

Joint Source/Channel Coding and MAP Decoding of Arithmetic Codes

Marco Grangetto, *Member, IEEE*, Pamela Cosman, *Senior Member, IEEE*, and Gabriella Olmo, *Member, IEEE*

Abstract—In this paper, a novel maximum *a posteriori* (MAP) estimation approach is employed for error correction of arithmetic codes with a forbidden symbol. The system is founded on the principle of joint source channel coding, which allows one to unify the arithmetic decoding and error correction tasks into a single process, with superior performance compared to traditional separated techniques. The proposed system improves the performance in terms of error correction with respect to a separated source and channel coding approach based on convolutional codes, with the additional great advantage of allowing complete flexibility in adjusting the coding rate. The proposed MAP decoder is tested in the case of image transmission across the additive white Gaussian noise channel and compared against standard forward error correction techniques in terms of performance and complexity. Both hard and soft decoding are taken into account, and excellent results in terms of packet error rate and decoded image quality are obtained.

Index Terms—Arithmetic coding, image transmission, joint source/channel coding (JSCC), maximum *a posteriori* (MAP) estimation.

I. INTRODUCTION

THE FUTURE of telecommunications is being driven by two impressive recent phenomena, namely the widespread diffusion of the Internet and the development of personal mobile communications. The migration to the wireless channel of Internet-based services such as multimedia communications, video conferencing, and digital image and music sharing is colliding with the bandwidth and power limitations imposed by the mobile environment. As a consequence, the research community is moving toward more interdisciplinary approaches involving networking, digital communications and multimedia expertise in order to improve the overall quality of service yielded by the system.

In this light, *joint source/channel coding* (JSCC) techniques are emerging as a natural integration of the multimedia and digital communication worlds. In fact, the wireless bandwidth limitation and the high data rates and latency constraints imposed by multimedia services are emphasizing the practical shortcomings of Shannon's source-channel separation theorem [1]. JSCC techniques are based on the fact that in practical

cases the source encoder is not able to ideally decorrelate the input sequence; some implicit redundancy is still present in the compressed stream and can be properly exploited by the decoder for error control. As a consequence, it is possible to improve the decoder performance considering source and channel coding jointly.

In the JSCC field, considerable attention has been devoted in the past to the error resilience of *variable length codes* (VLCs) [2]–[7]. VLCs represent the final entropy coding stage for many coding standards such as JPEG, MPEG-4 and H.263, and their robustness to transmission errors is clearly a major issue. In [2], [3] the residual redundancy in the source encoder output is represented by 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 implements Huffman coding of neighboring pixel differences. Another MAP decoding technique for VLCs is proposed in [4], and tested in the case of transmission of a first order Markov source. In [5] soft decoding is used, and results for MPEG-4 reversible VLCs are reported. In [6], a low complexity soft decoding of VLCs is proposed; the results also include joint source channel decoding when turbo codes are used for error correction. In [7] the turbo principle is applied to joint source-channel decoding of VLCs when cascaded with a channel code.

On the other hand, emerging coding standards such as JPEG 2000 [8] and JBIG2 [9] for still pictures and H.264 [10] for video sequences use *arithmetic coding* (AC) as the final entropy coding stage. AC can allocate fractional numbers of bits to input symbols, thus improving the compression efficiency [11]. Moreover, it easily encompasses efficient adaptive coding solutions. However, AC is very sensitive to transmission error; the arithmetic decoder has poor resynchronization properties and the high compression efficiency prevents MAP decoding based on residual redundancy. Hence there is a great interest in resilient AC, and in JSCC techniques based on AC [12]–[23]. In [12] an error recovery technique based on automatic repeat request (ARQ) is implemented by means of AC with proper insertion of markers. In [13], [14] an AC that embeds channel coding is presented; the idea is to enforce a minimum Hamming distance constraint among encoded sequences, allowing the implementation of a MAP estimator at the decoder side. AC with error detection capability is proposed in [15], where variable to fixed length AC is designed. Another way to obtain error detection, based on the insertion of a *forbidden symbol* in the input alphabet, was initially proposed in [16], and further studied in [17] and [18]. The forbidden symbol allows one to adjust the amount of coding redundancy to be embedded in the coded stream. At

Paper approved by A. K. Khandani, the Editor for Coding and Signal of the IEEE Communications Society. Manuscript received May 1, 2003; revised September 9, 2004 and December 1, 2004.

M. Grangetto and G. Olmo are with the Center for Multimedia Radio Communications (CERCOM), Politecnico di Torino, 10129 Torino, Italy (e-mail: marco.grangetto@polito.it; gabriella.olmo@polito.it).

P. Cosman is with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA 92093-0407 USA (e-mail: pcosman@code.ucsd.edu).

Digital Object Identifier 10.1109/TCOMM.2005.849690

the expense of compression efficiency, it permits error detection at the decoder side. In [17], an ARQ strategy is validated in the case of lossless image transmission across the BSC. The coding redundancy conveyed by the forbidden symbol can be recognized by the decoder in order to attempt not only error detection but also error correction. In [19], error correction is performed in the case of transmission across the additive white Gaussian noise (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 [20], sequential decoding of arithmetic coded data is discussed and a very simple arithmetic encoder is shown to achieve excellent performance in terms of error correction. In [21], a JSCC concatenated scheme based on AC and trellis coded modulation is applied to image transmission. In [22] and [23], some preliminary work on MAP decoding of AC with a forbidden symbol is presented in the case of image transmission across the BSC.

In this paper, the MAP decoding approach is employed for error correction of binary AC with a forbidden symbol; the MAP approach is employed for both hard and soft decoding in the case of transmission across the AWGN channel. Two different sequential search techniques are applied to the estimation problem and compared in terms of complexity and performance. AC with a forbidden symbol and MAP estimation allow us to design a novel joint source/channel coding and decoding scheme with attractive features in terms of error correction, adaptivity and rate flexibility. The original contributions of our work mainly rely on the use of a MAP decoding metric, where the *a priori* knowledge of the source is taken into account in the correction algorithm, along with the forbidden symbol error detection and the channel transition probability. Some of this work has appeared in [22], [23], where transmission across the binary symmetric channel was considered. In [19] another JSCC approach based on AC with a forbidden symbol is proposed. In [19] a 256-symbol AC is employed to encode pixel prediction errors and the error correction task is based on a maximum-likelihood (ML) criterion, i.e., minimum distance. Results for AWGN channel transmission with soft decoding are given. In the present paper, the viability of the proposed approach is demonstrated in the case of image transmission in two typical cases: lossless predictive coding and lossy progressive coding. The first application allows us to validate the proposed techniques and to compare the MAP estimation algorithms with the results in [19], based on ML decoding. The second set of experiments is particularly innovative, since it couples the proposed error resilient sequential entropy coding and the popular SPIHT image coder. The ability to sequentially decode the SPIHT embedded bitstream in the presence of transmission errors is particularly attractive and it exhibits a better performance than powerful error protection techniques based on FEC [24]. During the review of the present paper, an independent work on sequential arithmetic decoding for reliable image transmission has been published [25]. In this case synchronization markers, instead of the forbidden symbol, are used for error detection purpose. A preliminary comparison with this alternative technique is reported in Section VI.

The paper is organized as follows. In Section II, AC with a forbidden symbol is briefly reviewed. In Section III, the error correction task is formulated in terms of a MAP estimation problem, and the employed sequential decoding algorithms are

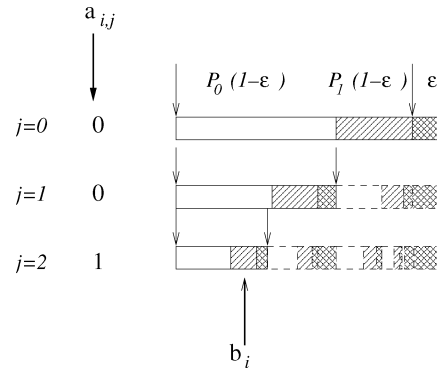


Fig. 1. Binary arithmetic encoder with forbidden symbol μ .

described in Section IV. In Sections V and VI, we evaluate the performance of the proposed technique. Finally, Section VII shows our conclusions and offers directions for future work.

