

# Fast Joint Source-channel Coding Algorithms for Internet/wireless Multimedia

Raouf Hamzaoui, Vladimir Stanković, and Zixiang Xiong

**Abstract** - We consider joint source-channel coding for Internet and wireless multimedia applications. Embedded source bitstreams are protected against noise in both binary symmetric channels (BSC) and packet erasure channels. For each channel, a local search joint source-channel coding (JSCC) algorithm is proposed to search for the optimal unequal error protection (UEP) scheme. Each algorithm starts with a linear-complexity step that maximizes the number of correctly decoded source bits, followed by a quick local refinement of the resulting UEP solution to minimize the average distortion. Experiments for both binary symmetric channels and packet erasure channels with the SPIHT, JPEG2000 and 3D SPIHT coders show that our local search algorithms are near-optimal in performance, whereas they are orders of magnitude faster than the best previous solutions.

## I. INTRODUCTION

Internet and wireless multimedia applications involve source coding, channel coding, and joint source-channel coding. While international standards like JPEG2000 [1] and MPEG-4 are becoming the state-of-the-art of multimedia compression, new near-capacity codes (e.g., turbo codes and low-density parity-check codes) are being developed for channel coding in the mean time. Since Sherwood and Zeger [2] showed the effectiveness of using the SPIHT coder [4] as a source coder and the concatenation of a cyclic-redundancy check (CRC) coder and a rate compatible punctured convolutional (RCPC) coder as a channel coder, many works have been done on JSCC for multimedia transmission over both BSC's and packet erasure channels. Recent optimal and fast near-optimal JSCC algorithms [3-12] have brought JSCC research into its maturity.

This paper builds upon our latest research on fast JSCC algorithms presented in [11], [13]. Throughout the paper, we assume that embedded bitstreams are generated by coders like JPEG2000 [1], SPIHT, and 3D SPIHT [14], although standard scalable audio/video coders [15], [16] can also be used. For channel coding, we used

R. Hamzaoui and V. Stanković are with the Institut für Informatik, Universität Leipzig, Augustusplatz 10-11, 04109 Leipzig, Germany. Z. Xiong is with the Dept of Electrical Engineering, Texas A & M University, College Station, TX 77843, USA.

rate-compatible punctured turbo (RCPT) codes [17] and Reed-Solomon codes [18] for a BSC and a packet erasure channel, respectively. These channel models are suitable for wireless and Internet multimedia applications. Extension of our work to wireless fading channels is currently studied.

Because the source code is embedded, it is reasonable to assume that when an error occurs to the source bitstream, all source code bits beyond the error have to be discarded even though they are correctly decoded. Thus, from the error protection point of view, we consider the first bit of the source bitstream to be the most important one, the second bit to be the second in importance, etc. This naturally leads to unequal error protection in JSCC design of Internet/wireless multimedia systems. The performance of these systems may be measured by the expected distortion or by the expected number of correctly received source bits subject to a target transmission rate. Whereas a rate-based optimal solution can be found in linear time for both the BSC [5], [10] and the packet erasure channel [13], the computation of a distortion-based optimal solution is prohibitive.

Realizing that the low-complexity rate-based optimal solution should provide a good approximation of the distortion-based optimal solution, we are able to quantify the approximation error via a lower bound on the expected distortion of a distortion-based optimal solution that depends only on a rate-based optimal solution. We further show that a distortion-based optimal solution provides a stronger error protection than a rate-based optimal solution and exploit this result to reduce the time complexity of the distortion-based optimization. Finally we are able to come up with fast iterative improvement algorithms that start from a rate-based optimal solution and converge to a local minimum of the expected distortion. Experiments with different source coders and types of channels show that our approach delivers a better or almost the same performance as the best previous algorithms, while having much lower complexity.

