

MAP DECODING OF ARITHMETIC CODES WITH A FORBIDDEN SYMBOL

¹Marco Grangetto, ²Pamela Cosman

¹grangetto@polito.it ²pcosman@code.ucsd.edu

¹ CERCOM - Center for Wireless Multimedia Communications

Politecnico di Torino, Dip. di Elettronica,

Corso Duca degli Abruzzi 24, 10129 Torino, Italy

² University of California, San Diego,

Dept. of Electrical and Computer Engineering,

9500 Gilman Drive, La Jolla, CA 92093-0407, USA

ABSTRACT

In this paper, a novel maximum *a posteriori* (MAP) technique for the decoding of arithmetic codes in the presence of transmission errors is presented. Arithmetic codes with a forbidden symbol and trellis search techniques are employed in order to estimate the best transmitted codeword. The viability of the proposed approach is demonstrated in the binary symmetric channel case in terms of both performance and decoding complexity. The results are compared with a traditional separated approach based on rate compatible convolutional codes. The MAP decoding is applied to the progressive transmission of SPIHT compressed images and competitive results in terms of average decoded quality are reported.

1. INTRODUCTION

Over the past decade, there has been great interest in the development of joint source/channel coding (JSCC) techniques. This is motivated by the impressive growth of personal mobile communications which aim to support ever more sophisticated services in hostile wireless environments. Channel bandwidth limitations require powerful source compression techniques, but also noisy tetherless channels severely impact the compressed data error sensitivity. In this context, JSCC has emerged as a viable approach to the problem.

JSCC techniques are based on the fact that in practical cases the source encoder is not able to completely decorrelate the input sequence; some implicit redundancy is still present in the compressed stream and can be properly exploited by the decoder. In [1, 2] the residual redundancy in the source encoder's output is represented with a Markov model, and is used as a form of implicit channel protection at the decoder side; exact and approximate maximum *a posteriori* (MAP) sequence estimators are proposed. Results are provided in the case of image transmission across the binary symmetric channel (BSC); the source coder is constituted by Huffman coding of neighboring pixel differences.

In this paper, we propose novel MAP techniques for decoding arithmetic codes in the presence of transmission errors. Arithmetic coding (AC) is known as one of the most powerful entropy coding techniques [3] but is extremely error sensitive as well. Unlike Huffman codes, AC has poor resynchronization capability, and a single bit error in the compressed stream can propagate all along the compressed block. Moreover, the residual redundancy in the compressed stream is usually negligible, preventing any MAP decoding attempt. Nevertheless, it is possible to perturb the arithmetic coder source model in order to keep some residual redundancy at the expense of compression efficiency. This idea was first introduced by Boyd *et al.* in [4] and extended in [5, 6] to provide continuous error detection during arithmetic decoding. The presence of known residual redundancy can be exploited for error correction as well. Some preliminary work can be found in [7], where the error correction is performed in the case of transmission over an AWGN channel; binary signalling with *null zone* soft decoding is employed. The performance is evaluated in terms of packet recovery rate for differentially encoded images.

In this paper, we propose a more general MAP framework that does not necessarily require soft decoding. The performance is evaluated in terms of bit error probability after arithmetic decoding, and is compared with a concatenated FEC scheme. Finally, the proposed system is tested in the case of SPIHT encoded [8] images transmitted across a BSC. The paper is organized as follows. In Section 2, we briefly review arithmetic coding with a forbidden symbol, and in Section 3 we discuss the MAP decoding. Two possible search techniques are covered in Section 4. Results and conclusions appear in Sections 5 and 6.

2. ARITHMETIC CODING

The objective of the arithmetic coder is to map a sequence of input symbols \mathbf{a} onto a binary string \mathbf{b} that represents

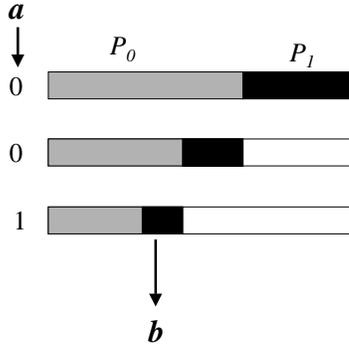


Figure 1: Binary arithmetic coder

the probability of the input sequence. The encoder performs this mapping based on the available source model; the compression performance mainly depends on the accuracy of that model.