II. ARITHMETIC CODING (AC)

In this paper we select a simple, yet significant, case study where the input sequence $\mathbf{a}_i = \{a_{i,0}, a_{i,1}, \dots, a_{i,L-1}\}$, $0 \leq i < 2^L$, is constituted by L binary outcomes of a binary memoryless source with symbol probabilities $P\{a_{i,j} = 0\} = P_0$ and $P\{a_{i,j} = 1\} = P_1 = 1 - P_0$, $0 \leq j < L$, $0 \leq i < 2^L$. It is worth noticing that the concepts introduced here can be generalized to more sophisticated source models as well as to adaptive AC.

Binary arithmetic encoding is an iterative task, whose objective is to map the input sequence \mathbf{a}_i , $0 \leq i < 2^L$ onto a variable length binary string \mathbf{c}_i , $0 \leq i < 2^L$, representing its probability. This task is performed by progressively refining the probability interval corresponding to \mathbf{a}_i . The probability interval is initialized to $(0,1)$ and then the interval portion corresponding to the encoded symbol is iteratively selected. After L iterations, an interval I , whose size corresponds to the input sequence probability, is obtained and encoded by means of the shortest binary sequence \mathbf{c}_i belonging to it; the expected number of bits required by this operation is $E[N_i] = -\log_2(I) = LH$, where $H = -\sum_{i=0}^1 P_i \log_2 P_i$ is the memoryless source entropy rate. The decoding task, in the error free case, simply follows the dual process and consists of iteratively finding the interval to which the encoded codeword belongs. It is worth pointing out that both encoding and decoding can be accomplished sequentially, avoiding excessive delay [11].

The decoding process is very fragile with respect to transmission errors; in fact, a single flipped bit can shift the correct codeword to the adjacent interval, causing irreversible desynchronization. This behavior has motivated the design of a number of error detection tools for AC, briefly described in Section I. In this paper, we use an error detecting AC based on the introduction of a forbidden symbol into the coding alphabet [16]. A forbidden symbol μ , which is never encoded and whose probability is fixed to an arbitrary value $P_\mu = \epsilon$, provides a form of coding redundancy. The introduction of μ clearly implies a modification of the symbol probabilities, that become less accurate with respect to the source model as ϵ increases. This corresponds to an amount of coding redundancy per encoded bit $R_\mu = -\log_2(1 - \epsilon)$ [17], that is forced in the encoded binary sequence at the expense of compression efficiency. In Fig. 1, we

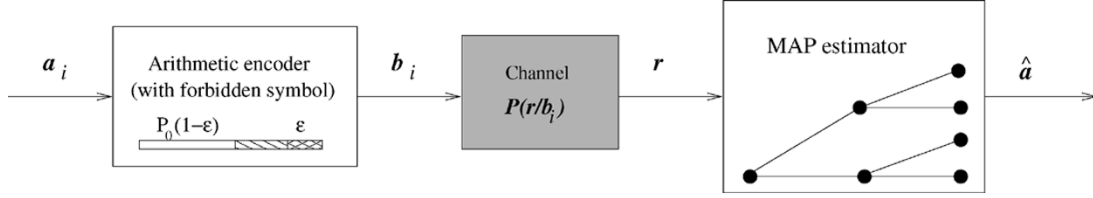


Fig. 2. Block diagram of the proposed transmission system.

show three iterations of the modified encoder with a forbidden symbol, which is based on the ternary alphabet “0”, “1” and “ μ ” with probabilities $P_0(1 - \epsilon)$, $P_1(1 - \epsilon)$ and ϵ , respectively. The presence of μ causes an interval shrinkage by a factor ϵ at each iteration, which corresponds to a codeword \mathbf{b}_i , whose expected length $E[N_i] = L(H + R_\mu)$ is increased with respect to the case without the forbidden symbol. At the decoder side, the presence of μ can be used for error detection; if the decoder detects the forbidden symbol, it means that transmission errors have occurred. Countermeasures can then be taken, such as retransmission. In [17], the reliability of this error detection mechanism is evaluated. In particular, if an error occurs in a certain position, the probability that the error detection delay is greater than n symbols is $(1 - \epsilon)^n$. Therefore, a large value of ϵ assures fast error detection, but it greatly reduces the compression efficiency. The performance of an error correcting decoder based on the forbidden symbol detection is described in the following sections. Moreover, the use of the forbidden symbol exhibits additional advantages if compared with other error detection techniques, such as the use of periodic cyclic redundancy check (CRC) or synchronization markers. First of all, the coding redundancy can be flexibly controlled by means of a single parameter, i.e., the value of the probability ϵ , which takes on continuous values and allows one to achieve any coding rate. Finally, the forbidden symbol guarantees the so called continuous error detection [18], since errors can be revealed sequentially during AC decoding; on the contrary, the use of periodic CRC or synchronization markers permits error detection only after a whole block of data has been completely decoded.

III. MAP DECODING METRIC

The coding redundancy associated with the forbidden symbol can be used by the decoder to select the best estimate of the encoded sequence, received through an error prone channel. In the following, we consider the transmission scheme whose block diagram is represented in Fig. 2. The variable length code word $\mathbf{b}_i = \{b_{i,0}, b_{i,1}, \dots, b_{i,N_i-1}\}$, corresponding to the input sequence $\mathbf{a}_i = \{a_{i,0}, a_{i,1}, \dots, a_{i,L-1}\}$, is transmitted across the channel with transition probability $P(\mathbf{r}|\mathbf{b}_i)$. Note that in the channel block we include modulation, channel transmission and demodulation. The receiver observes the demodulated sequence $\mathbf{r} = \{r_0, r_1, \dots, r_{N_i-1}\}$. It is worth noticing that, if one models sequences \mathbf{a} , \mathbf{b} and \mathbf{r} as random variables, they constitute a Markov chain, where AC introduces a variable amount of memory between source and coded symbols.

The objective of the MAP decoder is to find the most probable input sequence $\hat{\mathbf{a}}$

$$\hat{\mathbf{a}} : P(\hat{\mathbf{a}} = \mathbf{a}_i|\mathbf{r}) \geq P(\mathbf{a}_k|\mathbf{r}) \quad \forall k \neq i. \quad (1)$$

Therefore, the estimation is based on the following decoding metric:

$$P(\mathbf{a}_k|\mathbf{r}) = \frac{P(\mathbf{r}|\mathbf{a}_k)P(\mathbf{a}_k)}{P(\mathbf{r})} = \frac{P(\mathbf{r}|\mathbf{b}_k)P(\mathbf{a}_k)}{P(\mathbf{r})}. \quad (2)$$

In principle the decoding task consists of evaluating the metric (2) for all possible pairs $(\mathbf{a}_k, \mathbf{b}_k)$, such that the length of the encoded sequence \mathbf{b}_k is equal to N_i ; in the following we will denote the subset of codewords of length N_i as \mathbf{B}_{N_i} . The metric (2) includes the channel transition probability, the *a priori* source probability $P(\mathbf{a}_k)$ and the term $P(\mathbf{r})$. The first two terms are known and can be evaluated on the basis of the channel model and the *a priori* source model, respectively. The last term requires more attention. It can be written as

$$P(\mathbf{r}) = \sum_{k \in \mathbf{B}_{N_i}} P(\mathbf{r}|\mathbf{b}_k)P(\mathbf{a}_k). \quad (3)$$

It can be easily understood that the evaluation of this term is as complex as the whole decoding metric and requires knowledge of the subset \mathbf{B}_{N_i} , which is infeasible for practical values of N_i . Therefore, proper approximations will be adopted, as detailed in Sections III-A and III-B.