## II. THE BSC CASE

We consider a system that protects an embedded source bitstream using  $m$  channel codes  $c_1, \dots, c_m$  with

code rates  $r_1 < r_2 < \dots < r_m$  and probabilities of a packet decoding error  $p(r_1) < \dots < p(r_m) < 1$ . We assume that the channel codes have error detection and error protection capabilities. The channel code length is fixed at  $L$  for all rates, and there are totally  $N$  packets to transmit after JSCC. If the decoder detects an error, then the transmission stops and the source is reconstructed from the correctly received source bits. We assume that all errors can be detected.

Let  $\mathcal{R} = \{r_1, \dots, r_m\}$ . An  $N$ -packet error protection scheme (EPS)  $R = (r_{k_1}, \dots, r_{k_N}) \in \mathcal{R}^N$  assigns channel code rate  $r_{k_i}$  to the  $i$ -th packet.  $P_0(R) = p(r_{k_1})$  is the probability of an error in the first packet;  $P_i(R) = \prod_{j=1}^i (1 - p(r_{k_j})) p(r_{k_{i+1}})$  is the probability that no errors occur in the first  $i$  ( $1 \leq i \leq N - 1$ ) packets with an error in the next one and  $P_N(R) = \prod_{j=1}^N (1 - p(r_{k_j}))$  is the probability that all  $N$  packets are correctly received. The expected distortion of such an  $N$ -packet EPS is given by

$$E_N[d](r_{k_1}, \dots, r_{k_N}) = \sum_{i=0}^N P_i(R) d_i(R), \quad (1)$$

where  $d_0(R) = d_0$  is the source variance and  $d_i(R)$  for  $i \geq 1$  is the reconstruction error using the first  $i$  packets. Since the number of possible  $N$ -packet EPS's is equal to  $m^N$ , brute force cannot be used to minimize (1) when  $N$  is large. An approximate minimization based on a Viterbi algorithm was proposed in [8]. The algorithm has  $O(N^2)$  complexity for some particular sets of channel code rates.

However, if we replace the minimization of (1) by the maximization of the expected number of correctly received source bits

$$E_N[r](r_{k_1}, \dots, r_{k_N}) = \sum_{i=0}^N P_i(R) V_i(R), \quad (2)$$

where  $V_0(R) = 0$  and  $V_i(R) = \sum_{j=1}^i v(r_{k_j})$  for  $i \geq 1$ , with  $v(r_{k_j}) = Lr_{k_j}$  being the number of source bits in the  $j$ th packet, then an optimal solution can be computed in  $O(N)$  time [5], [10]. We henceforth say that an EPS that minimizes (1) is *distortion optimal* and that an EPS that maximizes (2) is *rate optimal*. Furthermore, we call a distortion-optimal solution with  $r_{k_1} \leq \dots \leq r_{k_N}$  *constrained distortion optimal*. This constraint reduces the number of candidates from  $m^N$  to  $\binom{m+N-1}{N}$ .

Even though a rate-optimal solution has very low complexity, experiments for the SPIHT source coder and a CRC-RCPC channel coder show that the average PSNR performance given by such a solution suffers little loss, compared to that from a distortion-optimal solution [9]. In the next section, we give an upper bound on the error

made when a rate-optimal solution is used instead of a distortion-optimal one [11].

### A. Error Bound

*Proposition 1:* Assume that the operational distortion-rate (D-R) function  $\phi$  of the source coder is nonincreasing and convex. Let  $T^*$  be a distortion-optimal  $N$ -packet EPS and  $R^*$  be a rate-optimal  $N$ -packet EPS. Then we have

- (i)  $E_N[d](T^*) \geq \phi(E_N[r](R^*))$ .
- (ii)  $V_N(T^*) \leq V_N(R^*)$  and the inequality is strict if  $T^*$  is not rate optimal.

Thus, the error  $E_N[d](R^*) - E_N[d](T^*)$  is bounded by the easily computable error  $E_N[d](R^*) - \phi(E_N[r](R^*))$ . Indeed,  $R^*$  can be computed in  $O(N)$  time with the algorithm of [10]. On the other hand, Proposition 1 (ii) allows us to reduce the complexity of the distortion-based optimization by discarding many schemes. This leads to the following local search algorithm that rapidly finds a local minimum of (1).

