MACROBLOCK-BASED RETRANSMISSION FOR ERROR-RESILIENT VIDEO STREAMING

J. Christian Schmidt and Kenneth Rose

Department of Electrical and Computer Engineering
University of California
Santa Barbara, California 93106–9560
Email: {schmidt,rose}@ece.ucsb.edu

ABSTRACT

This paper revisits the problem of source-channel coding for error-resilient video streaming. We propose a new method to enable adaptive redundancy in the bitstream: fine-grain retransmission. Redundancy decisions are made per macroblock (MB), which are locally adaptive and of low overhead, as opposed to coarse packet-level redundancy (e.g. forward error correction). In this scheme, the encoder jointly optimizes the coding mode and redundancy per MB. A corresponding algorithm is presented for exploiting this redundancy at the decoder. The proposed method is general in nature, and can be implemented on top of any (hybrid) video codec. An example implementation is provided, which uses the redundant slice mechanism of H.264 (JM 13.2 reference software). Simulation results show significant performance gains over conventional error-resilient coding techniques.

Index Terms— video streaming, error resilience, adaptive retransmission, redundant slices, H.264

1. INTRODUCTION

Despite recent advances in network technology and infrastructure, video streaming over packet networks remains a challenging problem, motivating ongoing research efforts into error resilience mechanisms to mitigate the impact of packet loss. In live streaming, source-channel coding algorithms are used to optimize the rate-distortion (RD) tradeoff between source compression and distortion at the receiver. Accurate estimation of the end-to-end distortion is central to this approach and is provided by the "recursive optimal per-pixel estimate" (ROPE) [1, 2], which enables end-to-end distortion estimation by tracking, per pixel, the first and second moments of the decoder reconstruction. ROPE accounts for all sources of distortion such as quantization, packet loss, error propagation, and error concealment at the decoder. It has been successfully applied to macroblock (MB) coding mode selection [1], error-resilient motion estimation/compensation [3] and reference picture selection [4], multiple description video coding [5], and joint mode and quality of service (QoS) selection [6]. For more details on ROPE, the reader may refer to [1, 2].

Fundamentally, an error-resilient encoding algorithm balances the conflicting objectives of mitigating channel loss and stopping error propagation (Intra-refresh, high rate cost) versus compression efficiency (Inter-coding, temporal prediction). Rather than allocate the entire bit budget for source coding, some rate may be designated for channel protection, thereby trading some source coding fidelity for a decrease in the effective packet-loss rate (PLR). Common channel coding mechanisms are forward error correction (FEC) or automatic repeat request (ARQ). Each is subject to practical drawbacks. ARQ depends on the availability of feedback, and requires a longer buffering period due to feedback delay. FEC can be applied per packet (in bit-error channels) or across packets (packet-erasure channels). In either case, FEC rate allocation is performed at the packet level, i.e. *after* encoding. Due to uneven packet sizes and necessary padding, it is difficult to estimate the effective rate. In [6], the authors proposed a Trellis-based algorithm to address this problem, albeit at the cost of delay and complexity.

In practice, FEC lowers the effective PLR as experienced by the source coder, leaving open the traditional issue of error propagation. Not all MBs are equally important to the overall reconstruction quality. We propose a new source encoding scheme that enables adaptive use of redundant coding per MB. Note that a similar scheme has been proposed in [9], and we will discuss similarities and differences below. Since the redundant encoding scheme depends on accurate estimation of end-to-end distortion and hence the importance of an MB, we extend the basic ROPE to the objective at hand. Section 2 introduces the redundant coding scheme using selective re-transmission per MB. We describe a simple encoding algorithm for joint MB mode and redundancy decisions, and propose a matching decoder error concealment algorithm. Fur-

This work was supported in part by the University of California MICRO program, Applied Signal Technology Inc., Cisco Systems Inc., Dolby Laboratories Inc., and Qualcomm Inc.

ther, we discuss the rate overhead of the transmission scheme and its impact. Then, in Section 3, we implement the algorithm on top of the JM 13.2 reference software [8] and compare its performance with other error-resilient coding methods. The paper concludes in Section 4 with a brief summary and future research directions.

2. REDUNDANT CODING

2.1. Adaptive retransmission per MB

Rate allocation to source and channel coding can be subsumed within the general RD framework. Optimizing this trade-off at the MB level enables fine-grain, adaptive control of redundancy, unlike coarse packet-level mechanisms such as FEC. Moreover, it minimizes coding-induced dependencies between MBs and thus simplifies the optimal encoder. A simple approach is to control the number N of transmissions per MB. Every MB is transmitted at least once per frame, and may be re-transmitted ($N \geq 1$). Let R and D denote the rate and end-to-end distortion costs for a certain MB coding mode. Assuming independent, identically distributed PLR p, transmitting an MB N times in *separate* packets results in an effective PLR p^N. The associated rate cost is NR and the distortion is $D_N < D$. Note that, while we assume iid packet loss for simplicity of implementation, ROPE has been successfully extended to bursty channels [5].

