JOINT PRODUCT CODE OPTIMIZATION FOR SCALABLE MULTIMEDIA TRANSMISSION OVER WIRELESS CHANNELS

Vladimir Stanković, Raouf Hamzaoui

University of Konstanz
Department of Computer and Information Science
D-78457 Konstanz
stankovi,hamzaoui@fmi.uni-konstanz.de

Zixiang Xiong

Texas A &M University
College Station, TX 77843
zx@lena.tamu.edu

ABSTRACT

State-of-the-art systems for the transmission of images over wireless channels generate an embedded bitstream and protect it with a product code where the row code is a concatenation of an outer cyclic redundancy check (CRC) code and an inner rate-compatible punctured convolutional (RCPC) code, and the column code is a Reed-Solomon (RS) code. In previous works, the product code was optimized by searching for the best RS protection for each RCPC code rate. We present a local search algorithm that jointly optimizes the RS and the RCPC codes. Experimental results show that our algorithm provides an approximately optimal solution, while its time complexity is much lower than that of the previous works.

1. INTRODUCTION

The design of efficient communication systems for the progressive transmission of multimedia over wireless networks has recently attracted a lot of interest because of the increasing demands for Internet and mobile wireless applications.

In this paper, we consider two wireless channel models. The first one is a combination of a packet erasure channel and a binary symmetric channel (BSC). It can describe situations where packets of data from a wireline network are sent to receivers over a wireless connection. The second one is a flat-fading Rayleigh channel, which is a good model for mobile communication.

The most powerful transmission systems over such channels are due to Sherwood and Zeger [1] and Sachs, Anand, and Ramchandran [2]. Both systems use an embedded wavelet-based source code and a product channel code. The row code of the product code is a concatenation of an outer CRC code and an inner RCPC code, while its column code is a systematic RS code. Both systems use equal error protection along the rows and unequal error protection along the columns. But whereas the system of [1] puts the earliest symbols of the embedded bitstream in the first rows, the system of [2] puts these symbols in the first columns. As a consequence, the first system has a better progressive ability. On the other hand, the system of [2] offers a better reconstruction quality (see [2] for a comparison).

In [2], the product code was optimized by determining the best RS protection for each RCPC code rate. This is time-consuming when many RCPC code rates are allowed. In contrast, we provide a local search algorithm that jointly optimizes the RS and the RCPC code. Experimental results show that our peak signal-to-noise ratio (PSNR) results are comparable to the previous approach. However, our algorithm is significantly faster because it only needs to inspect a few RCPC code rates to reach a near-optimal solution.

In Section 2, we introduce our terminology and present previous work. Section 3 describes our fast product code optimization. Section 4 gives numerical results for the two above channels with the SPIHT [3], JPEG2000 [4], and 3D SPIHT [5] source coders.

2. NOTATIONS AND PREVIOUS WORK

Let $\mathcal{R} = \{r_1, \dots, r_m\}$ be the set of RCPC code rates with $r_1 < \dots < r_m$. Let N be the number of packets sent and L be the length (in symbols) of a packet. For $r_i \in \mathcal{R}$, we denote by $L(r_i)$ the sum of the number of source symbols and RS redundant symbols used in a packet protected by r_i. Thus, we have $L(r_i)$ source segments $S_1, \dots, S_{L(r_i)}$, where segment $S_j,\ 1 \leq j \leq L(r_i)$, consists of $m_j \in \{1, \dots, N\}$ source symbols that are protected by $f_j = N - m_j$ RS symbols (see Table 1).

					+	+	o	o	o
x	x				+	+	o	o	o
x	x				+	+	o	o	o
x	x	x	x		+	+	o	o	o
x	x	x	x	x	+	+	o	o	o

Table 1. Product code structure. There are $N = 5$ packets of $L = 10$ symbols each. x denotes an RS redundant symbol, + a CRC symbol, o an RCPC redundant symbol, and an empty cell a source symbol.

