULP system, as it is generally upper-bounded to a value R, dictated by the application.

Finally, the end-to-end distortion can be easily worked out as in [10]

$$E_d = q_0 \sigma^2 + \sum_{j=1}^{M} q_j D(R_j) \tag{2}$$

with q_j denoting the probability that j out of M packets are received and $D(R_j)$ being the RRD function evaluated in R_j, for $1 \leq j \leq M$.

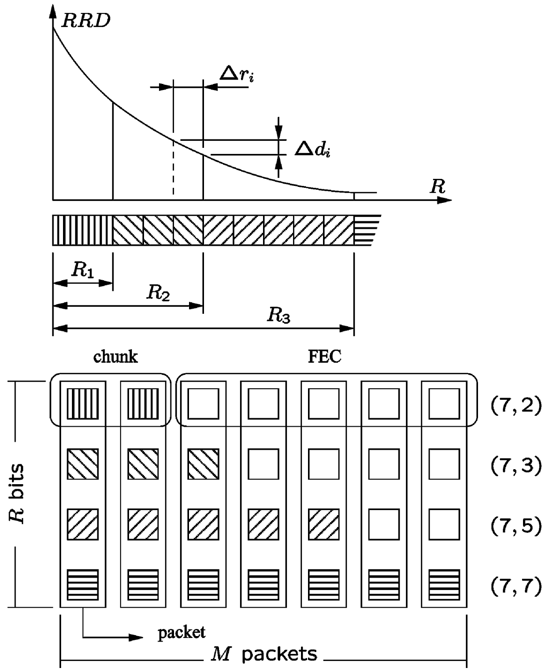

Fig. 2. Proposed allocation procedure. $M = 7$ packets, $c = 4$ chunks.

The optimization problem can be cast in terms of finding R_j, $j = 1, \ldots, M$, minimizing (2) while satisfying the constraint $R_t \leq R$. This problem is solved using Lagrangian optimization. The rates are located on the RRD function where slopes are in proportion to each other according to the ratio q_j/a_j. In a practical implementation, the rate R_j is defined at slice level, and the slopes on the RRD function are quantized and defined at slice level. Then, the algorithm should take into account the approximation $dD(R_j)/dR_j \approx \Delta d_i/\Delta r_i$, with $i = g(j)$ and g is a mapping function that generates the index of the slice that corresponds to the rate R_j in the RRD function.

Additional information may be required to inform the decoder about which slices have not been transmitted at all and should be simply concealed. In fact, the FEC allocation may truncate the source information in order to free space for the proper FEC codes. In any case, the impact of this extra information is negligible.

A. Decoder Operation

At the receiver side, packets are collected within a time window shorter than or equal to the playout deadline. Packets received after the end of the time window are assumed to be lost. Missed slices are recovered by the concealment stage. FEC decoding starts when the time window has elapsed. The algorithm decodes the most important slices first, as they are protected with stronger codes, which require fewer packets to be decoded. For example, said (M, k_{min}) the most powerful code employed, the decoding procedure can start when k_{min} packets are received. As more packets are received, further slices are decoded. Since slices are not received according to the corresponding frame ordering, the receiver should take into account a worst-case delay of one GOP time. In general, when all packets belonging to a GOP are received and decoded, the decoder can reorganize the sequence of slices according to the correct order and move it to the video application for displaying.

III. EXPERIMENTAL RESULTS

A set of simulations has been carried out using the first 90 frames of the standard CIF Foreman sequence at 30 fps. For the proposed algorithm, a GOP structure containing P slices only is adopted, and each slice is constrained to contain a maximum number of 66 MBs, leading to six slices per frame. The number of reference frames is set to five and the entropy coding scheme is CAVLC.

The packet size R is set to 400 bytes and is kept constant for all simulations so as to enable fair comparisons of the results under the same transmission constraints. In particular, for all the addressed schemes, each slice is smaller than 400 bytes and fits a packet.

Packet losses are assumed to be independent of each other, following a Bernoulli distribution with probability p. However, more sophisticated (e.g., Gilbert–Elliot) models can be easily accommodated in the algorithm. The performance is measured in terms of average luminance PSNR, obtained with 100 independent transmission trials. This amounts to the transmission of more than $4 \cdot 10^4$ packets, when working at 1000 kbps, which yields significant results from the statistical point of view.

The performance of the proposed SSR scheme is compared to three benchmark algorithms.

