EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Joint source/channel iterative arithmetic decoding with JPEG 2000 image transmission application

Sonia Zaibi[1], Amin Zribi[1,2*], Ramesh Pyndiah[2] and Nadia Aloui[1]

Abstract

Motivated by recent results in Joint Source/Channel coding and decoding, we consider the decoding problem of Arithmetic Codes (AC). In fact, in this article we provide different approaches which allow one to unify the arithmetic decoding and error correction tasks. A novel length-constrained arithmetic decoding algorithm based on Maximum *A Posteriori* sequence estimation is proposed. The latter is based on soft-input decoding using *a priori* knowledge of the source-symbol sequence and the compressed bit-stream lengths. Performance in the case of transmission over an Additive White Gaussian Noise channel is evaluated in terms of Packet Error Rate. Simulation results show that the proposed decoding algorithm leads to significant performance gain while exhibiting very low complexity. The proposed soft input arithmetic decoder can also generate additional information regarding the reliability of the compressed bit-stream components. We consider the serial concatenation of the AC with a Recursive Systematic Convolutional Code, and perform iterative decoding. We show that, compared to tandem and to trellis-based Soft-Input Soft-Output decoding schemes, the proposed decoder exhibits the best performance/complexity tradeoff. Finally, the practical relevance of the presented iterative decoding system is validated under an image transmission scheme based on the JPEG 2000 standard and excellent results in terms of decoded image quality are obtained.

1 Introduction

Joint Source/Channel (JSC) coding and decoding have become an area of strong interest because the separation between source and channel coding has turned out to be unjustified in practical systems due to limited block lengths and the residual redundancy in the data bits which remain after source encoding.

On the other hand, the hostile nature of the communication channel requires some form of error protection. The limitation on the bandwidth necessitates the use of efficient entropy-approaching source codes, generally, variable-length codes (Huffman or arithmetic codes–ACs). However, variable-length compressed bit-streams are susceptible to error propagation, and the need for error protection increases. In this scope, JSC decoding for variable-length coding is receiving increasing attention.

Arithmetic Coding [1,2] is currently being deployed in a growing number of compression standards such as JPEG 2000 [3] for still pictures and H264 [4] for video sequences. Arithmetic coding yields higher compression performance when compared to other lossless compression methods since it can allocate fractional numbers of bits to input symbols. How-ever, the arithmetic decoder has poor resynchronization properties which motivated the development of JSC techniques based on ACs.

The first contributions, classified as resilient entropy coding techniques, tends to prevent error propagation by using proper synchronization markers or considering some framing techniques [5]. A scheme reserving a space for an extra symbol (called forbidden symbol) that is not in the source alphabet, so never transmitted, has been proposed in [6]. The technique has been shown to provide excellent error detection while introducing redundancy in the compressed bit-stream. However, this extra rate is small considering the error detection capability. The forbidden symbol technique was then integrated in different decoding schemes to provide error

* Correspondence: amin.zribi@gmail.com
[1]Syscom Laboratory, National Engineering School of Tunis, BP 37, Le belvedere, 1002, Tunis, Tunisia
Full list of author information is available at the end of the article

correction. Associated to an automatic repeat request (ARQ) protocol, the technique was used for error correction in [7,8].

More recent studies tend to apply soft-input sequence estimation algorithms for arithmetic decoding instead of hard-input classical decoding. The proposed schemes consider finite state representations for the arithmetic decoding machine to apply well-known algorithms used in channel decoding such as Viterbi, List-Viterbi [9-15]. In [9], an AC that embeds channel coding is presented to enforce a minimum Hamming distance between encoded sequences, then a Maximum *A Posteriori* (MAP) estimator is proposed for arithmetic decoding. In [10,11], sequential decoding schemes were applied on binary trees and path pruning technique was used based on the forbidden symbol error detection. Sayir [12] used an arithmetic encoder adding redundancy in the compressed bit-stream by introducing gaps in the coding space, and performs sequential decoding. In [13], authors used Bayesian networks to model the quasi-arithmetic encoder, and considered adding redundancy by introducing synchronization markers. A new three-dimensional bit-synchronized trellis representing the finite precision AC was recently proposed in [14]. A similar trellis was used in [15], where authors computed bounds on the error probability obtained with an AC using the forbidden symbol technique.