We need to devise a mechanism to effectively send the MBs selected for re-transmission with low overhead, using additional data packets which contain only redundantly coded MBs. One possible approach is application-level packet tagging and filtering. For each packet, we need to indicate if it contains primary bitstream data or redundant MB transmissions. The streaming application then parses the packets and passes them to the decoder as necessary. The H.264 [7] baseline profile includes a mechanism called redundant slices which offers exactly this functionality at the bitstream-level, i.e. in a *standard-compliant* way. Each frame in the video stream consists of one or more primary slices, which contain the regular video data, and may be followed by secondary slices with redundant data.

2.2. Encoding algorithm

Traditional error-resilient MB mode selection iterates over all available coding modes and evaluates their rate (R) and distortion (D) costs, selecting the mode with the smallest Lagrangian $J = D + \lambda R$. We propose a simple encoding algorithm for joint coding mode and redundancy decisions per MB. The encoder now considers each MB mode in conjunction with different retransmission options and estimates the resulting RD costs. Ignoring some minor overhead for headers (to be discussed later), the rate cost for N retransmissions is NR. The more involved task is that of accurately estimating end-to-end distortion, D_N, where we resort to ROPE with effective PLR p^N. The pair of MB mode and retransmission value N achieving the minimal Lagrangian J_N is selected.

2.3. Decoding algorithm

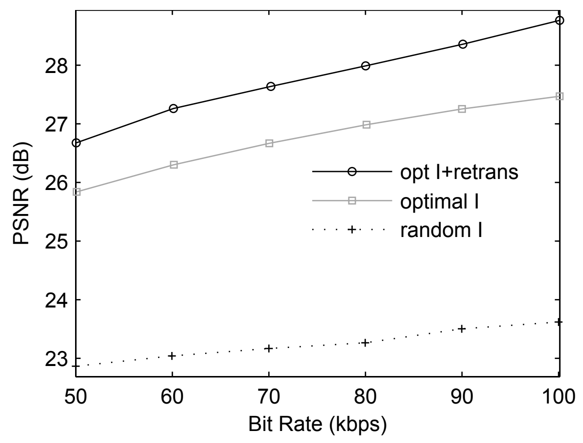
In order to take advantage of available redundant MBs, we need to modify the decoding/concealment algorithm. Such modifications do not conflict with the standard compatibility of the technique. A simple compliant H.264 decoder may simply discard the redundant slices. However, it is perfectly legitimate to design an enhanced algorithm which exploits available redundant information while remaining fully compatible with the standard bit stream. We note further that accurate end-to-end distortion estimation at the encoder, such as performed by ROPE, depends on knowledge of the concealment procedure employed by the decoder.

The decoder operation on primary data packets remains unchanged. Let the decoder track the decode status of each MB, i.e. whether it was received and reconstructed, or lost. When a redundant packet is received for a lost MB, the decoder reconstructs that MB from it and updates its status. MBs which are not retransmitted ($N = 1$) are signaled as "skip" in the redundant packet, minimizing overhead. Note that only *coded* MBs can be transmitted redundantly in this setting. MBs still marked lost after decoding of redundant packets will then be concealed via traditional techniques. We use simple copy, although more sophisticated concealment algorithms may be used, e.g. motion copy. But this is of secondary importance and orthogonal to the redundant slice decoding/concealment mechanism we propose.

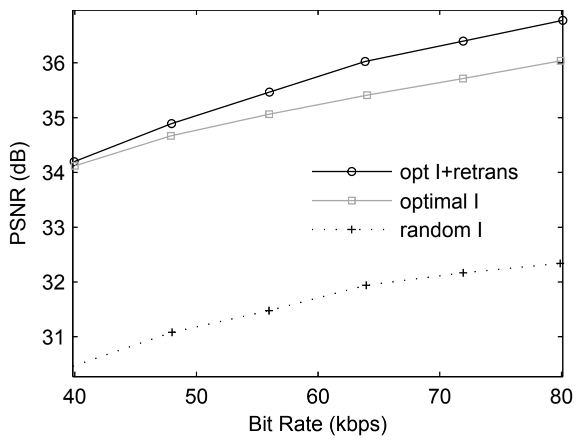
A similar scheme using redundant slices and flexible macroblock ordering (FMO) has been proposed in [9]. Here, a video frame is partitioned into slice groups, based on a simple model of the significance of the underlying MBs. While the signaling overhead for the slice groups in [9] is relatively low, slice groups trade off source coding efficiency for improved decoder concealment opportunities. A major advantage of our scheme is that it enables adaptive redundant encoding on a per-MB basis, instead of the entire frame or rectangular ROIs in [9]. Our proposed scheme evaluates the end-to-end RD cost on a per-MB basis using the optimal estimate (ROPE), instead of an MB significance model, limiting the rate overhead of the redundant slices.