The N packets are sent over a wireless channel. Each received packet is decoded with the RCPC decoder. If the CRC detects an error, then the packet is considered to be lost (we suppose that all errors can be detected). Suppose now that n packets of N are lost, then the RS codes ensure that all segments that contain at most $N - n$ source symbols can be recovered. By adding the constraint $f_1 \geq f_2 \geq \dots \geq f_{L(r_i)}$, we guarantee that the receiver can decode at least the first j segments whenever at most f_j packets are lost. In the following, we denote by $\mathcal{F}_i$ the set of $L(r_i)$-tuples $(f_1, \dots, f_{L(r_i)})$ such that $f_1 \geq f_2 \geq \dots \geq f_{L(r_i)}$ and $f_j \in \{0, \dots, N-1\}$ for $j = 1, \dots, L(r_i)$. Moreover, we denote by $p_N(n)$ the probability that n packets of N are lost. For example, suppose that RCPC code rate r_i is used and suppose that the channel is modeled as the concatenation of a BSC and a packet erasure channel with packet loss probability q. Then the probability that a packet is considered to be lost is $q' = q + (1-q)p(r_i)$, where $p(r_i)$ is the probability that a packet protected with rate r_i cannot be correctly decoded with the RCPC decoder. In this case, $p_N(n) = \binom{N}{n} q'^n (1-q')^{N-n}$.

Let ϕ denote the operational distortion-rate function of the source coder and let X be the random variable whose value is the number of packets lost. For a given code rate r_i, a *distortion-optimal* $L(r_i)$-RS protection minimizes the expected distortion

$$E[d](F) = \sum_{k=0}^{L(r_i)} P_k(F)\phi(t_k) \qquad (1)$$

over all $F = (f_1, \ldots, f_{L(r_i)}) \in \mathcal{F}_i$. Here $P_0(F) = P(X > f_1), P_k(F) = P(f_{k+1} < X \leq f_k)$ for $k = 1, \ldots, L(r_i) - 1$, $P_{L(r_i)}(F) = P(X \leq f_{L(r_i)}), t_0 = 0$, and $t_k = \sum_{j=1}^{k} m_j$ for $k = 1, \ldots, L(r_i)$. For $k = 1, \ldots, L(r_i) - 1$, we have $P_k(F) = 0$ if $f_k = f_{k+1}$ and $P_k(F) = \sum_{n=f_{k+1}+1}^{f_k} p_N(n)$, otherwise.

An optimal product code $(r_{i_k}, F^*_{i_k})$ is given by an RCPC code rate r_{i_k} and an $L(r_{i_k})$-RS protection $F^*_{i_k}$ that solve the minimization problem

$$\min_{r_i \in \mathcal{R}} \min_{F \in \mathcal{F}_i} \sum_{k=0}^{L(r_i)} P_k(F)\phi(t_k). \qquad (2)$$

Solving problem (2) by brute-force is impractical because the number of possible product codes is $\sum_{i=1}^{m} \binom{L(r_i)+N-1}{L(r_i)}$.

In [2], the authors use the Lagrange-based optimization algorithm of [6] to determine a near-optimal $L(r_i)$-RS protection for each $i = 1, \ldots, m$. The RS protection that yields the smallest expected distortion is selected. Even though the Lagrange-based optimization algorithm is fast, the overall optimization can be too expensive for real-time systems when the number of candidate channel code rates is large. One may accelerate this approach by replacing the Lagrange-based algorithm with a faster algorithm. For example, our iterative improvement algorithm [10] computes a near-optimal RS protection, and its complexity is much lower than that of all previous algorithms. This algorithm works as follows. By analogy with (1), define a *rate-optimal* RS protection as an RS protection scheme that maximizes the expected number of correctly received source symbols. That is, a rate-optimal $L(r_i)$-RS protection is a solution to the problem

$$\max_{F \in \mathcal{F}_i} E[r](F) = \max_{F \in \mathcal{F}_i} \sum_{k=0}^{L(r_i)} P_k(F)t_k. \qquad (3)$$

Then, as in [9], we obtain the following result.

