FINE GRAIN ADAPTIVE FEC (FGA-FEC) OVER WIRELESS NETWORKS

Yufeng Shan, John W. Woods, Shivkumar Kalyanaraman

Department of ECSE, Rensselaer Polytechnic Institute, Troy, NY 12180, USA

ABSTRACT

In this paper, we extend our proposed FGA-FEC coding scheme, a generalized MD-FEC method, to wireless networks. To protect the encoded scalable video bitstream over a lossy channel and facilitate content adaptation at intermediate nodes, we use product codes based on BCH/CRC codes as row codes and RS codes as column codes. We give a fast algorithm to optimize the product codes within several iterations from a near optimal point. Simulations show good performance in both content adaptation and protection.

***Index Terms*—** Scalable, content adaptation, source channel coding, wireless

1. INTRODUCTION

Simultaneously streaming video to heterogeneous devices is a challenging problem, since different users may have different video frame-rate, resolution, and quality preferences, as well as computational and connection-link capabilities. In [1], we proposed a fine grain adaptive forward-error correction (FGA-FEC) coding scheme for scalable video streaming, that can achieve efficient and precise adaptation of the encoded bitstream (adapting both the video and the error control codes) to satisfy heterogeneous users without complex transcoding at intermediate overlay nodes.

For fading channels, product channel coding [2] is proved to be an efficient error protection method for scalable end-to-end image transmission. Within packets, product codes use the concatenation of a rate compatible punctured convolutional code (RCPC) and an error detecting cyclic redundancy check (CRC) code as the row code. Across packets, RS codes are used as column codes. Sachs *et al* [3] introduced a multiple-description product code which aims at optimally generating multiple, equally-important wavelet image descriptions.

None of the papers consider the simultaneous adaptation of product codes and image or video data for multiple heterogeneous users. Compared to image coding, scalable video has more degrees of adaptation, and the users' requirements are also more diverse. A scalable video bitstream has three basic types of scalability: temporal (frame-rate), spatial (resolution), and SNR (quality) scalability. Bitstream subsets corresponding to lower frame-rate/resolution/quality of the video are embedded in bitstreams corresponding to higher frame-rate/resolution/quality. Different sub-bitstreams can be extracted in a simple manner without transcoding, to readily accommodate a variety of users considering their video preferences and connection bandwidth. Therefore, to protect scalable video for diverse users over error-prone channels, we should consider not only the protection scheme, but also the feasibility of bitstream and error-control code adaptation. In this paper, we generalize the FGA-FEC scheme [1] for both protection and adaptation of this kind of highly scalable bitstream over wireless networks. We attempt to show that FGA-FEC plus our scalable coder MC-EZBC fits in such scenarios and successfully generalizes them to spatial, temporal and SNR scalability for heterogeneous video delivery over wireless networks.

In Section 2, we describe the details of our method. Simulated and experimental results are given in Section 3.

This work is supported by ARO grant W911NF-04-1-0300, 2004.

2. FGA-FEC OVER WIRELESS CHANNEL

2.1. FGA-FEC encoding

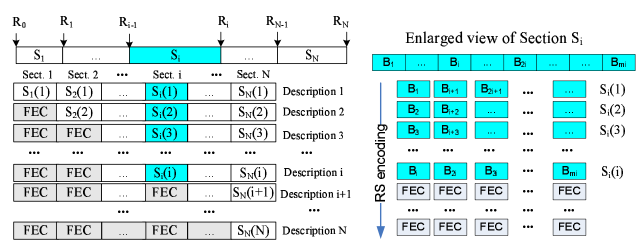


Fig. 1. FGA-FEC encoding of one GOP. Here, FEC is added vertically at block level and each horizontal row of blocks is packetized into one network packet.