Finally, in the case of memoryless channels, it is useful to express the decoding metric in the additive form

$$\begin{aligned} m_k &= \log [P(\mathbf{a}_k|\mathbf{r})] \\ &= \sum_{j=0}^{N_i-1} \left\{ \log [P(r_j|b_{k,j})] + \log [P(\mathbf{a}_{k,j})] - \log [P(r_j)] \right\} \\ &= \sum_{j=0}^{N_i-1} m_{k,j} \end{aligned} \quad (4)$$

where

$$m_{k,j} = \log [P(r_j|b_{k,j})] + \log [P(\mathbf{a}_{k,j})] - \log [P(r_j)]. \quad (5)$$

The term $P(\mathbf{a}_{k,j})$ represents the *a priori* probability of the source symbols output by the arithmetic decoder associated with the j th bit of codeword \mathbf{b}_k ; it is worth noticing that, due to the variable length nature of AC, the number of decoded source symbols is variable and depends on both the codeword \mathbf{b}_k , and the bit position j . The additive metric $m_{k,j}$ can be employed as the branch metric by the sequential techniques used to find the best estimate. In the following sections we will analyze the branch metric for transmission across the AWGN channel with hard and soft decoding.

A. BPSK With Hard Decoding

For transmission across an AWGN channel using binary phase-shift keying (BPSK) modulation with signal to noise ratio E_b/N_0 and hard decoding, the channel transition probability is

$$P(r_j|b_{k,j}) = \begin{cases} 1 - p, & \text{if } b_{k,j} = r_j \\ p, & \text{if } b_{k,j} \neq r_j \end{cases} \quad (6)$$

with $p = (1/2)\text{erfc}(\sqrt{E_b/N_0})$.

As already discussed, the evaluation of $P(\mathbf{r})$ is not straightforward. Hence we adopt the approximation $P(\mathbf{r}) = 2^{-N_i}$, which amounts to assuming that there are 2^{N_i} equally likely possible codewords \mathbf{r} of length N_i . Because of the variable length of AC, this assumption does not generally hold true; in other words, not all the possible sequences of length N_i are valid AC codewords. Nevertheless, the adopted assumption has provided satisfactory results for MAP decoding of VLC in [2], [3] and can be justified by the concept of typical sequence [26]. In fact, for large values of L , the typical sequences \mathbf{a} , defined as those ones with LP_0 "0"s and LP_1 "1"s, are equally likely and are mapped onto AC codewords of $LH \approx N_i$ bits. The number of such sequences is $\binom{L}{LP_0}$, and can be approximated to 2^{N_i} by means of Stirling's formula. In conclusion, by means of this latter approximation, the branch decoding metric (5) turns out to be

$$m_{k,j} = \log [P(r_j|b_{k,j})] + \log [P(\mathbf{a}_{k,j})] + \log [2]. \quad (7)$$

B. BPSK With Soft Decoding

For BPSK modulation with soft decoding, the MAP estimator observes the values $r_j = h_{k,j} + n_j$, $j = 0, \dots, N_i - 1$, where $h_{k,j} = \sqrt{E_b}(2b_{k,j} - 1)$ is the modulated level and n_j is a Gaussian noise sample with zero mean and variance σ^2 . This channel model yields

$$\log [P(r_j|b_{k,j})] = -\frac{1}{2} \log(2\pi\sigma^2) - \frac{(r_j - h_{k,j})^2}{2\sigma^2}. \quad (8)$$

Using conditional probabilities, we can write $P(r_j) = \sum_{k=0}^1 P(r_j|b_j = k)P(b_j = k)$, where b_j is the j th bit of the transmitted codeword, and is assumed to take values "0" and "1" with equal probability. As in the hard decoding case, the equiprobability hypothesis greatly simplifies the evaluation. Thus it is straightforward to obtain

$$\log [P(r_j)] = -\log 2 - \frac{1}{2} \log(2\pi\sigma^2) - \frac{(r_j + \sqrt{E_b})^2}{2\sigma^2} + \log \left[1 + e^{-2r_j\sqrt{E_b}/\sigma^2} \right]. \quad (9)$$

Finally, in the soft decoding case, the branch decoding metric becomes

$$m_{k,j} = -\frac{(r_j - h_{k,j})^2}{2\sigma^2} + \log [P(\mathbf{a}_{k,j})] + \log 2 + \frac{(r_j + \sqrt{E_b})^2}{2\sigma^2} - \log \left[1 + e^{-2r_j\sqrt{E_b}/\sigma^2} \right]. \quad (10)$$

IV. SEQUENTIAL SEARCH TECHNIQUES

This section is devoted to the description of the algorithms used to evaluate the MAP decoding metrics (7) and (10). Direct evaluation of the MAP metric over the subset \mathbf{B}_{N_i} is infeasible for two reasons. The first is that, for typical values of L , it is impractical to store all 2^L codewords \mathbf{b}_k , so as to isolate those belonging to \mathbf{B}_{N_i} . The second is that the large cardinality of \mathbf{B}_{N_i} would anyhow prevent direct evaluation of the MAP metric. Therefore it is essential to resort to suboptimal search techniques, able to sequentially travel along the subset \mathbf{B}_{N_i} in order to pick up the most likely sequence.

The first obstacle can be solved by exploiting the forbidden symbol μ : the search for the best estimate is enlarged to all binary sequences of length N_i , $\mathbf{x}_k \in \mathbf{X}_{N_i} \supseteq \mathbf{B}_{N_i}$; those sequences that are not admissible codewords \mathbf{b}_k can be discarded, upon μ detection. The proposed decoder is implemented by means of a search algorithm along the branches of the binary tree representing \mathbf{X}_{N_i} . During tree exploration, the branch metric (7) or (10) can be accumulated, and non admissible codewords can be pruned upon error detection. It is worth recalling that error detection occurs with a delay whose probability depends on the value of ϵ . The problem of the large cardinality of the search space can be tackled by means of the sequential search strategies described in the following sections.

A. Stack Algorithm (SA)

The *stack algorithm* (SA) [27] is known as a *metric first* technique. The best path selection is based on a greedy approach, extending at each iteration the best stored path, i.e., the one with the best accumulated metric (4). This is accomplished by storing all the visited paths in an ordered list, with predefined 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 . Moreover, a branch is pruned if its metric falls below a threshold, so as to avoid the extension of extremely unlikely paths. The branching goes on until the stopping criterion is fulfilled; in our implementation, the algorithm terminates as soon as the best path in storage corresponds to a valid input sequence \mathbf{a}_i , i.e., a path in the tree that corresponds to the decoding of L source bits, when all the N_i codeword bits have been consumed.

Similarly to the Viterbi algorithm, decoding can be performed sequentially, due to the merging of all paths after a certain delay of D processed bits [27], [26]. In fact, as the least likely branches in the tree are progressively pruned, the surviving paths tend to merge into a single one, while moving back toward the tree root. This property can be exploited to progressively output decoded bits as soon as the decoding tree has collapsed into one single path.

B. M-Algorithm (MA)

The *M-algorithm* (MA) limits the search space to a number M of paths at each depth j in the tree; for this reason, it can

be classified as a *breadth first* technique [27], the breadth being represented by the M parameter.

At each iteration, all the stored paths, which are characterized by the same depth j in the tree, are extended one step forward and their metrics $m_{k,j+1}$ are evaluated and accumulated; since all the partially explored paths $x_{k,j+1}$ have the same length, the term $\log [P(r_{j+1})]$ in (7) and (10) is the same for all sequences and can be skipped. The same dropping rules of the SA are then applied and only the M best paths at depth $j+1$ are stored for the next iteration. When the algorithm reaches the maximum depth $j = N_i - 1$, the best stored path is taken as the best estimate $\hat{\mathbf{a}}$. As in the previous case, sequential decoding can be easily obtained.