2.4. Rate overhead

Besides the additional rate for re-transmitting an MB (known at encode time), signaling the redundant MB representations incurs a small overhead. When one or more MBs are re-transmitted, the additional packet/slice header(s) and the run-length (skip) coding for non-redundant MBs cause a small overhead which cannot be accounted for exactly during RD optimization (at least not without incurring significant delay and complexity due to inter-MB dependencies).

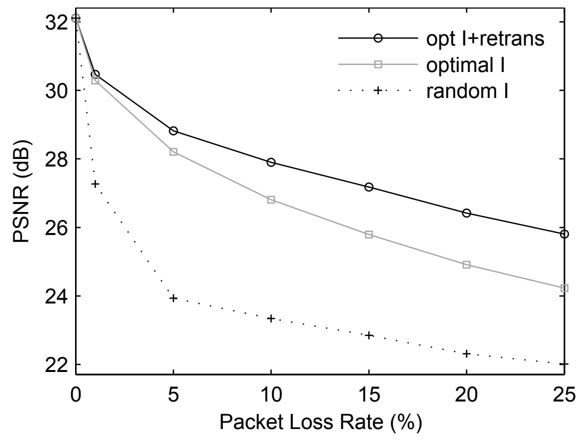


(a) foreman, 50–100 kbps

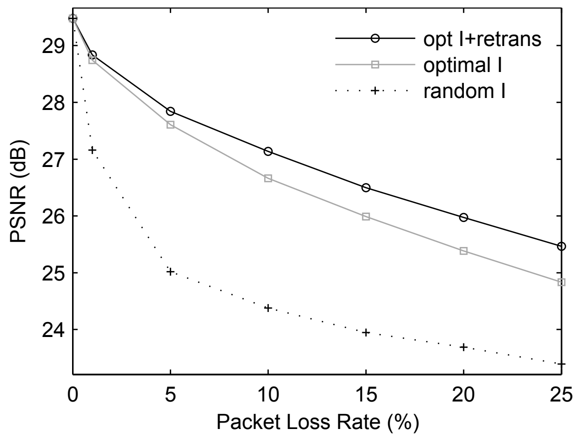


(b) hall monitor, 40–80 kbps

Fig. 1. Delivery performance, PSNR vs. bit rate (qcif, 10 fps, p=10%)



(a) foreman, 75 kbps



(b) coastguard, 75 kbps

Fig. 2. Delivery performance, PSNR vs. PLR p (qcif, 10 fps, 75 kbps)

This could be mitigated by verifying the *actual* RD costs prior to writing the redundant data. This may be a concern when only a few MBs are candidates for retransmission. When a sufficient numbers of MB is coded redundantly, the overhead is negligible relative to the actual retransmission rate, and is spread evenly across the redundant MBs. As we will see in the simulation results, it is safe to ignore these concerns in practical circumstances.

3. SIMULATION & RESULTS

We implemented the proposed algorithm for MB-based retransmission on top of the JM 13.2 reference software [8] (labelled "opt I+retrans" in Figures 1–3). For comparison, we also provide results of conventional ROPE optimal MB coding mode selection ("optimal I"), as well as random MB Intra refresh ("random I") implemented in the JM reference software. All sequences were encoded at 10 fps, QCIF resolution, packet size ≤ 512 bytes. The bitstreams were then simulated at different packet loss rates (PLR), and performance was averaged over 500 patterns at each PLR.

3.1. Performance vs. bit rate

In the first experiment (Figure 1), we compared the three methods across a range of effective bit rates, with the PLR fixed at 10%. Figure 1(a) shows foreman sequence encoded at target bit rates from 50–100 kbps. Actual bit rates vary slightly and are correctly denoted on the rate (x-) axis. The proposed MB-adaptive redundant coding consistently outperforms conventional ROPE MB coding, gaining between 0.7 db up to 1.2 dB. A second example is the hall monitor sequence (target bit rates 40–80 kbps, Figure 1(b)). Redundant coding achieves identical performance with optimal Intra at