For simplicity, we will restrict our attention to the case of a binary memoryless source with a fixed number of symbols L , i.e. $\mathbf{a} \in \{\mathbf{a}_i\}_{i=1}^{2^L}$. This simple source is fully described by the probability of the two symbols, P_0 and P_1 respectively. The concepts introduced in the following can be extended to more general cases.

Arithmetic encoding is performed by progressively isolating the probability interval corresponding to the input string. In Figure 1, an encoding example for the binary source $\mathbf{a} = \{001\}$ is shown; the output sequence \mathbf{b} corresponds to the shortest binary string contained in the isolated interval. Decoding follows the dual process. Both encoding and decoding can be accomplished sequentially: in the example of Figure 1, decisions on the output bits can be taken during the encoding process as soon as the right end of the selected interval becomes lower than 0.5 (see [3] for details).

Arithmetic decoding is extremely sensitive to errors; even a single flipped bit in the output string can cause irreversible desynchronization. Paradoxically, it is this poor resynchronization ability that allows powerful continuous error detection. In [4, 5] a dummy symbol μ with probability $P_\mu = \epsilon$ is introduced in the input alphabet, but it is never transmitted (see Fig. 2). This corresponds to perturbing the source model by a factor $1 - \epsilon$, reducing compression efficiency. The amount of added rate redundancy [5] is

$$R_x = -\log_2(1 - \epsilon) \text{ bits/symbol.} \quad (1)$$

The decoder is able to detect an error if the forbidden symbol μ is decoded. In the presence of a transmission error, due to the rareness of resynchronization, the decoder will reveal a forbidden symbol after a delay that is inversely proportional to ϵ . Details on error detection delay can be found in [5].

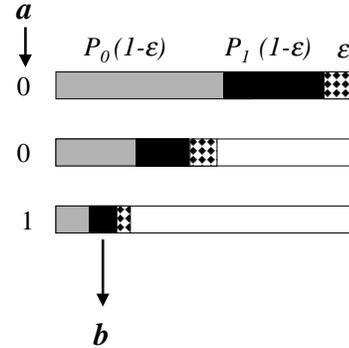


Figure 2: Binary arithmetic coder with forbidden symbol

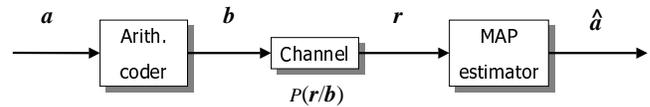


Figure 3: Transmission system block diagram

3. MAP DECODER

A forbidden symbol alters the source model and forces the encoder to keep an amount of redundancy that depends on ϵ . This known redundancy can be exploited by a MAP decoder in order to obtain the best estimate of the transmitted string \mathbf{a} in the presence of errors. In Fig. 3, the transmission system considered in the following is shown:

- the binary string $\mathbf{a} \in \{\mathbf{a}_i\}_{i=1}^{2^L}$ of length L is encoded by the arithmetic coder using a forbidden symbol with probability ϵ ;
- as a consequence the input sequence \mathbf{a} is mapped onto a variable length N codeword $\mathbf{b} \in \{\mathbf{b}_i\}_{i=1}^{2^L}$;
- the codeword is then transmitted across the channel with transition probability $P(\mathbf{r}/\mathbf{b})$;
- the received string \mathbf{r} is processed by a MAP estimator that will select the most probable sequence $\hat{\mathbf{a}}$:

$$P(\hat{\mathbf{a}} = \mathbf{a}_k/\mathbf{r}) \geq P(\mathbf{a}_i/\mathbf{r}) \quad \forall i \neq k \quad (2)$$