It is worth noticing that both SA and MA can fail the decoding. In fact, because of the search space limitation, the correct path can be irreversibly dropped during the recursions. In such a case, the decoder can either declare a failure, or forward to the subsequent system layer the best partially decoded path; the latter solution is particularly effective if coupled with selective ARQ or when a progressive source decoder is used as in Section VI.

C. Codeword Termination Strategy

The forbidden symbol guarantees error detection only after a certain delay [18]; hence it allows continuous error detection during the sequential search. On the other hand, pruning of codewords $\mathbf{x}_k \notin \mathbf{B}_{N_i}$ is not assured when the decoder approaches the last bit position $j = N_i - 1$. To fix this shortcoming, a proper AC termination strategy is implemented. The encoder terminates each input sequence with an *end of block* (EOB) symbol, with probability ω ; the EOB is always encoded as the $(L + 1)$ th input symbol, using a modified source model which includes the two binary symbols with probability $P_0(1 - \omega)$ and $P_1(1 - \omega)$, and the EOB symbol with probability ω . The same termination rule is enforced at the decoder side and the estimates that do not respect the EOB constraint are discarded; this supplementary error detection tool is activated when the search algorithm reaches a leaf of the binary tree at depth $j = N_i - 1$.

V. LOSSLESS IMAGE COMPRESSION AND TRANSMISSION

The JSCC system described in Sections III and IV can be profitably exploited for joint data compression and error protection. In the following, we present the experimental results for lossless compression and transmission of still images. All the reported results are worked out on the *GIRL 256 × 256* test image.

The experiments use a simple lossless image compression scheme based on the predictor proposed in the JPEG lossless coding system [28]. The predicted pixel at row i and column j is $\hat{x}_{i,j} = x_{i,j-1} + x_{i-1,j} - x_{i-1,j-1}$; in the case of 8-bit grayscale images, the prediction error $e_{i,j} = x_{i,j} - \hat{x}_{i,j}$ is represented by a 9-bit symbol. In our implementation, the most probable prediction error values, i.e., those that are small in magnitude, are mapped on 9-bit codewords with the maximum possible number of "0"s; this way, an unbalanced binary input for the subsequent entropy coder is obtained. It is worth noticing that this mapping

rule has not been optimized from the point of view of compression efficiency, which is not the main target of the present paper.

Packets of 256 symbols ($L = 2304$ bits) are formed and encoded by means of binary nonadaptive AC; each packet is terminated with the EOB symbol with probability $\omega = 10^{-5}$, which has been selected as a tradeoff between rate overhead and error detection capability. The resulting variable length packets are then transmitted across the AWGN channel. The packet length, along with the *a priori* bit probability P_0 , is sent as side information to the decoder. To this end, a header is added to each packet and is protected using a (3,1,4) convolutional code. The proposed scheme achieves a coding rate of 5.1 bits per pixel (bpp) on the *GIRL 256 × 256* test image when no forbidden symbol is used; note that, as already mentioned, the optimization of the compression performance is beyond the scope of the paper.

In the following, the JSCC AC scheme will be compared with a traditional separated approach, where 256 symbol packets are first compressed with binary AC ($\epsilon = 0$) and then protected against transmission errors by means of rate compatible punctured convolutional (RCPC) codes; we employ the code family with memory $\nu = 6$ and non punctured rate $1/3$, proposed in [29]. In the following, the proposed JSCC and the separated coding systems will be compared for the same value of the overall coding rate transmitted over the channel. The separated approach yields an overall coding rate equal to H/R_C bits per each binary input, where H is the first order entropy of the binary input sequence and R_C is the employed RCPC coding rate. In the JSCC case, the overall coding rate is $H + R_\mu$ bits per input symbol. As a consequence, the use of the forbidden symbol μ , with probability ϵ , corresponds to an equivalent channel coding rate $R_C = H/(H + R_\mu)$.

First of all, the MAP decoders based on SA and MA are validated in terms of the tradeoff between performance and complexity, the critical parameter being represented by the choice of the maximum memory M . In Table I, both SA and MA are tested with $\epsilon = 0.05$, corresponding to a coding rate $R_C = 8/9$, for $E_b/N_0 = 4.323, 5.208, \text{ and } 6.789$ dB, yielding channel transition probabilities $p = 10^{-2}, 5 \times 10^{-3}, \text{ and } 10^{-3}$, respectively. The hard decoding metric (7) is used. The results obtained with the RCPC code of equivalent rate are shown for comparison. The performance in terms of the packet error rate (PER) is reported. Two complexity measures, $E[V]$ and T , are included as well. $E[V]$ is the average number of visited nodes normalized by the codeword length; thus $E[V] = 1$ means that a single path in the tree is explored from depth $j = 0$ to $j = N_i - 1$ without any branching. For comparison, the Viterbi decoder evaluates $2^{\nu+1}$ metrics in the convolutional trellis at each iteration, and therefore the selected RCPC code has $E[V] = 128$. T is the average packet decoding time, obtained by a Pentium IV at 1.8 GHz. In Table I, both SA and MA outperform the separated RCPC codes in terms of PER for the selected values of $256 \leq M \leq 4096$. When $p = 5 \times 10^{-3}$ and $p = 10^{-2}$, all the decoders exhibit a poor correction performance in terms of PER; this is clearly due to the use of a large channel coding rate $R_C = 8/9$ which is inadequate to counteract a very noisy channel. Nevertheless, the JSCC system always yields better results than separated RCPC coding. It can be noticed that the two algorithms converge to the same performance in the limit

TABLE I
PER, AVERAGE VISIT $E[V]$, AND AVERAGE FRAME DECODING TIME T FOR RCPC, SA, AND MA AS A FUNCTION OF M . CODING RATE $R_C = 8/9$,
CORRESPONDING TO $\epsilon = 0.05$, HARD DECODING

	M	$p = 10^{-3}$			$p = 5 \cdot 10^{-3}$			$p = 10^{-2}$		
		PER	$E[V]$	T	PER	$E[V]$	T	PER	$E[V]$	T
SA	512	10^{-2}	3.7	13 ms	$3.2 \cdot 10^{-1}$	50	161 ms	$8.7 \cdot 10^{-1}$	119	382 ms
	1024	$3.7 \cdot 10^{-3}$	3.4	16 ms	$2.7 \cdot 10^{-1}$	80	361 ms	$7.8 \cdot 10^{-1}$	205	919 ms
	2048	$1.9 \cdot 10^{-3}$	3.5	22 ms	$2 \cdot 10^{-1}$	123	850 ms	$7.2 \cdot 10^{-1}$	411	2.7 s
	4096	$7.5 \cdot 10^{-4}$	3.4	25 ms	$1.1 \cdot 10^{-1}$	128	1.7 s	$5.8 \cdot 10^{-1}$	629	7.8 s
MA	256	$1.6 \cdot 10^{-3}$	510	858 ms	$1.8 \cdot 10^{-1}$	507	864 ms	$6.2 \cdot 10^{-1}$	505	865 ms
	512	$8.4 \cdot 10^{-4}$	1019	1.8 s	10^{-1}	1016	1.8 s	$5.9 \cdot 10^{-1}$	1016	1.9 s
	1024	$6.5 \cdot 10^{-4}$	2036	3.9 s	$7.6 \cdot 10^{-2}$	2033	4.4 s	$5.5 \cdot 10^{-1}$	2034	4.6 s
RCPC	-	$2.7 \cdot 10^{-2}$	128	24 ms	$4.7 \cdot 10^{-1}$	128	24 ms	$9.2 \cdot 10^{-1}$	128	24 ms

of large M , but MA saturates faster than SA. Nevertheless, the SA greedy search is quicker than MA when channel conditions are not harsh [27]. For $p = 10^{-3}$, SA exhibits an average visit as low as 3.4, and the decoding time is as fast as the Viterbi decoder for RCPC codes. Conversely, when $p = 10^{-2}$, the complexity of SA may become prohibitive. On the other hand, MA exhibits a deterministic behavior and $E[V] \simeq 2M$, i.e., the number of visited nodes at each depth. In fact, the MA computational burden is mainly due to the cost of the sorting operation of the $2M$ metrics at each iteration, amounting to $2M \log_2 2M$; the same sorting task affects SA complexity, but in this case the list is not always full, and the computational effort is more concentrated on erroneous sections of the received codeword.