### B. The Local Search Algorithm

Let  $\mathcal{R}$  be the set of code rates  $\{r_1, \dots, r_m\}$  and  $R \in \mathcal{R}^N$  be an  $N$ -packet EPS with nondecreasing rates. Then the *neighborhood* of  $R$  is the set of EPS's  $S \in \mathcal{R}^N$  such that (a)  $V_N(S) < V_N(R)$ , (b)  $R$  and  $S$  differ in only one code rate and (c) The rates of  $S$  are nondecreasing. We also define the  $k$ -th neighbor of  $R$  as the one whose new code rate is the  $k$ -th largest one.

The local search algorithm works by iterative improvement. We start from a rate-optimal solution. We then consider the first neighbor of the current solution. If the expected distortion of this neighbor is smaller than that of the current solution, then we update the current solution; otherwise we consider the next neighbor and repeat the procedure. A pseudo-code for the local search algorithm is given below:

1. Set  $k = l = 1$  and  $n = 0$ . Use the algorithm of [10] to compute a rate-optimal  $N$ -packet EPS  $R_n$ .
2. Let  $r$  be the  $k$ -th highest rate used by  $R_n$  and  $j$  the index of the first packet that  $R_n$  protects with  $r$ . If  $r = r_1$ , stop. Otherwise, let  $r_c \in \mathcal{R}$  be the  $l$ -th highest rate smaller than  $r$  and define  $R_c$  to be the EPS obtained from  $R_n$  by protecting packet  $j$  with  $r_c$ .
3. If  $E_N[d](R_c) < E_N[d](R_n)$ , set  $R_{n+1} = R_c$ ,  $n = n + 1$ ,  $k = 1$ ,  $l = 1$ , and go to Step 2.
4. If  $j \neq 1$  and  $r_c$  is greater than the rate of packet  $j - 1$ , set  $l = l + 1$ . If  $j \neq 1$  and  $r_c$  is equal to the rate of packet  $j - 1$ , set  $l = 1$  and  $k = k + 1$ . If  $j = 1$  and  $r_c \neq r_1$ , set  $l = l + 1$ . If  $j = 1$  and  $r_c = r_1$ , stop.
5. Go to Step 2.

### C. Results

Using SPIHT and JPEG2000 coded bitstreams of the  $512 \times 512$  gray-scale Lenna image, we compared the mean-squared error (MSE) performance and the time complexity of a constrained MSE-optimal solution, the solution computed by the Viterbi algorithm of [8], and the solution of the local search algorithm. In all experiments,  $L = 2048$  bits and  $N = 512^2 R_T / L$ , where  $R_T$  is the transmission rate in bits-per-pixel (bpp). The packets were sent over a binary symmetric channel with 10% bit error rate and protected with a concatenation of a CRC-32 coder and an RCPT coder consisting of two identical recursive systematic convolutional encoders with memory length 4 and generators (31, 27) (octal). The rate of the mother code was 1/3 and the puncturing rate was 20, yielding 41 possible channel code rates. We used iterative maximum a posteriori decoding, which was stopped if no correct sequence was found after 20 iterations. The probability of a packet decoding error for each code rate was computed with 50000 Monte Carlo simulations. Only the channel code rates that may be selected by a rate-optimal EPS were considered in MSE-minimization.