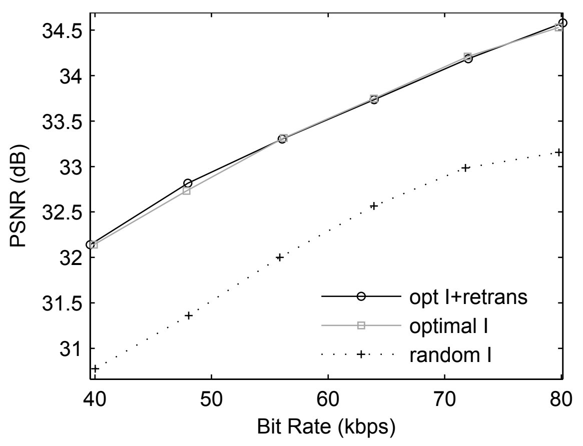


Fig. 3. Worst case in all simulations: gains diminished but no loss (container, qcif, 10 fps, p=10%)

low bit rates, and achieves up to 0.7 dB more towards higher bit rates. The performance of random Intra refresh trails significantly behind both methods for both sequences.

3.2. Performance vs. PLR p

We compared the three algorithms across a PLR range of 1–25%. Figure 2(a) shows the results for the foreman sequence, encoded at 75 kbps. The new algorithm has similar performance to conventional ROPE for p=1%, but its performance degrades more gracefully as the PLR increases, achieving gains of up to 1.6 dB over the conventional method. Figure 2(b) shows the results for the coastguard sequence, encoded at 75 kbps. Again, the performance is similar for p=1%, but the new adaptive retransmission method achieves gains of up to 0.6 dB towards higher PLR.

3.3. Validity of overhead assumptions

In Section 2.4, we discussed possible issues stemming from ignoring a small rate overhead during encoding. Figure 3 shows the worst result we have obtained so far, for the sequence container at bit rates 40–80 kbps (p=10%). In this example, which demonstrates the worst measured impact of neglecting overhead in rate estimates, the new algorithm achieves no gains over conventional ROPE MB mode selection, but it does not underperform its competitors. Hence, it appears justified to ignore the overhead for redundant coding. We note again that it is possible to eliminate these assumptions at the cost of complexity and delay.

4. CONCLUSION & FUTURE WORK

This paper presents a new error-resilience mechanism that enables fine-grain adaptation between source coding fidelity and channel error resilience per MB. The algorithm makes a joint decision for coding mode and redundancy (in the form of retransmission) per MB. Initial results show RD performance gains over conventional ROPE-based MB coding mode selection.

The proposed scheme can be further enhanced beyond these preliminary results: Currently, retransmission consists of identical repetition of the MB. The algorithm could involve principles of multiple descriptions (e.g. via prediction from different reference pictures) and enable improved reconstruction when both descriptions are received. It is also possible to only retransmit partial MB information based on RD tradeoff optimization, etc. Another interesting extension would be data partitioning, as enabled by the high profile of H.264. Here, the individual components affecting reconstruction quality, e.g. coding mode, motion information and residual, and intra prediction data, are separated and can be clearly identified. Adaptive redundancy for these components should also enable performance gains.

5. REFERENCES

[1] R. Zhang, S. L. Regunathan, and K. Rose, “Video coding with optimal inter/intra-mode switching for packet loss resilience”, *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 966–976, 2000.

[2] H. Yang, and K. Rose, “Advances in Recursive Per-Pixel End-to-End Distortion Estimation for Robust Video Coding in H.264/AVC”, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, no. 7, pp. 845–856, 2007.

[3] H. Yang, and K. Rose, “Rate-Distortion Optimized Motion Estimation for Error Resilient Video Coding”, *IEEE ICASSP*, vol. 2, pp. 173–176, 2005.

[4] A. Leontaris, and P. C. Cosman, “Video compression for lossy packet networks with mode switching and a dual-frame buffer”, *IEEE Trans. Image Processing*, vol. 13, no. 7, pp. 885–897, 2004.

[5] B. A. Heng, J. G. Apostolopoulos, and J. S. Lim, “End-to-end rate-distortion optimized mode selection for multiple description video coding”, *IEEE ICASSP*, vol. 5, pp. 905–908, 2005.

[6] E. Masala, H. Yang, K. Rose, and J. C. De Martin, “Rate-distortion optimized slicing, packetization and coding for error-resilient video transmission”, *DCC*, pp. 182–191, 2004.

[7] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, “Advanced video coding for generic audiovisual services”, *ITU-T Recommendation H.264*, ISO/IEC 14496-10 AVC, March 2005.

[8] H.264/AVC Software Coordination: *http://iphome.hhi.de/suehring/tml/*

[9] P. Baccichet, S. Rane, A. Chimienti, and B. Girod, “Robust Low-Delay Video Transmission using H.264/AVC Redundant Slices and Flexible Macroblock Ordering”, *IEEE ICIP*, vol. 4, pp. 93–96, 2007.