In conclusion, the results reported in Table I show that the proposed decoders are able to outperform standard convolutional coding and exhibit a scalable complexity, which depends on the choice of M . On the other hand, the optimal choice of the system parameters remains an open issue; in fact, given a channel state, i.e., the channel SNR, and a source model, i.e., the value of P_0 in the case of the memoryless source, the error correction performance depends on the joint selection of the parameters M and ϵ . A large value of ϵ means more coding redundancy and assures a more efficient pruning of the decoding tree. On the other hand, increasing the sequential search memory M guarantees that a larger number of candidate paths are stored in the tree. The optimal performance in terms of both error correction and computational complexity corresponds to the optimal tradeoff between an efficient tree pruning and a sufficient search memory. In the following, a large value of M will be employed to test the presented decoders in terms of coding redundancy, while keeping the search space as large as possible. In particular, SA with $M = 4096$ will be selected to validate the performance of the MAP decoder, because of its better tradeoff between performance and complexity when the channel is not in extremely harsh conditions.

In Figs. 3–5, PER as a function of the channel SNR is plotted for $\epsilon = 0.05, 0.097, 0.185$, corresponding to coding rates $R_C = 8/9, 4/5$, and $2/3$, which represent three puncturing choices of the selected RCPC code. The performance of both hard (7) and soft (10) MAP estimation is compared with that of RCPC. It can be noticed that the MAP estimator exhibits a considerable coding gain of about 1 dB over RCPC codes, and that soft out-

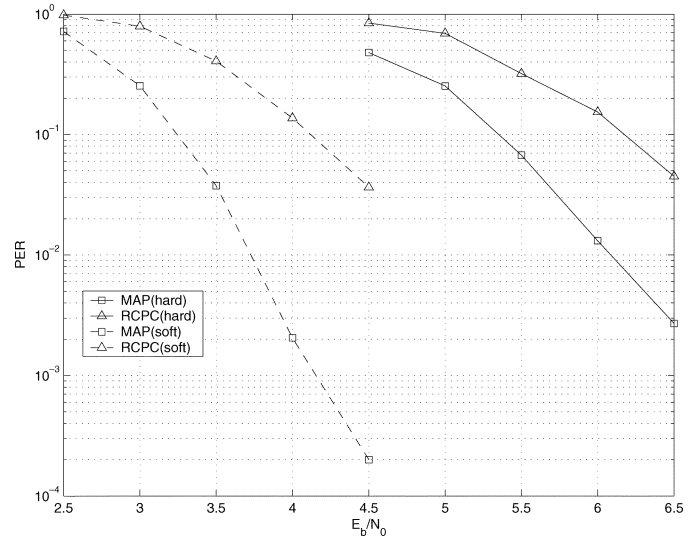


Fig. 3. PER versus E_b/N_0 for MAP decoding with SA and $\epsilon = 0.05$ (square markers) and RCPC with corresponding rate $R_C = 8/9$ (triangle); hard (solid); soft (dash) decoding.

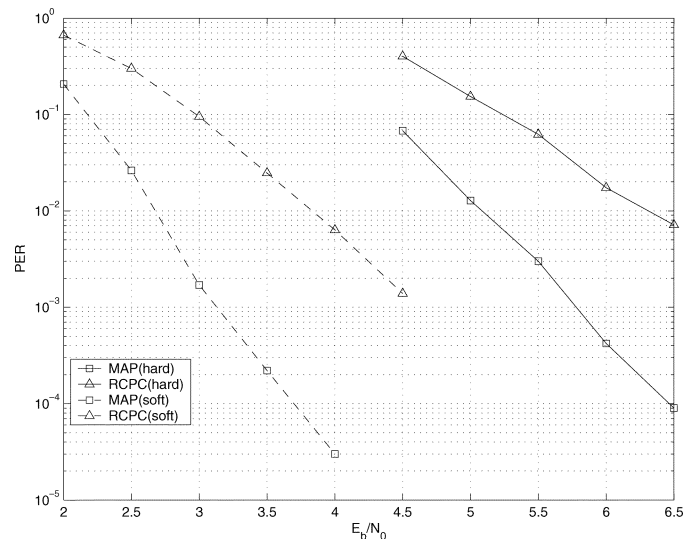


Fig. 4. PER versus E_b/N_0 for MAP decoding with SA and $\epsilon = 0.097$ (square markers) and RCPC with corresponding rate $R_C = 4/5$ (triangle); hard (solid); soft (dash) decoding.

performs hard decoding by about 2 dB. The excellent performance of the proposed system is due to the JSCC approach,

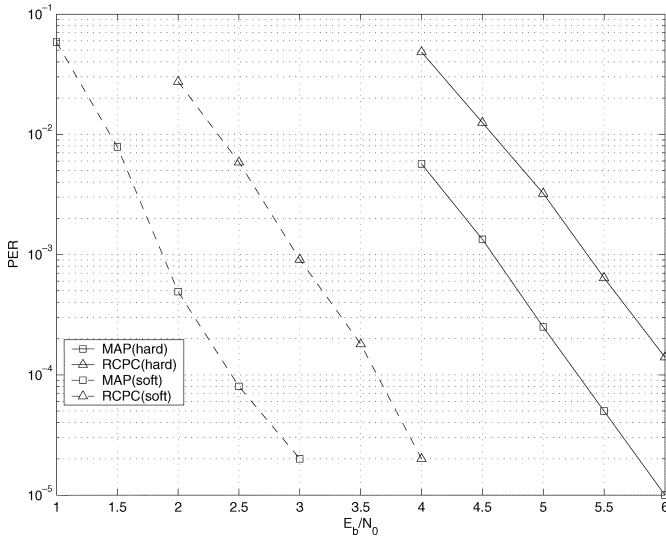


Fig. 5. PER versus E_b/N_0 for MAP decoding with SA and $\epsilon = 0.185$ (square markers) and RCPC with corresponding rate $R_C = 2/3$ (triangle); hard (solid); soft (dash) decoding.

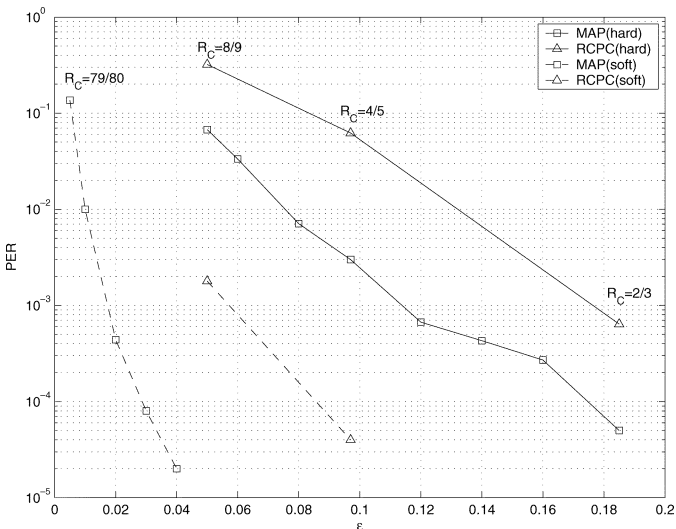


Fig. 6. PER versus ϵ for MAP decoding (square markers) and RCPC (triangle) in the case $E_b/N_0 = 5.5$ dB; hard (solid); soft (dash) decoding.

which allows one to integrate the source knowledge provided by the AC source model, and the efficient continuous error detection obtained with the forbidden symbol, in order to perform forward error correction, even in the absence of explicit channel coding. Moreover, the implemented system demonstrates the feasibility of both hard and soft MAP estimation at a reasonable computational cost.