The first one, named *multiple description coding polyphase spatial sub-sampling* (PSS-MDC), has been proposed in [12]. Two sub-sequences are generated by splitting odd and even rows of each frame, and then H.264/AVC encoded. A similar version of the algorithm, which addresses four descriptions, is not considered here as a benchmark as its performance is proven to be inferior [13].

The second benchmark algorithm is a state-of-the-art multiple description coding (MDC) scheme called *redundant slice MDC* (RS-MDC) and proposed in [14]. It is based on the redundant slice option defined in the H.264/AVC standard. The redundant representation of a slice is used to replace missing portions of the compressed bitstream, thus yielding a certain degree of error resilience.

The third benchmark is a scheme which simply relies on the Nokia spatiotemporal concealment [15] without any FEC strategy. The complexity of this scheme is extremely limited. It allows one to evaluate if the extra complexity of using unequal FEC allocation is worthy. Finally, the error-free H.264/AVC curve is reported as a performance upper-bound.

In Fig. 3, the average PSNR obtained with SSR and the benchmark algorithms, $p = 10\%$ and GOP size 11, are shown versus the total rate. For all the algorithms, the total output rate varies from approximately 100 to 1600 kbps, according to the quantization parameter (QP), which is in the range 22–51. As for SSR, different values of γ have been tested ($\gamma = 0, 2$, and $\gamma \rightarrow \infty$). As shown, γ has little impact on the average PSNR (below 0.5 dB).

From the reported results, it can be noticed that the proposed approach exhibits a noticeable performance improvement with respect to PSS-MDC; the gain in PSNR ranges from 2 to 5 dB. SSR significantly outperforms (up to 8 dB) the Nokia concealment system; this confirms that a well-conceived ULP strategy is extremely effective in case of significant packet loss rates. Compared with RS-MDC, the proposed algorithm yields

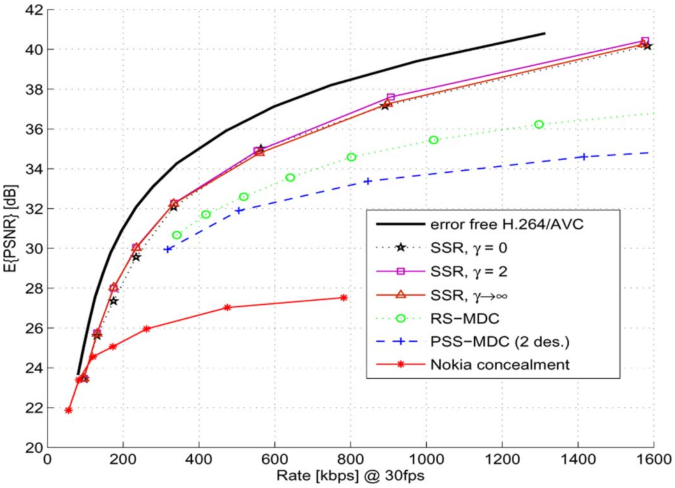

Fig. 3. Foreman CIF sequence, 30 fps, $GOP = 11, p = 10\%$. Average PSNR versus total rate for SSR and benchmark algorithms.

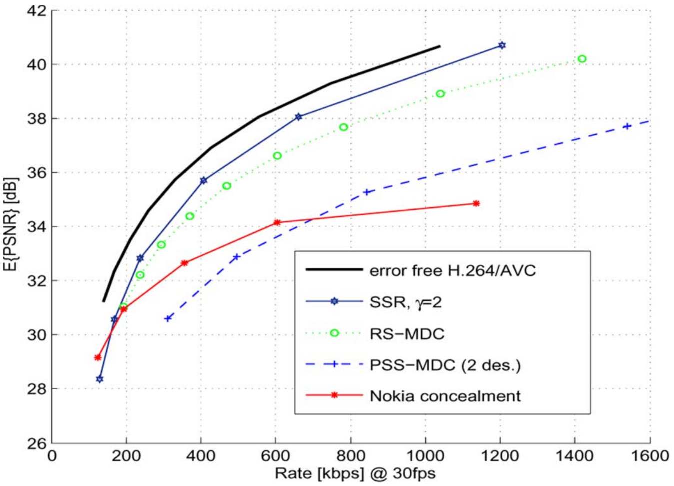

Fig. 4. Foreman CIF sequence, 30 fps, $GOP = 45, p = 1\%$. Average PSNR versus total rate for SSR and benchmark algorithms.

a PSNR enhancement of about 2.5 dB. It is worth noticing that RS-MDC adopts CABAC, whereas in SSR, CAVLC is employed. Preliminary results have shown that if CABAC is used also in SSR, an additional 0.5 dB gain can be achieved.