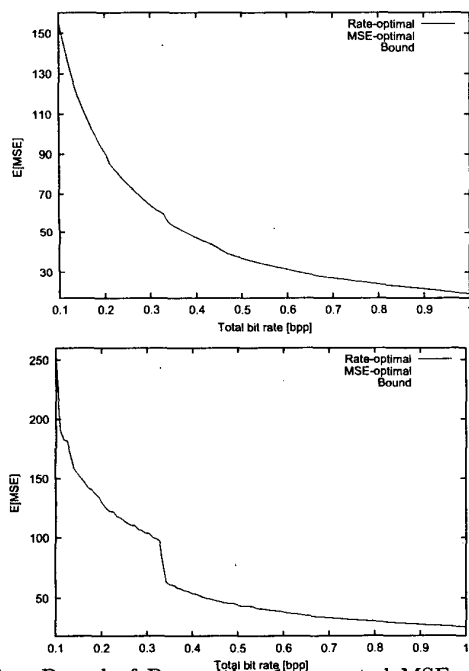


Fig. 1. Bound of Proposition 1, expected MSE of a constrained MSE-optimal solution, and expected MSE of a rate-optimal solution for SPIHT (top) and JPEG2000 (bottom).

Figure 1 shows the bound predicted by Proposition 1 (i), the expected MSE performance of a constrained

MSE-optimal solution and that of a rate-optimal solution. Because the difference between the lower bound and the performance of a constrained MSE-optimal solution is very small, we conclude that a constrained MSE-optimal solution is near MSE-optimal. The rate-optimal solution was a good approximation to the constrained MSE-optimal solution, although the approximation error was larger for JPEG2000.

Figure 2 (top) shows for the SPIHT coder the difference in expected MSE between a solution found by rate optimization and a constrained MSE-optimal solution, together with the difference between a solution found by the local search algorithm and a constrained MSE-optimal solution. Figure 2 (bottom) presents similar results for JPEG2000 (for clarity of display, we give the difference in PSNR of the expected MSE.) These results show that except for one transmission rate where the initial rate-optimal solution was poor, the solution found by local search was very close to the optimal one.

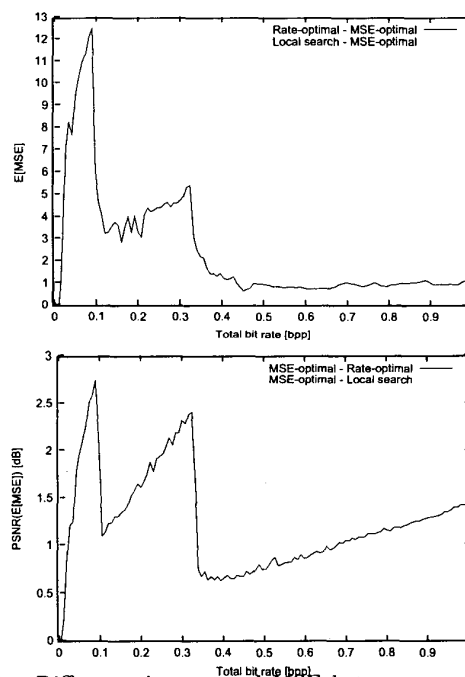


Fig. 2. Difference in expected MSE between, respectively, a rate-optimal solution, a local search solution and a constrained MSE-optimal solution for SPIHT (top) and JPEG2000 (bottom).

Table I compares for SPIHT the CPU time and the expected MSE of, respectively, a constrained MSE-optimal solution, a solution found by the Viterbi algorithm of [8] and a solution computed with our local search algorithm. The CPU time was measured on an SGI Origin 200 hav-

ing four 270 MHz MIPS R12000 processors and a main memory size of 1536 Megabytes. The local search algorithm found a solution with about the same quality as the constrained distortion-optimal solution. Moreover, the time complexity of local search was much lower than that of the Viterbi algorithm. We obtained similar results for JPEG2000.

TABLE I  
EXPECTED MSE AND CPU TIME IN SECONDS AT VARIOUS TRANSMISSION RATES FOR A SOLUTION FOUND BY THE LOCAL SEARCH ALGORITHM (LS) AND ONE OBTAINED WITH THE VITERBI ALGORITHM [8]. THE COLUMN LABELED CDO GIVES THE EXPECTED MSE OF THE CONSTRAINED MSE-OPTIMAL SOLUTION.