Furthermore, the proposed JSCC approach allows a fine grain coding rate scalability. In fact, any desired coding rate can be achieved by selecting the proper value of ϵ ; on the other hand, RCPC codes are constrained to a limited number of puncturing patterns. In Fig. 6, PER as a function of ϵ is shown for $E_b/N_0 = 5.5$ dB; hard (solid square) and soft (dash square) MAP estimators are compared with hard (solid triangle) and soft (dash triangle) RCPC decoding. The advantage in terms of rate scalability is clear and, for instance, the soft MAP decoder can work with a very low level of coding redundancy; the lowest sim-

ulated value of ϵ is 5×10^{-3} , corresponding to a coding rate $R_C = 79/80$, unfeasible by means of RCPC codes.

Finally, our results are compared to previous work [19]. As already mentioned, the JSCC scheme in [19] is based on memoryless symbol level AC, where symbols are differences between neighboring pixels, represented by 8 bits. On the other hand, in the present paper we use binary AC on similar data,¹ after a proper binary mapping. In order to make a fair comparison between the two systems, we present a series of experiments, addressing the same channel coding rates. In [19], the channel coding rate is approximately $R_C = 4.6/(4.6 + R_\mu)$, where 4.6 bpp is the image coding rate in the absence of the forbidden symbol, and $R_\mu = -\log_2(1-\epsilon)$ is the coding redundancy measured in bpp. This formula for R_C takes into account that the forbidden symbol is added to a 256-symbol input alphabet, as opposed to the binary alphabet considered in this paper. The coding rate achieved by our scheme is $R_C = 5.1/(5.1 + 9R_\mu)$, where 5.1 bpp is the image coding rate and $9R_\mu$ is the redundancy per pixel, recalling that each prediction error is mapped to 9 bits. In the following, we consider R_C equal to $\{0.82, 0.9, 0.95\}$, corresponding to $\epsilon = \{0.08, 0.04, 0.02\}$ in our scheme and to $\epsilon = \{0.5, 0.29, 0.16\}$ in [19]. Two sequential decoding approaches are proposed in [19], which require knowledge of the soft demodulated values at the decoder side. BPSK signalling is employed and the number of branching points in the binary tree is limited according to a threshold Δ , in order to select the most likely erroneous positions in the received codeword. A depth first algorithm, based on error detection and adaptive selection of the Δ value, and a breadth-first technique, based on MA and Euclidean distance, are introduced. It is worth noticing that the second approach corresponds to the ML criterion, which does not take into account the *a priori* knowledge of the source, as in the MAP metric proposed in this paper. It is well known that MAP decoding can be reduced to ML decoding in the case of equally likely source symbols; however, AC is employed to compress non uniformly distributed symbols, and thus ML decoding turns out to be suboptimal in most cases. A certain computational overhead is required to evaluate the MAP metric (2). This amount to L supplementary multiplications for each explored path in the decoding tree; nevertheless, the added complexity is negligible with respect to the computational cost of the sequential search algorithm, dominated by the sorting of the M stored paths. In Table II, the packet recovery rates obtained by means of our proposed MAP estimators, with both hard and soft decoding, are compared with those in [19] as a function of p . We point out that the results were obtained on different images for the two systems; nevertheless, significant system parameters such as the packet length are the same. In the soft decoding case, the proposed MAP decoder, based on SA, outperforms both techniques in [19]. In particular, it is worth noticing the performance gap between the proposed MAP and the breadth first ML decoder in [19]. The results in the case of hard decoding are reported for sake of comparison and they confirm that the proposed approach exhibits a good performance also in the absence of soft demodulated values, which are required by both algorithms in [19].

¹The pixel predictor used in [19] is not given in the paper.

TABLE II
PACKET RECOVERY RATES (%) FOR THE PROPOSED MAP ESTIMATOR
WITH SOFT AND HARD DECODING, AND DEPTH FIRST (D.F.) AND
BREADTH FIRST (B.F.) ALGORITHMS IN [19]

R_C	Algorithm	p			
		$10^{-1.5}$	10^{-2}	$10^{-2.5}$	10^{-3}
0.82	MAP - hard	0.00	77.20	99.63	99.99
	MAP - soft	82.76	99.99	99.99	99.99
	[19] - D.F.	0.19	88.23	99.51	99.66
	[19] - B.F.	73.24	99.33	99.95	99.99
0.9	MAP - hard	0.0	22.90	90.63	98.54
	MAP - soft	16.53	99.80	99.99	99.99
	[19] - D.F.	0.00	71.09	99.21	99.56
	[19] - B.F.	16.63	99.30	99.95	99.99
0.95	MAP - hard	0.00	2.88	59.27	97.29
	MAP - soft	0.00	94.47	99.98	99.99
	[19] - D.F.	0.00	17.04	95.94	99.17
	[19] - B.F.	0.00	92.03	99.94	99.99

VI. RESILIENT SPIHT IMAGE TRANSMISSION

The experimental results presented in this section deal with the reliable transmission of lossy compressed images across the AWGN channels. The popular SPIHT [30] codec is employed and the obtained progressive stream is encoded by means of binary memoryless AC with a forbidden symbol and transmitted across the error prone channel. SPIHT is an efficient encoder, and therefore AC offers little compression gain and is mainly employed to provide robustness against transmission error.

The proposed system preserves SPIHT progressiveness, since no packetization is required. This feature can be profitably exploited at the decoder side; when the MAP sequential estimator is not able to correctly terminate the decoding, because of the implemented suboptimal search strategy, a certain number of reliable bits can still be forwarded to the SPIHT decoder, yielding an image quality that increases with the available rate. A certain number of bits decoded just before the error detection are discarded, since they are likely to contain residual errors; the number n of flushed bits depends on ϵ and is evaluated as $n = \log_2(1 - \delta) / \log_2(1 - \epsilon)$ [17], where δ represents the desired probability that all decoding errors are discarded. In our implementation we selected $(1 - \delta) = 10^{-12}$.

The simulation results, reported in the following, are obtained on the test image *GIRL*, when the overall coding rate, which includes the forbidden symbol redundancy, is fixed to 0.25 bpp. The employed MAP decoder is based on SA with $M = 4096$, and the performance is measured in terms of the average decoded PSNR over 1000 independent image transmissions. It is worth noticing that, in the case of the low correlated SPIHT bitstream, the gap between ML and MAP decoding could be highly reduced. Nevertheless, when the value of ϵ is not sufficient to assure a fast pruning of the decoding tree, or when the transmissive channel is particularly noisy, the *a priori* term turns out to significantly improve the performance. As an example, ML and MAP decoding yield an average PSNR of 28.31 and 29.61 dB, respectively, in the case $\epsilon = 0.2$ and hard decoding with $p = 10^{-2}$; in

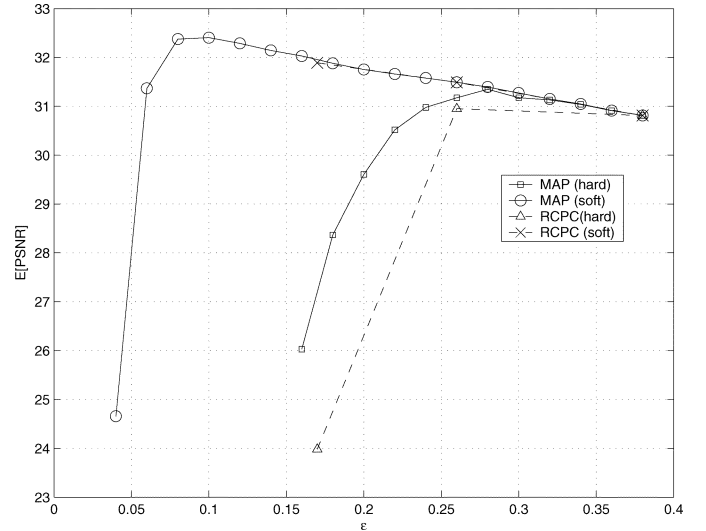


Fig. 7. Average PSNR as a function of ϵ in the case $E_b/N_0 = 4.32$ dB; MAP estimator with hard (square) and soft (circle) metric is compared versus RCPC/CRC scheme in the case of hard (triangle) and soft (cross) decoding.