The term $P(\mathbf{a}_i/\mathbf{r})$ in Equation 2 represents the so called decoding metric, and can be expressed as

$$\begin{aligned} P(\mathbf{a}_i/\mathbf{r}) &= \frac{P(\mathbf{r}/\mathbf{a}_i)P(\mathbf{a}_i)}{P(\mathbf{r})} = \\ &= \frac{P(\mathbf{r}/\mathbf{b}_i)P(\mathbf{a}_i)}{P(\mathbf{r})} \end{aligned} \quad (3)$$

where $P(\mathbf{r}/\mathbf{b}_i)$ and $P(\mathbf{a}_i)$ are known terms and represent the channel transition probability and the probability of transmitting the string \mathbf{a}_i respectively. However, the term $P(\mathbf{r})$ cannot be easily evaluated and in the following will be approximated by 2^{-N} , where N is the length of the received codeword. This approximation assumes that all the received sequences of a certain length are equally likely; the assumption is certainly incorrect for variable length codes, but its exact evaluation would require as much effort as the MAP decoding itself. This abrupt simplification is also performed in [1, 2] for complexity reasons and it provided good results. The most direct approach to MAP decoding should be the evaluation of metric (3) for the subset \mathbf{B}_N , containing the codewords \mathbf{b}_i of length N ; it is evident that for reasonable input string length L , this exhaustive approach is infeasible due to the large search space. We must resort to a sub-optimal criterion to reduce the search space dimension. A number of search techniques were proposed in the past, the most popular one being the Viterbi algorithm for decoding convolutional codes; a complete survey can be found in [9]. These techniques usually require a trellis representation of the search space and an additive branch metric.

We can recast the search for the best \mathbf{b}_i as a search among all the possible binary strings of length N $\{\mathbf{x}_i\}_{i=1}^{2^N} \supseteq \mathbf{B}_N$. The $\{\mathbf{x}_i\}_{i=1}^{2^N}$ can be represented by a tree that grows exponentially with N . The metric (3), in its logarithmic form, can be easily decomposed into additive branch terms. The channel term is computed comparing the received \mathbf{r} sequence and the explored branch. The source term is obtained attempting partial arithmetic decoding of a given tree path; in the case $\mathbf{x}_i \notin \mathbf{B}_N$, the forbidden symbol will be revealed at some point, and the explored path will be pruned.

4. SEARCH TECHNIQUES

Due to the exponential dimension of the search tree, we resort to techniques able to reduce the exploration to the most probable paths. Moreover, the search strategy must implement sequential decoding in order to prevent long decoding delay. In this section, we quickly review two search algorithms we applied to explore the $\{\mathbf{x}_i\}_{i=1}^{2^N}$ tree, namely the *stack algorithm* and the *M-algorithm*. For a more systematic disquisition, see [9].

4.1. Stack algorithm

The stack algorithm (SA) is known as a *metric first* technique: the best path selection is performed in a greedy way, extending at each iteration the best stored path, i.e., the one with the best accumulated metric (3).

This is accomplished by storing all the visited paths in an ordered list, with maximum length M . Each element of the list contains the accumulated metric and the state information for

sequential arithmetic decoding. At each iteration, the best stored path is extended one branch forward. The extended path is dropped if the forbidden symbol is revealed or if the number of decoded symbols exceeds L . The branching goes on until the stopping criterion is fulfilled; in our implementation, the algorithm terminates when the best path in storage corresponds to a valid input sequence $\{\mathbf{a}_i\}_{i=1}^{2^L}$.

As with the Viterbi algorithm, decoding can be performed sequentially since there is a similar merging effect of all paths after a certain delay of D received bits.

4.2. M-algorithm

The M-algorithm (MA) limits the search space to the M best paths at each depth of the tree; for this reason MA belongs to the *breadth first* techniques, where the breadth is M .

At each iteration, all the stored paths, which are characterized by the same depth n in the tree, are extended one step forward; the same dropping rules of the SA are then applied and only the M best paths at depth $n + 1$ are stored. When the algorithm reaches the maximum depth $n = N$, the best stored path is considered the best estimate $\hat{\mathbf{a}}$. As in the previous case, sequential decoding can be easily obtained.

It is worth noticing that both of the presented algorithms can fail the decoding: in fact, because of the search space reduction, the correct path can be irreversibly dropped during the recursions.

5. EXPERIMENTAL RESULTS

The proposed algorithms have been tested in the case of transmission across the *Binary Symmetric Channel* (BSC) with bit error probability p_b ; it is worth pointing out that the proposed MAP approach can be applied to any kind of channel as long as the transition probability $P(\mathbf{r}/\mathbf{b})$ is known. The performance is compared with separated source and channel coding, employing *rate compatible punctured convolutional codes* (RCPC) [10].