Similar results are reported in Fig. 4 for $p = 1\%$. A PSNR gain of approximately 3.5 dB can be obtained with respect to PSS-MDC, 1.5 dB versus RS-MDC, and up to 4.5 dB versus the Nokia spatiotemporal concealment. This results are significant, as they reveal that an optimized ULP strategy is effective also at low packet loss rates.

As for complexity, if compared to standard H.264/AVC encoding running on an Intel Pentium IV 2 GHz, 1 GB of RAM, the generation of the RRD function increases the encoding time of about 2.5%, whereas the FEC allocation procedure has negligible impact on the encoding time. As for decoding, the FEC decoding complexity is strictly dependent on the code family. For example, if Raptor codes are applied, the decoding time is linear with the number M of packets to be decoded.

IV. CONCLUSIONS AND FUTURE WORK

In this letter, we proposed a novel ULP-based scheme that protects a video sequence according to the relevance of its slices. This is evaluated at the encoder side using a normalized residual distortion metric, worked out by emulating the decoder concealment stage, and properly weighted in order to take the drift effect into account. Simulation results showed that the proposed algorithm significantly outperforms state-of-the-art approaches, especially at high packet loss rates. Future developments include the use of digital fountain/raptor codes, a more detailed study of the complexity of the encoding/decoding stages, and further study on the allocation procedure.

ACKNOWLEDGMENT

The authors would like to thank STMicroelectronics for providing the H.264/AVC codec.

REFERENCES

[1] J. Li, K. Nahrstedt, and H. Zhang, "Editorial: Special Section on Peer-to-Peer Video Streaming," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1551–1553, Dec. 2007.

[2] P. Baccichet, J. Noh, E. Setton, and B. Girod, "Content-aware P2P video streaming with low latency," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2007.

[3] K. Stuhlmüller, N. Färber, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[4] T. Wiegand, G. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[5] Y. Xu and Y. Zhou, "H.264 video communication based refined error concealment schemes," *IEEE Trans. Consum. Electron.*, vol. 50, no. 4, pp. 1135–1141, Nov. 2004.

[6] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 819–828, Jun. 2000.

[7] J. Goshi, A. Mohr, R. Ladner, and E. Riskin, "Unequal loss protection for H.263 compressed video," *IEEE Trans. on Circuits Syst. Video Technol.*, vol. 15, no. 3, pp. 3232–3234, Mar. 2005.

[8] T. Fang and L.-P. Chau, "A novel unequal error protection approach for error resilient video transmission," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 2005, vol. 4, pp. 4022–4025.

[9] N. Thomos, S. Argyropoulos, N. Boulgouris, and M. Strintzis, "Robust transmission of H.264/AVC video using adaptive slice grouping and unequal error protection," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2006, pp. 593–596.

[10] R. Puri and K. Ramchandran, "Multiple description source coding using forward error correction codes," in *Proc. 33rd Asilomar Conf. Signals, Systems and Computers*, Oct. 1999, vol. 1, pp. 342–346.

[11] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Foundations of Computer Science*, Nov. 2002, vol. 1, pp. 271–280.

[12] R. Bernardini, M. Durigon, R. Rinaldo, L. Celetto, and A. Vitali, "Polyphase spatial subsampling multiple description coding of video streams with H.264," in *Proc. IEEE Int. Conf. Image Processing*, Oct. 2004, pp. 3213–3216.

[13] A. Vitali, F. Rovati, R. Rinaldo, R. Bernardini, and M. Durigon, "Low-complexity standard-compatible robust and scalable video streaming over lossy/variable bandwidth networks," in *Proc. ICCE 2005 Digest of Technical Papers. Int. Conf. Consumer Electronics*, Jan. 2005, pp. 7–8.

[14] T. Tillo, M. Grangetto, and G. Olmo, "Redundant slice optimal allocation for H.264 multiple description coding," *IEEE Trans. Circuits Syst.*, vol. 18, no. 1, pp. 59–70, Jan. 2008.

[15] Y.-K. Wang, M. Hannuksela, V. Varsa, A. Hourunranta, and M. Gabbouj, "The error concealment feature in the H.26L test model," in *Proc. IEEE Int. Conf. Image Processing*, Sep. 2002, vol. 2, pp. 729–732.