the case of a larger value of $\epsilon = 0.28$ ML and MAP decoding yield almost the same performance with an average PSNR of 31.30 and 31.35 dB, respectively. In the following, the proposed MAP decoding approach is compared with the technique in [24], where the SPIHT bitstream is fragmented into small packets of 200 bits, and each packet, followed by a 16-bit CRC, is protected by means of the same RCPC code with memory 6 employed in the previous section. The decoder is implemented by a Viterbi list algorithm, where the outer CRC enables error detection.

In Fig. 7 the average PSNR is reported as a function of ϵ in the case of transmission across an AWGN channel with $E_b/N_0 = 4.32$ dB, yielding a bit transition probability $p = 10^{-2}$. The results obtained by means of hard (square) and soft (circle) MAP decoding are shown and superimposed on those achieved by the concatenated RCPC/CRC scheme with both hard (triangle) and soft (star) decoding. The advantages of the proposed MAP decoder are manifold. First of all, the coding rate can be adjusted freely, and therefore the best tradeoff between source and channel coding can be selected for the given channel conditions and overall transmitted rate; conversely, the RCPC/CRC approach allows only a coarse rate adjustment, in our simulation limited to values of $R_C = 0.8, 0.72$ and 0.6 , respectively. Moreover, the proposed system does not require packetization, which impacts on the SPIHT coding rate granularity. Finally, in the reported plots both hard and soft MAP decoders exhibit a gain of about 0.4 dB in terms of average PSNR, over the RCPC/CRC system. In particular, the best soft MAP decoder performance is obtained with a value of ϵ as low as 0.1, corresponding to an equivalent coding rate $R_C = 0.8857$.

In Fig. 8, the results obtained in the case $E_b/N_0 = 6.79$ dB, corresponding to $p = 10^{-3}$, are shown. With such a high channel SNR, the soft RCPC code decoding does not improve considerably on hard decoding, since both hard and soft decoding are able to correct all the errors in each transmitted image. On the other hand, the JSCC system allows one to employ a very small amount of coding redundancy, raising the performance gap to more than 1 dB in terms of decoded image

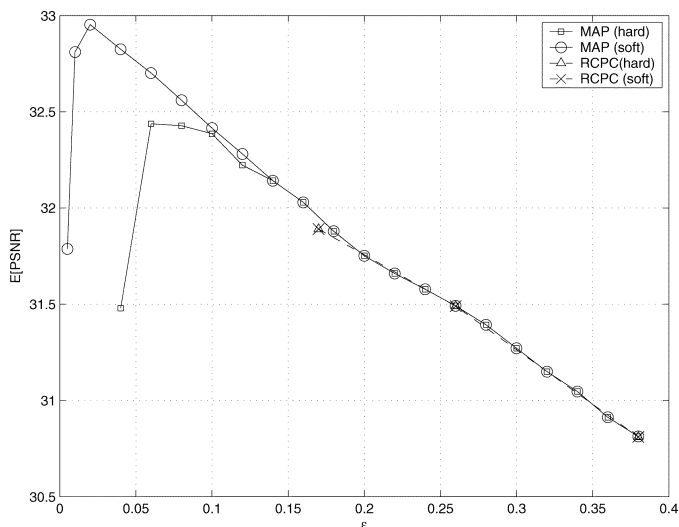


Fig. 8. Average PSNR as a function of ϵ for $E_b/N_0 = 6.79$ dB; MAP estimator with hard (square) and soft (circle) metric is compared versus RCPC/CRC scheme in the case of hard (triangle) and soft (cross) decoding.

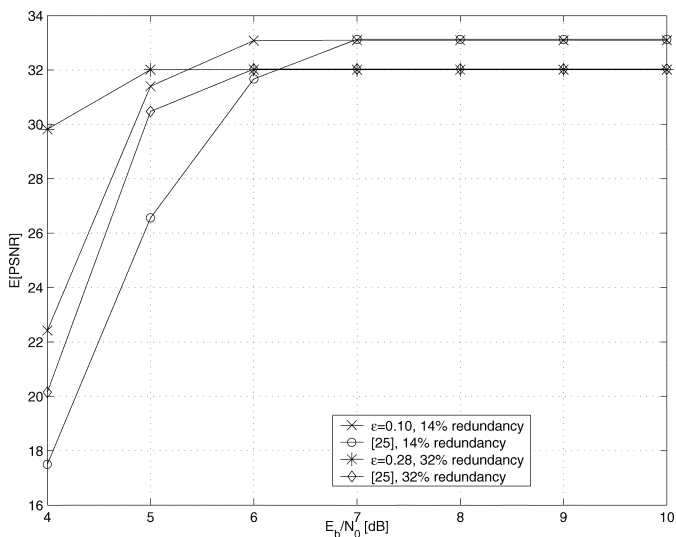


Fig. 9. Average PSNR as a function of E_b/N_0 obtained with the proposed MAP decoder and soft decoding technique [25].

quality in the case of soft MAP decoding. In fact, the soft MAP decoder optimal performance is achieved with $\epsilon = 0.02$, corresponding to $R_C = 0.9939$, which yields an average PSNR of 32.95 dB to be compared with the best performance of the RCPC/CRC, amounting to 31.89 dB.

Finally, as mentioned in the introduction, we compare the proposed technique with a similar approach which contemporarily appeared in [25]. In [25], the MAP decoding approach is applied to AC with soft synchronization markers; these latter are resynchronization patterns, which are inserted in the symbol sequence at some known positions in order to provide error detection at the decoder side. In Fig. 9 the average PSNR obtained with the proposed MAP estimator with soft decoding is compared to results available in [25] with the same amount of coding redundancy. The results are worked out on the standard test image *Lena* 512×512 compressed at 0.5 bpp. In [25] the adopted image coder is JPEG 2000, whereas in the current paper SPIHT is used; nonetheless, the reported results are

comparable since the adopted codecs exhibit almost the same coding efficiency. In order to make a fair comparison, the MA technique with $M = 20$, as reported in [25], is employed for sequential pruning of the decoding tree. The results reported in Fig. 9 show that the proposed system significantly outperforms [25]; this allows us to claim that the proposed approach based on continuous error detection, yields better performance than error detection obtained with periodic synchronization markers.

VII. CONCLUSIONS

In this paper, we have proposed novel MAP decoding techniques for AC with a forbidden symbol. The approach has been tested for image transmission across an AWGN channel with both hard and soft decoding in the case of BPSK signalling. The system represents a JSCC approach and shows a number of significant advantages. First, the encoder exhibits the same complexity as traditional AC, whereas the MAP decoder computational load can be scaled according to memory and decoding delay constraints. Second, MAP decoding of AC achieves an error correction performance that outperforms standard RCPC codes. In the case of lossless image compression, we are able to provide both compression and error protection with the same tool. Moreover, the proposed system has been profitably exploited in the case of SPIHT image transmission, where it preserves coding progressiveness. Finally, the amount of coding redundancy can be finely adjusted, simply acting on the probability ϵ attributed to the forbidden symbol.