$R_T$ (bpp)	CDO MSE	Viterbi		LS	
		MSE	T(s)	MSE	T(s)
0.25	70.42	70.42	6.85	70.52	0.2
0.5	36.27	36.34	72.42	36.28	0.22
0.75	24.40	24.53	288.57	24.44	0.23
1.0	18.02	18.13	777.68	18.03	0.30

### III. THE PACKET ERASURE CHANNEL CASE

We assume that after JSCC, an embedded bitstream is sent in  $N$  packets over a packet erasure channel. Each packet has a fixed length of  $L$  symbols. Several researchers proposed efficient UEP solutions for such systems. In [6], Puri and Ramchandran proposed a Lagrange multiplier-based algorithm. In [7], Mohr, Ladner and Riskin proposed a fast algorithm based on the computation of the convex-hull of the operational D-R function of the source coder. The algorithm has a worst case time complexity of  $O(hN \log N)$ , where  $h$  is the number of vertices of the convex hull of the operational D-R function. Recently, Dumitrescu, Wu, and Wang [12] proposed an  $O(N^2L^2)$  algorithm that finds an optimal solution in the general case. They also gave an  $O(NL^2)$  algorithm that is close to optimal in the general case and optimal if the operational D-R function is convex and the probability of losing  $n$  packets decreases with  $n$ .

#### A. Problem Statement

We introduce our terminology and state the packet loss protection problem as a combinatorial optimization problem. Our notation closely follows that of [12].

The JSCC systems in [6], [7] build  $L$  source segments  $S_1, \dots, S_L$  of  $m_i \in \{1, \dots, N\}$  symbols each and protects each segment with an  $(N, m_i)$  systematic Reed-Solomon code of maximal distance. For each  $i = 1, \dots, L$ , let  $f_i = N - m_i$  denote the number of RS redundancy symbols that protect segment  $S_i$ . If  $n$  packets of  $N$  are lost,

then the RS codes ensure that all segments that contain at most  $N - n$  source symbols can be recovered. Thus, by adding the constraint that  $f_1 \geq f_2 \geq \dots \geq f_L$ , if at most  $f_i$  packets are lost, then the receiver can decode at least the first  $i$  segments. In the following, we denote by  $\mathcal{F}$  the set of  $L$ -tuples  $(f_1, \dots, f_L)$  such that  $f_i \in \{0, \dots, N - 1\}$  for  $i = 1, \dots, L$  and  $f_1 \geq f_2 \geq \dots \geq f_L$ . Let  $p_N(n)$  denote the probability of losing exactly  $n$  packets of  $N$  and let  $c_N(k) = \sum_{n=0}^k p_N(n)$ ,  $k = 0, \dots, N$ . Then  $c_N(f_i)$  is the probability that the receiver correctly recovers segment  $S_i$ . Let  $\phi$  denote the D-R function of the source coder and let  $X$  be the random variable whose value is the number of packets lost. An optimal packet loss protection for the system consists of finding a protection scheme  $F = (f_1, \dots, f_L) \in \mathcal{F}$  that minimizes the expected distortion

$$E_L[d](f_1, \dots, f_L) = \sum_{i=0}^L P_i(F) \phi(r_i), \quad (3)$$

where  $P_0(F) = P(X > f_1)$ ,  $P_i(F) = P(f_{i+1} < X \leq f_i)$  for  $i = 1, \dots, L - 1$ ,  $P_L(F) = P(X \leq f_L)$ ,  $r_0 = 0$ , and  $r_i = \sum_{k=1}^i m_k = iN - \sum_{k=1}^i f_k$  for  $i = 1, \dots, L$ . Note that for  $i = 1, \dots, L - 1$ , we have  $P_i(F) = 0$  if  $f_i = f_{i+1}$  and  $P_i(F) = \sum_{n=f_{i+1}+1}^{f_i} p_N(n)$ , otherwise.