We first introduce some notation. We can analytically evaluate the overall transmitted rate, expressed in bit per symbol (bps), as $R = H + R_x$, where $H = -\sum_{i=0}^1 P_i \log(P_i)$ is the source entropy and R_x is the rate redundancy introduced in (1). This corresponds to an equivalent coding rate $R_C = H/R$.

5.1. Memoryless source

In this section, we report the results obtained with a random memoryless source with symbol probabilities P_0 and P_1 . In Fig. 4, an example of sequential decoding, obtained by SA with decoding delay $D = 100$, is shown. The number of decoded symbols is reported as a function of the depth (number of received bits of the transmitted codeword \mathbf{b}_i) of the best path in storage. In the reported example, $L = 2000$,

$P_0 = 0.9$, $p_b = 10^{-3}$, $\epsilon = 0.08$ and $M = 2048$. It is worth noticing the initial decoding delay and the non uniform output rate of the decoder, typical of the variable length code. The effect of arithmetic coder termination is also noticeable.

Tables 1 and 2 show the performance obtained with arithmetic MAP decoding in terms of decoded bit error rate (BER) and algorithm complexity. Decoded BER is evaluated on source bits, without including in the computation those cases which correspond to a decoding failure; as previously mentioned, the suboptimal SA and MA techniques may fail the decoding, due to the reduced cardinality of the explored space. The algorithm complexity is evaluated in terms of the decoding time required by a Pentium Pro 200 MHz, and both the average and the peak decoding times are reported. The memory limits M for SA and MA have been chosen in order to obtain the same average decoding complexity in the worst case ($R_C = 8/9$); for this reason $M = 2048$ and $M = 256$ have been used for SA and MA respectively.

The performance of SA and MA is compared with the separated approach constituted by conventional arithmetic coding followed by RCPC (convolutional code with memory 6 and punctured rate 1/3). In Table 1 the number of failures N_f , the decoded BER, the average decoding time ΔT and the peak decoding time ΔT_M are reported, in the case $P_0 = 0.9$ and $p_b = 5 \cdot 10^{-3}$ for three values of the coding rate R_C ; all the results are averaged on 1000 iterations. It can be observed that, in terms of decoded BER, MAP decoding offers good results at all coding rates. The worse decoded BER performance of the RCPC is due to the fact that the Viterbi decoder always finds an estimate of the transmitted sequence, and the arithmetic decoder blindly finds the corresponding source string, heavily propagating the channel decoding errors to the source. On the contrary, the JSCC approach prevents this disruptive arithmetic decoding, thereby improving decoded BER. Nevertheless, when the added redundancy is small, a significant number of decoding failures appears. It is worth noticing that many applications (e.g., image or video transmission) can take advantage of the improved decoded BER, even in the presence of failures. In case of failure, the sequential MAP decoder can still provide to the application the symbols estimated before error detection. With a quality scalable encoder, we may be able to obtain a satisfactory approximation of the transmitted signal, preventing dangerous error propagation. The efficient retransmission of the lost part of the stream could be an alternative solution. The proposed results also underline that SA exhibits a higher number of failures but slightly better decoded BER.

The complexity of MAP decoding turns out to be generally quite prohibitive, if compared to conventional channel coding. It is important to notice that MA requires a long decoding time regardless of the coding rate; this behavior is due to the almost deterministic searching rule that always explores approximately NM nodes in the tree. On the other

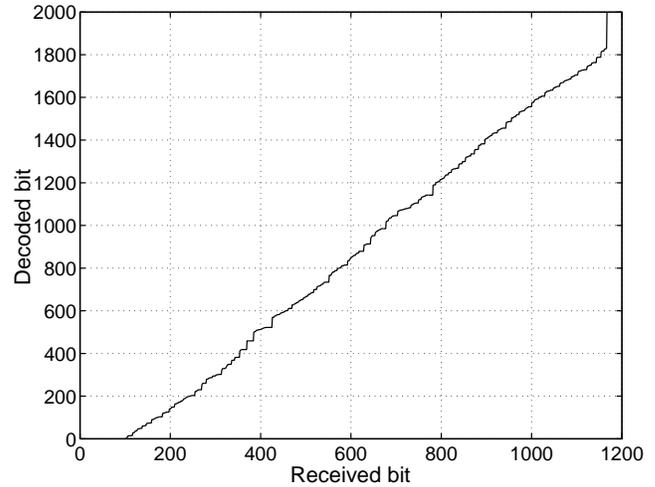


Figure 4: Stack algorithm: sequential decoding example in the case $D = 100$, $L = 2000$, $P_0 = 0.9$, $p_b = 10^{-3}$, $\epsilon = 0.08$ and $M = 2048$; the number of output symbols is reported as a function of the number of received bits being processed.