Our FGA-FEC encoding method (Fig. 1) extends MD-FEC [4] by adding scalability (adaptation) features. Given a GOP of scalable-coded video bitstream organized from MSB (R_0) to LSB (R_N), shown at the top in Fig. 1, suppose we want to encode this GOP into N descriptions, we first run an optimal bit allocation scheme and divide the bitstream into N sections S_i, ($i \in [1, N]$), marked with source-rate break points $R_0, R_1, R_2, ..., R_N$, where $R_0 \leq R_1 \leq R_2 \leq ... \leq$

R_N and $R_0 = 0$. Section S_i ($i \in [1, N]$) is further split into equal size subsections with each subsection i blocks. These subsections are encoded by an RS(N, i) code vertically at block level to generate parity blocks. Since each block column is independently coded, at intermediate node, we can adapt the bitstream by easily removing related columns and/or dropping descriptions [1, 5], both source data and parity bits, to satisfy diverse users.

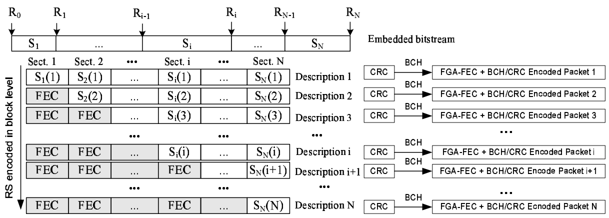


Fig. 2. Generalized FGA-FEC with product codes

For a wireless network, after FGA-FEC encoding, we further encode each description using a BCH code with CRC error detection to protect it from bit errors as shown at Fig. 2. We choose systematic BCH over RCPC based on its simplicity in decoding/encoding, as needed for intermediate node adaptation. FGA-FEC can encode and adapt the product codes based on both channel conditions and user video preference, as well as user predefined adaptation order. Here, *adaptation order* is the user's chosen order to adapt quality, frame rate and resolution, as needed. A user can chose to adapt downward the three factors in any particular order.

Suppose a user's video preference is to view a video at L_t temporal layer, L_s spatial layer and PSNR $\geq \gamma'$ dB, and the user's minimum tolerable bitstream is at $L_{t\,\min}$ temporal layer, $L_{s\,\min}$ spatial layer and PSNR $\geq \gamma$ dB, $\gamma' \geq \gamma$. Therefore, the user's video request ranges from $\{L_t, L_s, \gamma'\}$ to $\{L_{t\,\min}, L_{s\,\min}, \gamma\}$. Along the user's adaptation order, the server or intermediate nodes need to find the best possible video for this user within its requested bitstream range in response to available bandwidth.

The product code optimization problem is to find a concatenated column RS code assignment c_c and row BCH code assignment c_r from a set of RS codes C_{RS} and BCH codes C_{BCH}, such that the end-to-end distortion is minimized and the corresponding PSNR$\geq \gamma$ dB.

$$c_c, c_r = \underset{c_c \in C_{RS},\ c_r \in C_{BCH}}{\text{argmin}} \quad E[D|C_{RS}, C_{BCH}, C_{CRC}], \quad (1)$$

subject to:

$$R_s + R_{RS} + R_{CRC} + R_{BCH} \leq B,$$

where C_{CRC} is the CRC code set; R_s is the source rate, R_{RS} is the rate allocated to RS parity bits, and R_{CRC} (R_{BCH}) are the rates allocated to CRC (BCH) check bits. Here, B denotes the maximum available channel bitrate. We will use a fixed 32 bit CRC code in this paper, hence, R_{CRC} is a constant.

Given a BCH (n, k, t) codeword, number of bit errors larger than t in the codeword cannot be corrected, the probability of decoding error is

$$P_{BCH}(E) = \sum_{j=t+1}^{n} \binom{n}{j} p_b^j (1 - p_b)^{n-j}, \quad (2)$$

where p_b is the channel bit-error rate. Decoding failures in the row codes are treated as erasures when decoding the column RS codes. Given p_b, and the probability of a packet being dropped due to congestion/route disruption is p_{drop}, the probability of a packet erasure p after BCH decoding is approximately:

$$p = p_{drop} + (1 - p_{drop}) \times P_{BCH}(E). \quad (3)$$

After assigning a BCH code and a CRC code, the available bandwidth for RS codes and video data becomes $B - R_{CRC} - R_{BCH}$. We need to optimize the assignment of column RS codes under this rate constraint. The goal is to find the bitrate partition $R = \{R_1, R_2, ..., R_N\}$ in Fig. 2, which minimizes the end-to-end mean distortion $E[D(R)]$, and the corresponding PSNR $\geq \gamma$ dB. Please note that at every adaptation level, we need to use different $D(R)$ curves for the RS code optimization, an example is shown at Fig. 3.