Recent contributions proposed to exploit the efficiency of turbo decoding by integrating the arithmetic decoder in an iterative decoding process [13,16-18]. These contributions used finite-state machine representations for the AC to apply Soft-Input Soft-Output (SISO) decoding algorithms. Iterative decoding is then performed by concatenating the ACs with a Recursive Systematic Convolutional Code (RSCC). It is worth pointing out that the techniques introduced in [13,16] were applied for JPEG 2000 compressed image transmission. We notice that all the proposed techniques rely on specific trellis constructions with eventually pruning, which results in various efficiencies in terms of error correction performance. However, in the proposed trellises the states number increases with the source symbol sequence length L, and the source alphabet size U. Thus, the decoding complexity becomes intractable for large values of L and U. Recently, some contributions considered JSC decoding methods including a Low-Density Parity-Check Code (LDPC) for channel coding with application to image transmission. In [19], the authors considered rate-compatible LDPC codes to apply unequal error protection on the compressed JPEG 2000 bit-stream. Extra information provided by the error-resilience mode of the JPEG 2000 encoder was delivered to the LDPC decoder and iterative decoding was performed in [20].

This article is devoted to a different decoding algorithm, with low-complexity, for soft-input decoding of ACs in the case of transmission over a noisy channel. The main objective is the development of a SISO arithmetic decoder that is able to improve the error correction performance with a reasonable complexity and efficient compression behavior. First, we propose a new low-complexity arithmetic decoder that supposes the decoder to know the source symbol sequence length L and the compressed bit-stream size l. Then, the decoding task is based on the search of the MAP sequence among length-valid sequences (bit-streams of length l decoding exactly L symbols). The proposed algorithm is inspired from the Chase algorithm [21] and called Chase-like arithmetic decoder. The second contribution of this study is a new scheme for SISO arithmetic decoding. The latter is obtained through a slight modification of the Chase-like arithmetic decoder and generates additional bits reliability measure. Results corresponding to iterative decoding in the case of serial concatenation of an AC with a RSCC are presented and compared to the tandem decoding and to a trellis-based iterative decoding scheme [17,18]. The last major contribution of the article is the implementation of the proposed SISO arithmetic decoder within the JPEG 2000 decoder and the analysis of the improvements obtained by iterative JSC decoding. In fact, the proposed SISO arithmetic decoder is applied to the JPEG 2000 entropy encoding stage which uses an adaptive binary AC (MQ coder).

The article is organized as follows. Section 2 briefly introduces the principles of arithmetic coding. In Section 3, the system model and the MAP sequence decoding metric are reported. The Chase-like arithmetic decoder is also detailed in this section and its performances are compared with a recent solution using a trellis representation of the AC with Viterbi-like decoding [15]. Section 4 addresses a new scheme for low-complexity SISO arithmetic decoding. Numerical results corresponding to iterative JSC are discussed and compared to tandem decoding and trellis-based arithmetic decoding presented in [17,18]. In Section 5, the application of the proposed iterative decoding approach to a JPEG 2000 image communication system is described. Finally, Section 6 draws our conclusions and offers directions for future work.

2 Overview of arithmetic coding

The objective of the arithmetic encoder is to map a sequence of symbols s = $\{s_{1,}\ s_2,\dots,\ s_L\}$ onto a binary string b that represents the probability of the input sequence [1]. The encoder performs this mapping based on the available source model. In the following, we will consider the case of a binary memoryless source fully described by the probabilities $p_0 = \Pr(s_k = 0)$ and $p_1 = \Pr(s_k = 1)$. Arithmetic encoding is performed by recursively computing the probability interval I(s) = [Low, High) corresponding to the input string. At initialization, I(s) is set to $I_0 = [0, 1)$. Then, for every recursion k, $0 \leq k < L$, the update of I_k is done as follows:

$$\text{High}_k = \text{Low}_{k-1} + \left(\text{High}_{k-1} - \text{Low}_{k-1}\right) K_{sk}, \qquad (1)$$

$$\text{Low}_k = \text{Low}_{k-1} + \left(\text{High}_{k-1} - \text{Low}_{k-1}\right) K_{s_{k-1}}, \qquad (2)$$

where K_{s_k} is the cumulative probability of symbol s_k given by $K_{s_k} = \sum_{i=0}^{i=k-1} p_i$. A coding example is presented in Figure 1 with a binary source alphabet with probabilities $p_0 = 0.7$ and $p_1 = 0.3$. We represent the interval partitioning corresponding to the encoding of the sequence s = 010.