hand, SA decoding time depends on the coding rate and on the channel conditions; in particular, when the correct path is largely more probable than the incorrect ones, e.g., the channel bit error probability p_b is negligible or the redundancy ϵ is quite large, SA explores a reduced number of nodes, providing faster decoding. For this reason, SA performance is attractive in terms of both decoded BER and complexity for the largest value of ϵ in Table 1, where the decoding is even faster than traditional RCPC.

In Table 2, the results in the case $P_0 = 0.8$ are reported in order to notice that performance is affected by the source model as well. In the RCPC case, the decoded BER values are higher than in Table 1 because the input strings are less easy to compress, and more errors appear in the corresponding longer transmitted codeword. In the MAP case, the number of more likely input sequences increases (in the limit $P_0 = 0.5$ all the possible input strings are equally likely and the source term in the decoding metric becomes useless) impairing the performance in terms of both decoded BER and decoding time.

5.2. Image transmission

The proposed MAP techniques can be profitably applied to the transmission of scalably compressed sources. In this paper, we use MAP arithmetic decoding to obtain reliable transmission of SPIHT [8] compressed images. SPIHT is a popular progressive image codec in the wavelet domain; it uses a bit-plane approach and partitions wavelet coefficients into trees connecting different resolution levels in the same spatial

Table 1: RCPC (RC), SA with $M = 2048$, MA with $M = 256, P_0 = 0.9, p_b = 5 \cdot 10^{-3}, L = 2000$, 1000 iterations

R_C	ϵ	R (bps)		N_f	BER	ΔT	ΔT_M
8/9	0.04	0.53	RC	0	$3.7 \cdot 10^{-2}$	138 ms	170 ms
			SA	57	$9.0 \cdot 10^{-3}$	7.6 s	108 s
			MA	1	$1.4 \cdot 10^{-2}$	7.19 s	8.3 s
4/5	0.078	0.59	RC	0	$5.6 \cdot 10^{-3}$	137 ms	160 ms
			SA	7	$2.7 \cdot 10^{-4}$	805 ms	105 s
			MA	2	$1.4 \cdot 10^{-3}$	7.44 s	8.3 s
2/3	0.15	0.71	RC	0	$1.3 \cdot 10^{-4}$	141 ms	170 ms
			SA	0	$1.2 \cdot 10^{-5}$	32 ms	2 s
			MA	0	$2.2 \cdot 10^{-5}$	8.1 s	8.9 s

Table 2: RCPC (RC), SA with $M = 2048$, MA with $M = 256, P_0 = 0.8, p_b = 5 \cdot 10^{-3}, L = 2000$, 1000 iterations

R_C	ϵ	R (bps)		N_f	BER	ΔT	ΔT_M
8/9	0.06	0.81	RC	0	$9.5 \cdot 10^{-2}$	210 ms	230 ms
			SA	91	$2 \cdot 10^{-2}$	15 s	133 s
			MA	3	$3.8 \cdot 10^{-2}$	11 s	11.7 s
4/5	0.11	0.90	RC	0	$2 \cdot 10^{-2}$	211 ms	230 ms
			SA	10	$1.2 \cdot 10^{-3}$	1.7 s	150 s
			MA	1	$5.7 \cdot 10^{-3}$	11 s	12 s
2/3	0.22	1.08	RC	0	$5 \cdot 10^{-4}$	215 ms	240 ms
			SA	0	$6.3 \cdot 10^{-5}$	49 ms	2.1 s
			MA	0	$1.6 \cdot 10^{-5}$	12 s	13 s



Figure 5: GIRL 256x256 test image.

Table 3: SPIHT stream transmission across BSC with $p_b = 10^{-3}$.