Proposition 1 *Let ϕ be the operational distortion-rate function of the source coder and let $r_i \in \mathcal{R}$. Suppose that ϕ is nonincreasing (resp. nondecreasing) and convex (resp. concave). Let T^* be a distortion-optimal $L(r_i)$-RS protection and let R^* be a rate-optimal $L(r_i)$-RS protection. Let $V(T)$ denote the number of source symbols protected with T. Then we have*
(i) $E[d](T^*) \geq \phi(E[r](R^*))$.
(ii) $V(T^*) \leq V(R^*)$, and the inequality is strict if T^* is not rate optimal.

Proposition 1 (i) gives an easily computable lower bound on the performance of a distortion-optimal protection. Proposition 1 (ii) states that for a fixed length $L(r_i)$, a distortion-optimal protection is stronger than a rate-optimal one.

The local search algorithm of [10] first computes a rate-optimal protection. This is straightforward because

$$E[r](f_1, \ldots, f_{L(r_i)}) = \sum_{j=1}^{L(r_i)} m_j \sum_{i=0}^{f_j} p_N(i). \qquad (4)$$

Thus, a rate-optimal solution is the equal erasure protection $(f_r, \ldots, f_r)$, where

$$f_r = \arg \max_{i=0,\ldots,N-1} (N-i) \sum_{n=0}^{i} p_N(n).$$

Then the algorithm searches for the best candidate in its neighborhood. If this candidate is better than the current solution, we adopt it and repeat the search from the new solution. Otherwise, we stop. In accordance with Proposition 1 (ii), a neighborhood of a solution is restricted to solutions that provide a stronger protection.

Definition 1 *Let $F = (f_1, \ldots, f_{L(r_i)}) \in \mathcal{F}_i$. The neighborhood of F consists of the solutions of the form $(f_1 + 1, f_2, \ldots, f_{L(r_i)}), (f_1 + 1, f_2 + 1, \ldots, f_{L(r_i)}), \ldots, (f_1 + 1, f_2 + 1, \ldots, f_{L(r_i)-1} + 1, f_{L(r_i)} + 1)$ that are included in $\mathcal{F}_i$.*

When the convexity assumption of ϕ is severely violated, it may be advantageous to determine our local search solution by using a piecewise affine approximation of ϕ (see [10]).

3. FAST JOINT OPTIMIZATION

In this section, we present a fast method that finds an approximately optimal solution to problem (2). In contrast to [2], we do not try to minimize (1) for each RCPC code rate. We observe that the total number of redundant protection symbols (RS and RCPC) for the product code corresponding to the distortion-optimal RS protection with the smallest expected distortion (among all RCPC code rates) is greater than that corresponding to the rate-optimal RS protection with the largest expected rate. This result, which extends Proposition 1 (ii), can be proved in a similar way. We propose therefore to start with the rate-optimal RS protection that gives the largest expected rate and try to improve the associated product code by progressively increasing the number of protection symbols. This is done by alternately applying the local search algorithm of [10] and decreasing the RCPC code rate. We also exploit the fact that if S is our current RS protection, then one can exclude all RCPC code rates for which the lower bound of Proposition 1 (i) is greater than $E[d](S)$. Indeed, the distortion-optimal RS protection corresponding to one such code rate cannot be better than S. Before giving our algorithm, we need a definition.

Definition 2 *Let $r_j < r_i$ be two codes rates in $\mathcal{R}$. Let $F_i = (f_1, \ldots, f_{L(r_j)}, \ldots, f_{L(r_i)}) \in \mathcal{F}_i$ be an $L(r_i)$-RS protection. Then the $L(r_j)$-RS protection $F_j = (f_1, \ldots, f_{L(r_j)}) \in \mathcal{F}_j$ is called a child of F_i in $\mathcal{F}_j$.*

The following algorithm uses the above results to compute an approximately overall distortion-optimal RS protection S^*.