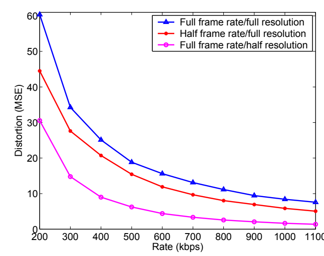


Fig. 3. $D(R)$ curves at various adaptation levels.

Optimal column code assignment at a certain given BCH code and CRC code is a constrained optimization problem and can be solved by using Lagrange multiplier method [4, 5]. Then, the optimal product code could be achieved by exhaustively searching over all possible BCH codes along the adaptation order. At a certain adaptation level, assign BCH code, update $R_s + R_{RS} = B - R_{CRC} - R_{BCH}$, optimize RS code (this is called one iteration), until exhaust all possible BCH codes. We propose a fast search algorithm which can find the product code assignment within several iterations.

2.2. Fast BCH code optimization

From (3), we know that BCH decoding error contributes to packet loss probability. At a certain BER, stronger code would result in a lower probability of decoding error, thus reduce the probability of packet erasure. Also allocating more bandwidth to BCH code would result in less bandwidth allocated to source and RS code, hence higher distortion. Therefore, we can find the optimal point by leveraging the two factors.

We tested via exhaustive search over videos *Foreman* (CIF, 18 GOPs), *Mobile* (SIF, 8 GOPs) and *Football* (SIF, 7 GOPs) at various BER, available bandwidth, and number of descriptions. Fig. 4 shows an example in one of these tests. The task is to protect an MC-EZBC encoded *Foreman* CIF sequence, GOP #7. Here, $N = 64$, $B = 980$Kbps, $p_{drop} = 0.05$, $p_b = 2 \times 10^{-3}, 1 \times 10^{-3}, 5 \times 10^{-4}$ and 1×10^{-4}, respectively. Fig. 4(a) shows the probability of successfully decoding these BCH(n, k, t) codes at given BERs. Fig. 4(b) shows the zoomed corresponding optimized PSNR vs. t. Table 1 shows the optimization results of this test.

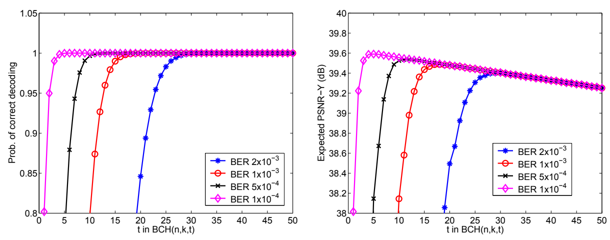


Fig. 4. (a) probability of successful BCH(n, k, t) decoding at various channel BER vs. t; (b) Average PSNR of video vs. t

BER	$E[D]$(PSNR)	BCH(n, k, t)
0	39.63	none
1×10^{-4}	39.59	BCH(8191,8125,5)
5×10^{-4}	39.54	BCH(8191,8034,12)
1×10^{-3}	39.49	BCH(8191,7956,18)
2×10^{-3}	39.40	BCH(8191,7800,30)

Table 1. The results of optimal assignment at different BER

Our key observation is that the points near the knee of Fig. 4(a) are near optimal points in Fig. 4(b). Therefore we can pick up a starting point t from these knee points and locally search to find the optimization result. Since the expected distortion $E[D]$ curve is concave around the optimal point, we can fist test three points $(t-1, t, t+1)$, find the search direction of t. After that, we progressively allocate bandwidth to BCH codes along the search direction, and then optimize the column RS codes at each BCH code assignment, until find the best possible point.