Algorithm	ϵ	R_C	PSNR (dB)	
			ave.	max.
SA	0.08	0.91	31.97	32.56
MA	0.08	0.91	31.55	32.56
SA	0.15	0.82	32.09	32.09
MA	0.15	0.82	31.85	32.09

location. The tree encoding structure allows for a compact representation, but turns out to be very sensitive to transmission errors. The proposed sequential MAP decoding technique allows us to provide SPIHT with error resilience without affecting the progressive nature of the stream. The SPIHT encoder is simply cascaded with the arithmetic coder with forbidden symbol, without any packetization. The SPIHT bit-stream is modelled as a memoryless binary source; the probability P_0 is obtained by counting the zero occurrences during encoding. The P_0 and ϵ values employed by the encoder must be sent as side information to the decoder.

Table 3 shows the average and maximum Peak Signal to Noise Ratio (PSNR), obtained on 1000 transmissions of the 256x256 GIRL image (see Fig. 5) across the BSC with bit error probability $p_b = 10^{-3}$. Two different values of ϵ are used. In all cases, the overall transmitted rate is equal to 0.25 bpp. SA and MA with $M = 2048$ and $M = 256$ respectively are employed. In case of decoding failure, the available portion of output symbols is provided to SPIHT decoder. In Table 3, it is apparent that SA offers the best performance in term of decoded image quality, and therefore in the following we will limit our attention to this.

Finally, we compare the proposed image transmission system

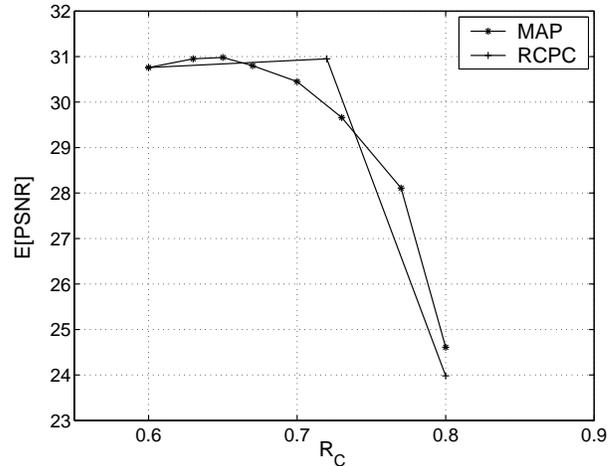


Figure 6: Average PSNR as a function of R_C , $p_b = 10^{-2}$.

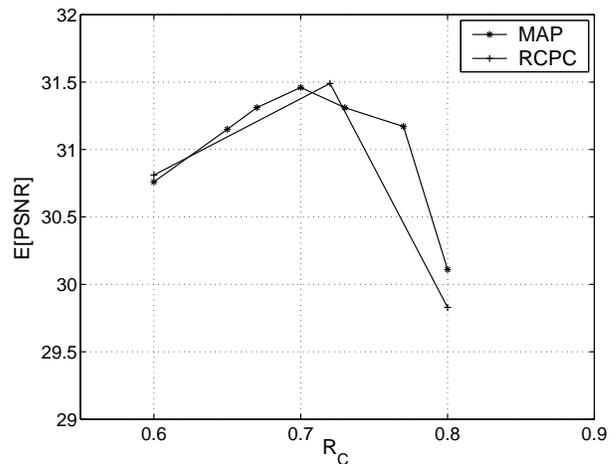


Figure 7: Average PSNR as a function of R_C , $p_b = 5 \cdot 10^{-3}$.

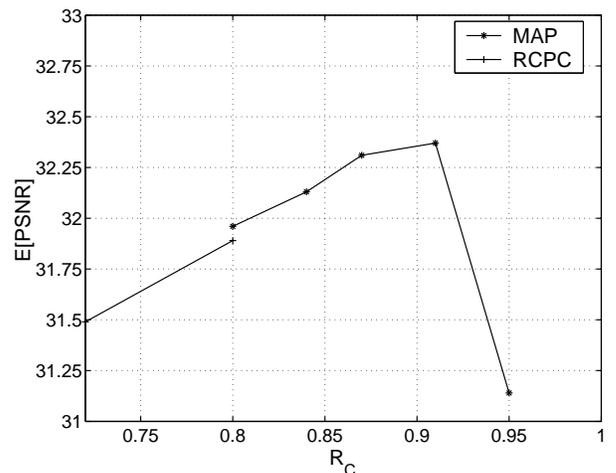


Figure 8: Average PSNR as a function of R_C , $p_b = 10^{-3}$.