1. Set $n = 0$ and $\mathcal{R}_n = \mathcal{R}$. For each code rate $r_i \in \mathcal{R}_n$, compute a rate-optimal RS protection F_i.
2. Determine $r_{k_n} = \arg \max_{r_i \in \mathcal{R}_n} E[r](F_i)$.
3. Apply the iterative improvement algorithm of [10] to F_{k_n}. This gives a solution S_{k_n}. Set $S^* = S_{k_n}$.

4. Find $\mathcal{R}_{n+1}$, the set of rates $r_i \in \mathcal{R}_n$ such that $i < k_n$ and $\phi(E[r](F_i)) > E[d](S^*)$. If $\mathcal{R}_{n+1} = \emptyset$, stop.
5. Let $r_{k_{n+1}}$ be the highest code rate in $\mathcal{R}_{n+1}$. Apply the iterative improvement algorithm of [10] to the child of S_{k_n} in $\mathcal{F}_{k_{n+1}}$. This gives a solution $S_{k_{n+1}}$. If $E[d](S_{k_{n+1}}) < E[d](S^*)$, set $S^* = S_{k_{n+1}}$.
6. Set $n = n + 1$ and go to Step 4.

In the worst case, our algorithm computes for each $r_i \in \mathcal{R}$ $(N-1)L(r_i) + 1$ times the cost function (1).

4. EXPERIMENTAL RESULTS

We compared the performance of two optimization methods. The first method searches for an optimal RS protection for each RCPC code rate and selects the one with the lowest expected distortion [2]. For a given RCPC code rate, the RS protection was determined with the Lagrange-based algorithm of [6], the algorithm of [7], and the two algorithms of [8] denoted in the following by Da and Db. Note that the product code is optimal when Da is used. The second method is the algorithm of Section 3. Because the goal was to maximize the expected PSNR, the objective function was $\sum_{k=0}^{L(r_i)} P_k(F)PSNR(t_k)$, where $PSNR(t_k)$ is the PSNR corresponding to the number of source symbols t_k.

We used a 16-CRC code with generator polynomial 0x15935. The generator polynomials of the RCPC were $(0117, 0127, 0155, 0171)$, the mother code rate was 1/4, and the puncturing rate was 8. Thus, the set of RCPC rates was $\{8/9, 8/10, \ldots, 8/32\}$. The decoding of the RCPC code was done with a list Viterbi algorithm where the maximum number of candidate paths was 100.

We first provide results for a channel modeled as a concatenation of a BSC and a packet erasure channel. The bit error rate of the BSC was 0.1, and the probability of a packet loss in the erasure channel was $q = 0.05$. We used only 10 RCPC code rates because the other 14 can be excluded a priori. Indeed, the probability of decoding error of many code rates was zero. Therefore only the highest one was kept. Other code rates were not useful because their residual bit error rate was greater than 0.1.

All programs were run on a PC with a Windows operating system having an AMD Athlon XP 1600+ 1400 MHz processor with a main memory size of 1 Gbyte. Table 2 and Table 3 show the PSNR in dB and the time in seconds versus the number of packets N for the SPIHT and JPEG2000 bitstreams of the 8 bits per pixel 512 × 512 Lenna. To generate the JPEG2000 bitstream, we used the Kakadu C++ implementation of [4] with the default settings. The length of a packet was equal to 48 bytes, including one byte for the header. Table 4 shows results for the 3D SPIHT bitstream of the Y component of the 176 × 144 Foreman video sequence. Here the packet length was 1000 bytes, and only nine RCPC code rates were used. Table 5 shows results for a Rayleigh channel where the average signal-to-noise ratio (SNR) was 10 dB and the normalized Doppler spread f_D was 10^{-5}. The packet length was as above. The probability function $p_N(n)$ was simulated with Jakes's method [11]. Here all 24 RCPC code rates were used.

Because Da needs to store N^2L^2 floating point numbers and N^2L^2 integers, it could not be used when N or L was large. Moreover, its running time was unacceptable. So we only show the PSNR. The memory requirements of Db are smaller than those of Da, N^2L floating point numbers and N^2L integers, but still too high for large L.