We use a threshold method to choose the starting point. We pick a value of the threshold ε, and test the probability of correctly decoding a BCH(n, k, t) code, $P_{BCH}(C)$, at a certain BER with ε, where $P_{BCH}(C) = 1 - P_{BCH}(E)$. The smallest point t with $P_{BCH}(C) > \varepsilon$ is the initial point. Obviously, different threshold ε corresponds to different optimization performance in terms of number of iterations to reach the optimal point. Simulations show that $\varepsilon = 0.999$ is a good starting point. Experiments show that good convergence is obtained with just three to five iterations on average. Formula (2) used in Fig. 4(a) can be stored in a small table. Algorithm 1 summaries our product code optimization method.

Algorithm 1: Product code assignment optimization

1 Pick start point t, such that $P_{BCH}(C) > \varepsilon$;
2 Assign BCH(n,k,t) code;
3 Calculate p as (3);
4 Optimize RS codes $t-1, t, t+1$, calculate $E[D]_{t-1}$, $E[D]_t$, and $E[D]_{t+1}$;
5 If $E[D]_t \leq$min$(E[D]_{t-1}, E[D]_{t+1})$, go to Step 9;
6 If $E[D]_{t-1} \leq$ min$(E[D]_t, E[D]_{t+1})$, search lower t, go to Step 8;
7 If $E[D]_{t+1} \leq$ min$(E[D]_{t-1}, E[D]_t)$, search higher t;
8 Iterate on t a few steps;
9 If PSNR$\geq \gamma$, solution found, **Stop**, otherwise, move down one adaptation level following user adaptation order. If adaptation level exhausted, **Stop**, no video is sent. Otherwise, go to Step 4;

While full Lagrange-based optimization is performed at the server, only FGA-FEC adaptation [1, 5] consisting of shortening and/or dropping packets is done at intermediate nodes.

3. SIMULATIONS AND EXPERIMENTS

We performed simulations and experiments using test sequences, *Foreman* CIF, 288 frames, *Football* SIF, 112 frames and *Mobile* SIF, 128 frames. All sequences are at 30 fps, 16 frames/GOP. The scalable source coder is MC-EZBC. BCH codes are applied to both MD-FEC and FGA-FEC. We present averages over at least ten runs.

3.1. FGA-FEC vs. MD-FEC in SNR adaptation

We compare FGA-FEC vs. MD-FEC in wireless network [3] to adapt to different bandwidth, by sending the encoded *Foreman* sequence to the receiver with bandwidth ranging from 200 Kbps to 1000 Kbps as shown in Fig. 5, where node1 is the sender, node2 is an intermediate node that can perform bitstream adaptation (detailed adaptation algorithm is in [5]) and BCH decoding/re-coding, and node3 is the receiver. The BER between node2 and node3 are set to 1×10^{-4}, $p_{drop} = 0.05$ at node2. There is no congestion between node1 and node2.

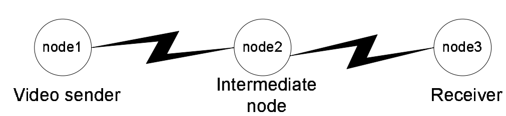


Fig. 5. The topology of simulations and experiments

We first encode each GOP of *Foreman* to 64 descriptions with $p_{drop} = 0.1$, $p_b = 1 \times 10^{-4}$, $B = 1$ Mbps and then send over the channel. Fig. 6 shows the observed video quality (PSNR) at different available bandwidths. Clearly, FGA-FEC has much better performance in response to channel condition. This is because MD-FEC responds to limited bandwidth between nodes 2 and 3 by only dropping packets, hence some useless data is sent within remaining packets, because they are not matched to the lower bitrate. On the other hand, FGA-FEC adaptation is performed actively by both packet shortening and packet drop, and so avoid transmission of useless date, thus saving bandwidth for useful data and hence has better adaptation performance.