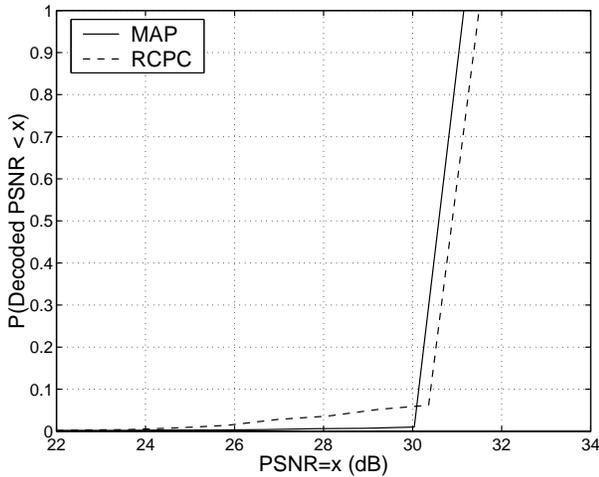


Figure 9: PSNR cumulative distribution, $p_b = 10^{-2}$.

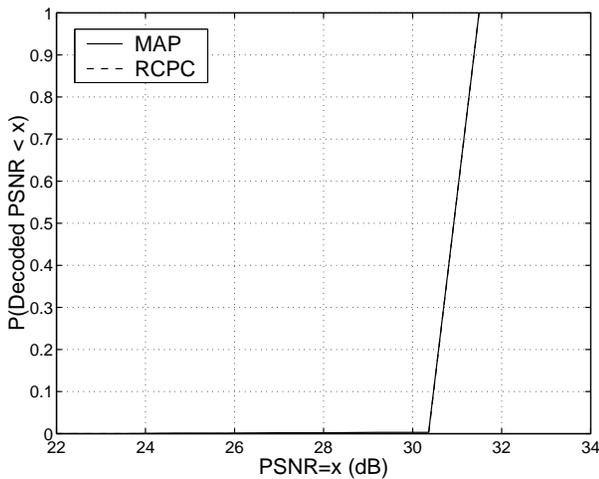


Figure 10: PSNR cumulative distribution, $p_b = 5 \cdot 10^{-3}$.

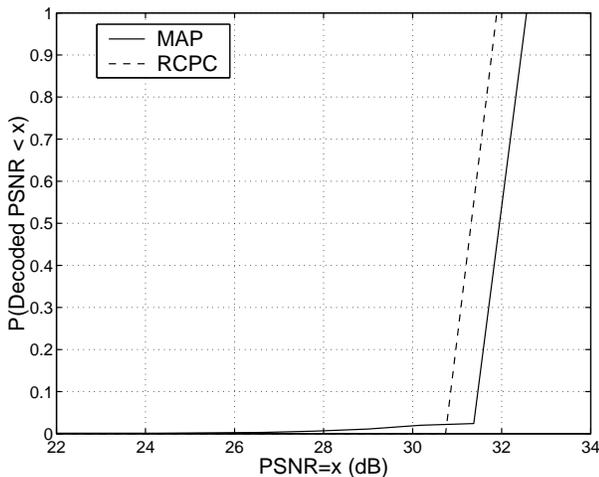


Figure 11: PSNR cumulative distribution, $p_b = 10^{-3}$.