In all cases, the average PSNR performance of our solution was almost optimal. Moreover, our algorithm was significantly faster than the previous approach. For example, our algorithm was up to 18 times faster for the Rayleigh channel. The bound criterion of Proposition 1 was very helpful. For the Rayleigh channel, this criterion excluded 19 code rates for SPIHT and JPEG2000 and 22 code rates for 3D SPIHT. In all experiments, the code rate selected by our algorithm was either the one corresponding to the best rate-optimal solution (Step 2) or the next tested one.

5. CONCLUSION

We proposed a fast joint optimization technique that provides a near-optimal product code solution for wireless channels. Our technique is much faster than the approach in [2] for two reasons. First, many RCPC code rates are eliminated very quickly. Second, the local search algorithm [10] used to improve an RS protection for a fixed code rate is faster than the previous algorithms [6, 7, 8].

Acknowledgments. We thank S. Dumitrescu, X. Wu, Z. Wang, D. Sachs, R. Anand, and K. Ramchandran for providing us with the C-codes of [8, 6, 2]. We also thank A. Mohr, R. Ladner, and E. Riskin for making the C-code of algorithm [7] available.

6. REFERENCES

[1] Sherwood, P. G., Zeger, K., *Error protection for progressive image transmission over memoryless and fading channels,* IEEE Trans. on Communications 46,12 (1998) 1555–1559.

[2] Sachs, D. G., Anand, R., Ramchandran, K., *Wireless image transmission using multiple-description based concatenated code*, in *Proc. SPIE'00,* vol. 3974, pp. 300–311, April 2000.

[3] Said, A., Pearlman, W. A., *A new fast and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits and Systems for Video Technology,* vol. 6, pp. 243–250, June 1996.

[4] Taubman, D., Marcellin, M., *JPEG2000: Image Compression Fundamentals, Standards, and Practice,* Kluwer, 2002.

[5] Kim, B.-J., Xiong, Z., Pearlman, W. A., *Low bit-rate scalable video coding with 3D set partitioning in hierarchical trees (3D SPIHT),* IEEE Trans. Circuits and Systems for Video Tech. vol. 10, pp. 1365–1374, Dec. 2000.

[6] Puri, R., Ramchandran, K., *Multiple description coding using forward error correction codes,* in: *Proc. 33rd Asilomar Conf. on Signals and Systems,* Pacific Grove, CA, Oct. 1999.

[7] Mohr, A., Ladner, R., Riskin, E., *Approximately optimal assignment for unequal loss protection,* in: *Proc. ICIP-2000,* vol. 1, pp. 367–370, Vancouver, Sept. 2000.

[8] Dumitrescu, S., Wu, X., Wang, Z., *Globally optimal uneven error-protected packetization of scalable code streams, Proc. DCC'02*, pp. 73–82, Snowbird, UT, April 2002.

[9] Hamzaoui, R., Stanković, V., Xiong, Z., *Rate-based versus distortion-based optimal joint source-channel coding, Proc. DCC'02*, pp. 63–72, Snowbird, UT, April 2002.

[10] Stanković, V., Hamzaoui, R., Xiong, Z., *Packet loss protection of embedded data with fast local search,* to appear in: *Proc. IEEE ICIP-02*, Rochester, NY, Sept. 2002.

[11] Jakes, W. C., *Microwave Mobile Communications,* Wiley, 1974.