#### B. Local Search Algorithm

In [13], we proposed a local search algorithm that is conceptually similar to the one described in Section 2.2 to find an approximately optimal solution to the problem of minimizing (3). It follows from [11] that if the operational D-R function of the source coder is convex, then the total number of protection symbols  $\sum_{i=1}^L f_i^*$  of a distortion-optimal solution  $(f_1^*, \dots, f_L^*)$  is greater than or equal to that of a rate-optimal solution. By rate-optimal solution, we mean a solution that maximizes the expected number of received source bits

$$E_L[r](f_1, \dots, f_L) = \sum_{i=0}^L P_i(F) r_i. \quad (4)$$

The local search algorithm in the packet erasure channel case again works by iterative improvement. We start at a rate-optimal solution and search 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. Note that the computation of a rate-optimal solution is straightforward in this case. Indeed, we have

$$E_L[r](f_1, \dots, f_L) = \sum_{i=1}^L P_i(F) \sum_{j=1}^i m_j \quad (5)$$

$$= \sum_{j=1}^L m_j \sum_{i=0}^{f_j} p_N(i). \quad (6)$$

Thus, by setting

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

(6) shows that a rate-optimal solution is  $(f_r, \dots, f_r)$ .

In the packet erasure channel case, we define the *neighborhood* of  $F$  to be the set of solutions of the form  $(f_1+1, f_2, \dots, f_L), (f_1+1, f_2+1, \dots, f_L), \dots, (f_1+1, f_2+1, \dots, f_{L-1}+1, f_L+1)$  that are included in  $\mathcal{F}$ . Again, in accordance with the result in [11], the neighborhood of a solution  $F$  is restricted to solutions that provide a stronger protection than  $F$ . For example, suppose that  $L=4$  and  $N=8$ . Then the neighbors of  $F=(6, 6, 5, 4)$  are the solutions  $(7, 6, 5, 4), (7, 7, 5, 4), (7, 7, 6, 4)$  and  $(7, 7, 6, 5)$ . Note that the solutions that do not belong to  $\mathcal{F}$  are not considered. For example, the set of neighbors of  $F=(7, 6, 5, 4)$  is empty.

In the worst case, our algorithm starts from the solution  $(0, \dots, 0)$  and stops at  $(N-1, f_2, \dots, f_L)$ . In this situation, we need  $L(N-1)+1$  computations and  $L(N-1)$  comparisons of the cost function (3), respectively. When we compute the cost function for the neighbors of a solution, we exploit the fact that two successive neighbors differ in only one segment. Thus, only two probabilities  $P_i$  have to be recomputed. For example, let  $F_1=(7, 6, 5, 4)$  and  $F_2=(7, 7, 5, 4)$ . Then  $P_0(F_2)=P_0(F_1), P_3(F_2)=P_3(F_1)$ , and  $P_4(F_2)=P_4(F_1)$ .

Because our local search algorithm exploits the assumption of the convexity of the operational D-R function of the source coder, it may provide unsatisfactory results when this assumption is severely violated. For example, we obtained much better results for the SPIHT coder than for JPEG2000. To overcome this problem, we propose for JPEG2000 to compute our solution by applying the local search algorithm to the piecewise affine function obtained by joining the points of the operational PSNR-rate curve at which the PSNR changes. An alternative is to use the convex hull of the operational D-R curve.

### C. Results

We compared the time complexity and the PSNR performance of our local search algorithm to those of algorithms in [7] and [12]. The algorithm (Mb) in [7] has a worst case time complexity of  $O(hN \log N)$ , where  $h$  is the number of vertices of the convex hull of the operational D-R function. The  $O(N^2L^2)$  algorithm in [12] finds an optimal solution in the general case. A second

$O(NL^2)$  algorithm (Db) in [12] is near-optimal in general and optimal if the operational D-R function is convex and the packet loss probability is monotonically decreasing.

Our goal was to maximize the expected PSNR. Thus, our objective function was  $\sum_{i=0}^L P_i(F)PSNR(r_i)$ , where  $PSNR(r_i)$  is the PSNR corresponding to source rate  $r_i$ . In all experiments, we had  $L=48$  bytes, and the packet loss probability function  $p_N(n)$  was exponentially decreasing in  $n$  with a mean loss rate of 0.2. All programs were run on a PC with a Windows operating system having an Intel Pentium III 550 MHz Processor with a main memory size of 128 Megabytes. Tables II and III show the PSNR in dB and the time in seconds vs. the number of packets  $N$  for, respectively, the SPIHT and JPEG2000 coded Lenna image. Table IV shows the results for the 3D SPIHT bitstream of the Y component of the QCIF Foreman video sequence.