with the technique introduced in [11], a reliable and efficient scheme based on standard error control coding. In [11] the SPIHT bit-stream is partitioned into packets in order to keep its scalability, and each packet is protected by a concatenated code consisting of an outer cyclic redundancy code (CRC) and an inner RCPC; the outer CRC is employed to provide error detection during RCPC Viterbi decoding. In the following the SA and RCPC/CRC schemes are compared for image transmission at 0.25 bpp across the BSC. An RCPC code with memory 6 and punctured rate 1/3 is used [10]. For MAP arithmetic decoding, SA with $M = 2048$ is employed. In Fig. 6-8, the average decoded PSNR is reported as a function of the coding rate R_C ; star and cross markers refer to the MAP and RCPC/CRC approaches respectively. In Fig. 6 and 7, the BSC bit error probabilities are $p_b = 10^{-2}$ and $p_b = 5 \cdot 10^{-3}$ respectively; in these two cases the two systems show similar performance in terms of best average PSNR. The proposed MAP decoding has the main advantage of offering a continuum of coding rates, depending on the ϵ value; on the other hand, the RCPC coding rate is limited to a finite set of puncturing choices. It is worth noticing that the proposed JSCC decoder obtains the same performance as the powerful concatenated channel schemes without the need for packetization. In Fig. 8, similar results are reported in the case $p_b = 10^{-3}$. The milder condition of the channel allows us to employ lower values of coding redundancy in the MAP decoder. These lower values are not available in the RCPC case; the last RCPC point is shown at $R_C = 0.8$. Therefore the JSCC system shows a gain of 0.5 dB in the best average PSNR. As already noticed, in mild channel conditions, SA is competitive in terms of decoding complexity as well.

The average decoded PSNR is generally used to evaluate the performance of image transmission systems, but it is evident that does not provide a complete view of the actual quality of service. For this reason in Fig. 9-11 we report the cumulative distribution of the decoded PSNR, i.e. the probability that the received image quality is below a given threshold, for all the simulated values of p_b . In each figure the cumulative distribution corresponding to the best MAP and RCPC/CRC case, in terms of average decode PSNR, is reported. It can be observed that both the techniques are characterized by low distribution tails; in other words the reception of poor quality images is unlikely and the quality of service is fairly high. In Fig. 9 ($p_b = 10^{-2}$) it is worth noticing that the proposed MAP decoder presents a slightly better distribution for low PSNR. In the case $p_b = 5 \cdot 10^{-3}$, reported in Fig. 10, the two distributions are completely overlapped. In the last case, corresponding to $p_b = 10^{-3}$, it is noticeable the advantage of the MAP decoder in terms of maximum decoded PSNR, while the discrepancies for low PSNR are negligible.

In conclusion, the proposed JSCC approach turns out to be well suited for scalable source coding; the main advantage is

represented by the possibility of choosing an arbitrary value of redundancy ϵ , without being limited to a finite number of coding rates as in the RCPC case. Furthermore, the redundancy can be adjusted to the source symbol priority, with the only constraint that the decoder must be able to apply the same updating rule on the received sequence.

6. CONCLUSIONS

Arithmetic coding with a forbidden symbol allowed us to design a powerful JSCC system for transmission of compressed data across noisy channels. The proposed approach exhibits the same performance as standard RCPC and has the advantage of being fully rate compatible. The coding rate can be chosen with absolute flexibility and can be adapted to the source and channel characteristics. Moreover, the MAP estimation allows sequential decoding, which can profitably be exploited in the case of SPIHT progressive image transmission.

7. REFERENCES

- [1] K. Sayood, H. Otu, and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 787–794, May 2000.
- [2] M. Park and D. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP source estimation," *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.
- [3] I. H. Witten, J. G. Cleary, and R. Neal, "Arithmetic coding for data compression," *Commun. ACM*, , no. 6, pp. 520–540, June 1987.
- [4] C. Boyd, J. Cleary, S. Irvine, I. Rinsma-Melchert, and I. Witten, "Integrating error detection into arithmetic coding," *IEEE Trans. Commun.*, vol. 45, no. 1, pp. 1–3, Jan. 1997.
- [5] J. Chou and K. Ramchandran, "Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 861–867, June 2000.
- [6] R. Anand, K. Ramchandran, and I. V. Kozintsev, "Continuous error detection (CED) for reliable communication," *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1540–1549, Sept. 2001.
- [7] B.D. Pettijohn, M.W. Hoffman, and K. Sayood, "Joint source/channel coding using arithmetic codes," *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1540–1548, Sept. 2001.
- [8] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, June 1996.
- [9] J. B. Anderson and S. Mohan, *Source and channel coding*, KLUWER, 1991.
- [10] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCPC codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1998.
- [11] O. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Processing Lett.*, vol. 4, pp. 189–191, July 1997.