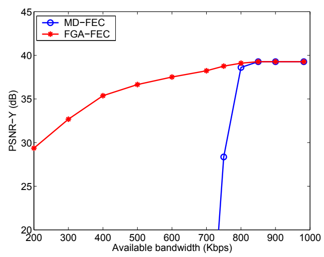


Fig. 6. FGA-FEC vs. MD-FEC SNR adaptation to different available bandwidth from 200 Kbps to 1 Mbps

3.2. FGA-FEC in frame-rate and resolution adaptation

In addition to SNR adaptation, FGA-FEC can do spatial and temporal adaptation as well. We set up a channel between node2 and node3 to test the adaptation capability of FGA-FEC and MD-FEC, where the channel BER and bandwidth changes over time as in Fig. 7.

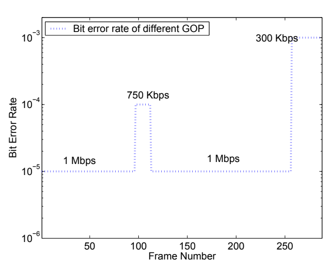
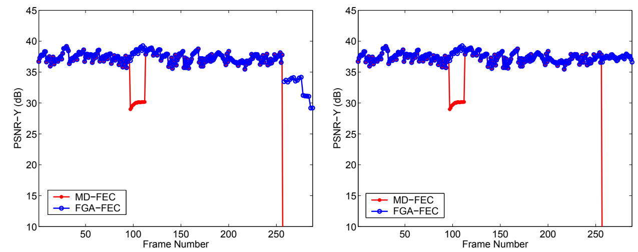


Fig. 7. Channel conditions between node2 and node3

Again, we sent the Section 3.1 encoded sequence to the receiver. At the bad condition (1×10^{-4} /750 Kbps), both FGA-FEC and MD-FEC use SNR adaptation. At the very bad channel state (1×10^{-3} /300 Kbps), FGA-FEC adaptation first does SNR adaptation, however, since this alone cannot satisfy the assumed user requirement, our algorithm further does frame-rate adaptation by 2 (Fig. 8(a)) and/or resolution adaptation by 2×2 (Fig.8(b)), implemented by packet shortening at a fine-grained block level (block size is 1 byte). Since MD-FEC only drops packets in this very bad condition, even the video base layer cannot go through the channel so that no video is decoded for the last two GOPs.



(a) Frame rate (b) Resolution

Fig. 8. Adaptation to different network conditions by frame rate and resolution.

4. CONCLUSION

In this paper, we generalize FGA-FEC for embedded video bitstream protection and content adaptation over wireless channels and propose a fast search algorithm to assign the optimal product codes. Simulations show the efficiency for simultaneous content protection and adaptation.

5. ACKNOWLEDGEMENTS

The authors would like to thank Dr. Rohit Puri for the Lagrange optimizer code used in the MD-FEC simulation.

6. REFERENCES

[1] Y. Shan, I.V. Bajic, S. Kalyanaraman, and J.W. Woods, "Joint source-network error control coding for scalable overlay streaming," in *Proc. ICIP, IEEE*, Sept. 2005.

[2] P.G. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. on Comm.*, vol. 2, Dec. 1998.

[3] D. G. Sachs, R. Anand, and K. Ramchandran, "Wireless image transmission using multiple-description based concatenated codes," *Proc. VCIP, SPIE*, Jan. 2000.

[4] R. Puri and K. Ramchandran, "Multiple description coding using forward error correction codes," *Proc. 33rd Asilomar Conf. (ACSS)*, Oct 1999.

[5] Y. Shan, I.V. Bajic, S. Kalyanaraman, and J.W. Woods, "Scalable video streaming with fine grain adaptive forward error correction," *IEEE Trans. on CSVT*, Submitted, Dec. 2006, available http://networks.ecse.rpi.edu/ỹfshan/FGAFEC.pdf.