TABLE II

CPU TIME IN SECONDS AND EXPECTED PSNR IN DB FOR THE SPIHT BITSTREAM OF THE  $512 \times 512$  LENA IMAGE.

$N$	Db [12]		Mb [7]		LS	
	PSNR	T(s)	PSNR	T(s)	PSNR	T(s)
100	28.27	0.5	28.26	0.3	28.26	0.01
200	30.99	1.02	30.97	0.4	30.93	0.02
300	32.65	1.6	32.64	0.61	32.60	0.03
400	33.81	2.23	33.87	0.91	33.84	0.03
600	34.57	3.63	35.6	2.05	35.58	0.05
800			36.81	3.82	36.79	0.07
1000			37.85	6.3	37.84	0.09

TABLE III

CPU TIME IN SECONDS AND EXPECTED PSNR IN DB FOR THE JPEG2000 BITSTREAM OF THE  $512 \times 512$  LENA IMAGE.

$N$	Db [12]		Mb [7]		LS	
	PSNR	T(s)	PSNR	T(s)	PSNR	T(s)
100	27.72	0.5	27.98	0.16	27.94	0.01
200	30.77	1.06	30.79	0.21	30.76	0.02
300	32.57	1.64	32.57	0.33	32.50	0.03
400	33.75	2.23	33.79	0.5	33.73	0.04
600	34.58	3.49	35.60	1.13	35.55	0.06
800			36.77	2.17	36.78	0.09
1000			37.73	3.73	37.69	0.12

Because the  $O(NL^2)$  algorithm in [12] needs to store  $N^2L$  floating point numbers and  $N^2L$  integers, our computer ran out of memory for relatively large  $N$ , and we were not able to report the results.

The tables show that our local search algorithm was always the fastest. The speed-up factor of our algorithm over Mb in [7] increased monotonically with  $N$ , reaching 70 for the SPIHT coder with  $N=1000$ . The PSNR

TABLE IV  
CPU TIME IN SECONDS AND EXPECTED PSNR IN DB FOR  
THE 3D-SPIHT BITSTREAM OF THE FOREMAN SEQUENCE.

$N$	Db [12]		Mb [7]		LS	
	PSNR	T (s)	PSNR	T (s)	PSNR	T(s)
400	21.90	2.22	21.92	0.5	21.92	0.02
600	22.87	3.49	23.06	0.6	23.04	0.03
800			23.80	0.81	23.77	0.05
1000			24.39	1.03	24.34	0.06
1200			24.94	1.51	24.92	0.09
1400			25.40	1.97	25.36	0.11
1600			25.80	2.72	25.73	0.12
1800			26.07	3.58	26.01	0.15

performance of our solution was similar to that of both Db and Mb. All three solutions were close to optimal.

We compared the solution of our local search algorithm to that of Mb for the 3D SPIHT bitstream of the Foreman sequence when, as in [7],  $L = 1000$  bytes,  $N = 811$  packets, and the packet mean loss was 0.05. On an SGI Origin 200 having four 270 MHz MIPS R12000 processors and a main memory size of 1536 Megabytes, Mb yielded a solution of average PSNR 38.35 dB in 6.14s, whereas our algorithm produced a solution of average PSNR 38.31 dB in 2.96s.

We did not implement the algorithm of [6], and the program was not available. However, the Lagrangian nature of this algorithm and the experimental results given in [6] indicate that our algorithm is faster.