	Da [8]	Db [8]		P [6]		M [7]		LS	
N	PSNR	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
100	26.16	26.14	0.20	26.14	0.04	26.15	0.06	26.14	0.01
200	28.53	28.53	0.45	28.50	0.07	28.51	0.19	28.51	0.02
300	30.12	30.12	0.70	30.08	0.11	30.10	0.42	30.08	0.03
400	31.30	31.30	0.96	31.26	0.15	31.29	0.76	31.27	0.04
500	32.28	32.28	1.19	32.24	0.21	32.26	1.25	32.27	0.06
600	33.07	33.07	1.52	33.03	0.27	33.04	1.96	33.03	0.06
800	34.33	34.31	2.19	34.28	0.43	34.31	3.85	34.26	0.10
1000				35.28	0.58	35.32	6.66	35.31	0.13

Table 2. CPU time in seconds and expected PSNR in dB for the SPIHT bitstream of the 512×512 Lenna image. The results are given for N packets of $L = 48$ bytes each. One byte is used for the header. The packet mean loss rate of the erasure channel is 0.05 and the bit error rate of the BSC is 0.1. The abbreviations Da, Db, P, and M denote the approach of [2] when the RS protection for each RCPC code rate was determined with the optimal algorithm of [8], algorithm B of [8], the algorithm of [6], and the algorithm of [7], respectively. LS is our local search algorithm (see Section 3).

	Da [8]	Db [8]		P [6]		M [7]		LS+PA	
N	PSNR	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
100	25.80	25.53	0.23	25.77	0.03	25.78	0.05	25.74	0.01
200	28.27	28.27	0.45	28.14	0.06	28.21	0.13	28.18	0.02
300	29.91	29.91	0.74	29.87	0.11	29.85	0.25	29.88	0.03
400	31.17	31.17	1.05	31.13	0.15	31.11	0.46	31.11	0.05
500	32.18	32.18	1.33	32.13	0.20	32.17	0.75	32.15	0.05
600	32.94	32.94	1.64	32.89	0.26	32.89	1.12	32.87	0.07
800	34.28	34.26	2.38	34.23	0.38	34.21	2.23	34.27	0.09
1000				35.25	0.52	35.29	3.85	35.30	0.14

Table 3. CPU time in seconds and expected PSNR in dB for the JPEG2000 bitstream of the 512×512 Lenna image. The results are given for N packets of $L = 48$ bytes each. One byte is used for the header. The packet mean loss rate of the erasure channel is 0.05 and the bit error rate of the BSC is 0.1. LS+PA means that our solution was found by applying the local search algorithm to the piecewise affine function associated to the PSNR-rate curve.

	P [6]		M [7]		LS		LS+PA	
N	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
100	24.79	0.13	24.80	0.22	24.75	0.04	24.75	0.05
200	26.80	0.41	26.80	0.48	26.79	0.09	26.79	0.11
300	28.05	0.77	28.04	0.75	28.06	0.14	28.06	0.16
400	29.04	1.37	29.04	1.06	28.95	0.15	28.98	0.19
500	29.76	2.19	29.76	1.52	29.68	0.17	29.70	0.26
600	30.47	3.07	30.47	2.03	30.38	0.18	30.40	0.30
800	31.61	4.99	31.61	3.41	31.52	0.27	31.56	0.58
1000	32.58	7.24	32.58	4.98	32.45	0.33	32.51	0.70

Table 4. CPU time in seconds and expected PSNR in dB for the 3D SPIHT bitstream of the Foreman sequence. The results are given for N packets of $L = 1000$ bytes each. The packet mean loss rate of the erasure channel is 0.05 and the bit error rate of the BSC is 0.1.

Source		Da [8]	Db [8]		P [6]		M [7]		LS	
coder	N	PSNR	PSNR	Time	PSNR	Time	PSNR	Time	PSNR	Time
SPIHT	200	31.01	31.01	2.14	30.98	0.15	31.00	0.45	30.95	0.03
JPEG2000	200	30.83	30.81	3.10	30.78	0.13	30.79	0.34	30.75	0.02
3D SPIHT	200				29.93	1.59	29.93	1.44	29.93	0.08

Table 5. CPU time in seconds and expected PSNR in dB for a Rayleigh fading channel with SNR=10 dB and $f_D = 10^{-5}$. The results are given for N packets of 48 bytes each for SPIHT and JPEG2000 (one byte is used for the header), and 1000 bytes each for 3D SPIHT.