Future work includes the generalization of the proposed technique to adaptive AC. In fact, the concept of adaptiveness can be applied not only to the source model but also to the amount of coding redundancy, thus designing a joint and adaptive source channel coding system. Moreover, the proposed approach is being applied to concatenated schemes, where iterative decoding is taken into account.

ACKNOWLEDGMENT

The authors would like to thank the Associate Editor and the anonymous reviewers for their valuable comments and suggestions that have highly improved the quality of the paper.

REFERENCES

- [1] S. Vembu, S. Verdù, and Y. Steinberg, "The source channel theorem revisited," *IEEE Trans. Inf. Theory*, vol. 41, no. 1, pp. 44–54, Jan. 1995.
- [2] 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.
- [3] 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.
- [4] K. P. Subbalakshmi and J. Vaisey, "On the joint source/channel coding of variable-length encoded sources: The BSC case," *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2052–2055, Dec. 2001.
- [5] M. Bystrom, S. Kaiser, and A. Kopansky, "Soft source decoding with applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 10, pp. 1108–1120, Oct. 2001.
- [6] M. Jeanne, J. Carlach, P. Siohan, and L. Guivarch, "Source and joint source-channel decoding of variable length codes," in *Proc. IEEE Int. Conf. Communications*, 2002, pp. 768–772.
- [7] J. Hagenauer and R. Bauer, "The turbo principle in joint source channel decoding of variable length codes," in *Proc. Information Theory Workshop*, 2001, pp. 33–35.
- [8] *JPEG2000 Image Compression Standard*, I. 15 444–1, 2000.

- [9] *JBIG2 Lossy/Lossless Coding of Bi-Level Images*, I. 14 492, 2001.
- [10] *Joint Committee Draft (CD)*, J. V. T. J. of ISO/IEC MPEG and I-T. VCEG, 2002.
- [11] I. H. Witten, J. G. Cleary, and R. Neal, "Arithmetic coding for data compression," *ACM Commun.*, no. 6, pp. 520–540, Jun. 1987.
- [12] G. Elmasry, "Joint lossless-source and channel coding using automatic repeat request," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 953–955, Jul. 1999.
- [13] —, "Embedding channel coding in arithmetic coding," *Proc. Inst. Elect. Eng.—Commun.*, vol. 146, no. 2, pp. 73–78, Apr. 1999.
- [14] G. Elmasry and Y. Shi, "MAP symbol decoding of arithmetic coding with embedded channel coding," in *Proc. Wireless Communications and Networking Conf.*, 1999, pp. 988–992.
- [15] H. Chen, "Joint error detection and VF arithmetic coding," in *Proc. IEEE Int. Conf. Communications*, vol. 9, 2001, pp. 2763–2767.
- [16] C. Boyd, J. Cleary, S. Irvine, I. Rinsma-Melchert, and I. Witten, "Integrating error detection into arithmetic coding," *IEEE Trans. Commun.*, vol. 45, pp. 1–3, Jan. 1997.
- [17] J. Chou and K. Ramchandran, "Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 861–867, Jun. 2000.
- [18] 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, Sep. 2001.
- [19] B. Pettijohn, M. Hoffman, and K. Sayood, "Joint source/channel coding using arithmetic codes," *IEEE Trans. Commun.*, vol. 49, no. 9, pp. 1540–1548, Sep. 2001.
- [20] K. Sayir, "On coding by probability transformation," Ph.D. dissertation, Dept. Inf. Technol. Eng., Swiss Federal Inst. Technology (ETH), Zurich, Switzerland, 1999.
- [21] C. Demiroglu, M. Hoffman, and K. Sayood, "Joint source channel coding using arithmetic codes and trellis coded modulation," in *Proc. DCC*, Mar. 2001, pp. 302–311.
- [22] M. Grangetto and P. Cosman, "MAP decoding of arithmetic codes with a forbidden symbol," in *Proc. ACIVS*, Ghent, Belgium, Sep. 2002, pp. 82–89.
- [23] M. Grangetto, G. Olmo, and P. Cosman, "Image transmission by means of arithmetic codes with forbidden symbol," in *Proc. ICASSP*, Hong Kong, Apr. 2003, pp. 273–276.
- [24] O. G. Sherwood and K. Zeger, "Progressive image coding for noisy channels," *IEEE Signal Process. Lett.*, vol. 4, no. 7, pp. 189–191, Jul. 1997.
- [25] T. Guionnet and C. Guillemot, "Soft decoding and synchronization of arithmetic codes: Application to image transmission over noisy channels," *IEEE Trans. Image Process.*, vol. 12, no. 12, pp. 1599–1609, Dec. 2003.
- [26] J. Proakis and M. Salehi, *Communication Systems Engineering*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
- [27] J. B. Anderson and S. Mohan, *Source and Channel Coding*. Norwell, MA: Kluwer, 1991.
- [28] *Lossless and Near-Lossless Compression of Continuous-Tone Still Images*, I. 14 495, 2000.
- [29] J. Hagenauer, "Rate-compatible punctured convolutional codes (RCP codes) and their applications," *IEEE Trans. Commun.*, vol. 36, no. 4, pp. 389–400, Apr. 1988.
- [30] 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. 6, pp. 243–250, Jun. 1996.



Marco Grangetto (S'99–M'03) received the M.S. degree (*summa cum laude*) in electrical engineering and the Ph.D. degree from Politecnico di Torino, Torino, Italy, in 1999 and 2003, respectively.

He is currently a Postdoctoral Researcher at the Image Processing Laboratory, Politecnico di Torino. He was awarded the premio Optime by Unione Industriale di Torino in September 2000, and a Fulbright grant in 2001 for a research period with the Department of Electrical and Computer Engineering, University of California, San Diego. His research interests are in the fields of multimedia signal processing and communications. In particular, his expertise includes wavelets, image and video coding, data compression, video error concealment, error resilient video coding unequal error protection, and joint source channel coding. He is currently participating in the ISO activities on JPEG 2000 (Part 11, wireless applications).



Pamela Cosman (S'88–M'93–SM'00) received the B.S. degree (hons.) in electrical engineering from the California Institute of Technology, Pasadena, in 1987, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, in 1989 and 1993, respectively.

She was a National Science Foundation Postdoctoral Fellow at Stanford University and a Visiting Professor at the University of Minnesota, Twin Cities, during 1993–1995. In 1995, she joined the faculty of the department of Electrical and Computer

Engineering at the University of California at San Diego, La Jolla, where she is currently Professor and Co-Director of the Center for Wireless Communications. Her research interests are in the areas of image and video compression and processing.

Dr. Cosman is the recipient of the ECE Departmental Graduate Teaching Award (1996), a Career Award from the National Science Foundation (1996–1999), and a Powell Faculty Fellowship (1997–1998). She was an Associate Editor of the IEEE COMMUNICATIONS LETTERS (1998–2001), a Guest Editor of the June 2000 special issue of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS on "Error-resilient image and video coding," and was the Technical Program Chair of the 1998 Information Theory Workshop in San Diego. She is currently an Associate Editor of the IEEE SIGNAL PROCESSING LETTERS, and a Senior Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She is a member of Tau Beta Pi and Sigma Xi.



Gabriella Olmo (S'89–M'91) received the Laurea (*cum laude*) and Ph.D. degrees in electronic engineering from the Politecnico di Torino, Torino, Italy.

She is currently Associate Professor at the Politecnico di Torino. Her main recent interests are in the fields of wavelets, remote sensing, image and video coding, resilient multimedia transmission, joint source-channel coding, and distributed source coding. She has coordinated several national and international research programs in the fields of wireless multimedia communications, under contracts by the European Community and the Italian Ministry of Education. She has coauthored more than 110 papers in international technical journals and conference proceedings. She has been a member of the Technical Program Committee and session chair for several international conferences.

Dr. Olmo is Member of IEEE Communications Society and IEEE Signal Processing Society, and is part of the Editorial Board of IEEE TRANSACTIONS ON SIGNAL PROCESSING.