#### IV. CONCLUSION

We proposed iterative improvement algorithms for UEP of embedded data in both BSC and packet erasure channels for Internet/wireless multimedia applications. Our algorithms provide state-of-the-art PSNR performance with lower complexity than that of the best previous solutions. Our algorithm are especially successful with embedded coders whose operational D-R function can be well approximated by a convex function. We believe that we have made contributions to the field of JSCC as our local search technique can be applied to many Internet/wireless multimedia system designs.

#### ACKNOWLEDGMENTS

We thank Michael Hiller for implementing the Viterbi algorithm of [8] and Brian Banister for fruitful discussions and for providing us with the brute-force algorithm that computes a constrained distortion-optimal solution. We are very grateful to Sorina Dumitrescu, Xiaolin Wu, and Zhe Wang for sending us their paper [12] and providing

us with the C-codes of their algorithms. We also thank Alexander Mohr, Richard Ladner, and Eve Riskin for making the C-code of algorithm [7] available.

#### References

- [1] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*, Kluwer, 2001.
- [2] P. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels", *IEEE Signal Processing Letters*, vol. 4, pp. 189-191, July 1997.
- [3] P. Sherwood and K. Zeger, "Error protection for progressive image transmission over memoryless and fading channels," *IEEE Trans. Communications*, vol. 46, pp. 1555-1559, December 1998.
- [4] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits and Systems for Video Tech.*, vol. 6, pp. 243-250, June 1996.
- [5] V. Chande and N. Farvardin, "Joint source-channel coding for progressive transmission of embedded source coders," *Proc. DCC'99*, pp. 52-61, Snowbird, UT, March 1999.
- [6] R. Puri and K. Ramchandran, "Multiple description coding using forward error correction codes," *Proc. 33rd Asilomar Conf. on Signals and Systems*, Pacific Grove, CA, October 1999.
- [7] A. Mohr, R. Ladner and E. Riskin, "Approximately optimal assignments for unequal error protection," *Proc. ICIP'00*, Vancouver, BC, Canada, September 2000.
- [8] B. Banister, B. Belzer and T. Fischer, "Robust image transmission using JPEG2000 and Turbo-codes," *Proc. ICIP'00*, vol. 1, pp. 375-378, Vancouver, BC, Canada, September 2000.
- [9] V. Chande and N. Farvardin, "Progressive transmission of images over memoryless channels," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 850-860, June 2000.
- [10] V. Stanković, R. Hamzaoui, and D. Saupe, "Fast algorithm for optimal error protection of embedded wavelet coders," *Proc. IEEE Workshop Multimedia Signal Proc.*, France, October 2001.
- [11] R. Hamzaoui, V. Stanković, and Z. Xiong, "Rate-based versus distortion-based optimal joint source-channel coding," *Proc. DCC'02*, Snowbird, UT, April 2002 (to appear).
- [12] S. Dumitrescu, X. Wu, and Z. Wang, "Globally optimal uneven error-protected packetization of scalable code streams," *Proc. DCC'02*, Snowbird, UT, April 2002 (to appear).
- [13] V. Stanković, R. Hamzaoui, and Z. Xiong, "Packet loss protection of embedded data with fast local search," submitted.
- [14] B. Kim, Z. Xiong, and W. Pearlman, "Very low bit-rate embedded video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. on Circuits and Systems for Video Tech.*, December 2000.
- [15] K. Brandenburg, O. Kunz, and A. Sugiyama, "MPEG-4 natural audio coding," *Signal Processing: Image communication*, vol. 15, pp. 423-444, January 2000.
- [16] H. Radha, M. van der Schaar, and Y. Chen, "The MPEG-4 fine-grained scalable video coding method for multimedia streaming over IP," *IEEE Trans. Multimedia*, vol. 3, pp. 53-68, March 2001.
- [17] D. Rowitch and L. Milstein, "Rate compatible punctured turbo (RCPT) codes in a hybrid FEC/ARQ system," *Proc. Globecom'97*, vol. 4., pp. 55-59, Phoenix, AZ, November 1997.
- [18] S. Wicker, *Error Control Systems for Digital Communications and Storage*, Prentice Hall, 1995.