Once all L symbols have been processed, the output **b** corresponds to the shortest binary string contained in I (s). Decoding follows the dual process.

For long source sequences, such an algorithm needs an infinite precision machine (Low and High get smaller and smaller while encoding). In [2], the authors proposed a new implementation that made arithmetic coding feasible in practice. They used integer representations for probabilities with scaling techniques. The initial interval [0, 1) was substituted by [0, W), where $W = 2^p$, $p >= 2$ being the bit size of the initial interval. The scaling is performed by doubling the size of the interval I(s) = [Low, High) when one of the following conditions holds:

- E1: $0 \leq$ High $< W/2$: We double Low and High and we output 0 followed by U_3 ones, then U_3 is reset to 0.
- E2: $W/2 \leq$ Low $< W$: We double Low and High after subtracting $W/2$ and we output 1 followed by U_3 zeros, then U_3 is reset to 0.
- E3: $W/4 \leq$ Low $< W/2 \leq$ High $< 3W/4$: We double Low and High after subtracting W/4 and we increase U_3 by 1 (no output).

Note that U_3 represents the number of the last E3 scalings done, and is initialized to 0.

The described AC is based on the binary source statistics, and it is essential that, for an encoded symbol index i, the encoder and the decoder use the same probabilities p_0 and p_1. Static arithmetic coding supposes that source statistics were transmitted to the decoder with no error, which results in additional bits and consequently a compression loss. Such situation is outperformed with adaptive arithmetic coding, where p_0 and p_1 are initialized to 0.5, then, for every symbol encoding step they are updated. Such scheme induces no remarkable compression loss when long source symbol sequences are used [2]. In the following, we address new soft-input decoding scheme which can be applied to both adaptive and static ACs.

To manage AC sensitivity to errors, authors of [6] proposed to use an extra symbol μ with probability $\varepsilon > 0$ to detect transmission errors. This symbol is introduced in the source alphabet but never transmitted. The forbidden symbol technique implies a reduction of the coding space by a factor of (1-ε), thus, reduces compression efficiency. The amount of added rate redundancy is

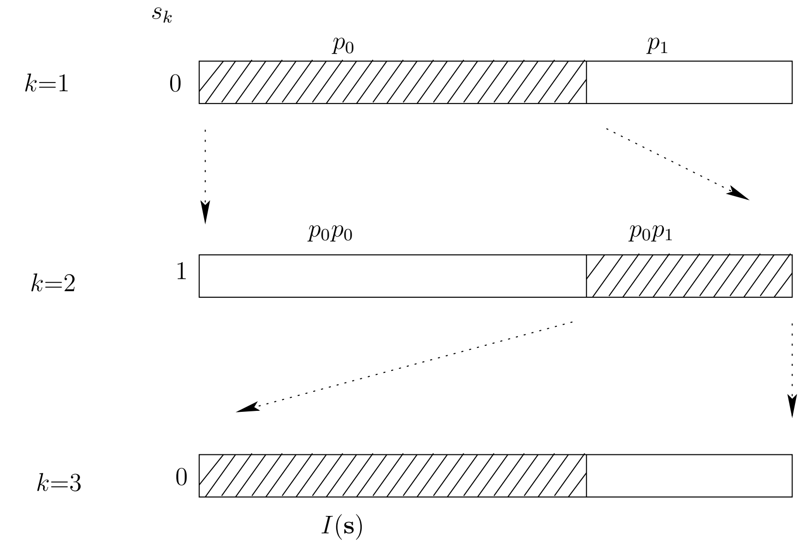


Figure 1 Arithmetic encoding example with a binary source (p_0 = 0.7) and s = 010.

R_{ac} = -log$_2$ (1-ε) bits/symbol [7]. In the presence of transmission error, due to the low resynchronization probability, the decoder will reveal a forbidden symbol after a delay that is inversely proportional to ε.

3 Low-complexity soft-input arithmetic decoding method

3.1 Soft-input arithmetic decoding

The considered system consists of a finite alphabet source, a classical arithmetic encoder, an Additive White Gaussian Noise (AWGN) channel and an arithmetic decoder. The source generates packets of L symbols s = (s_1,..., s_L). Each packet is, then, compressed using the arithmetic encoder and the resulting binary stream $\mathbf{b}$ = (b_1,..., b_l) is transmitted over an AWGN channel. The latter delivers the sequence $\mathbf{r}$ = (r_1,..., r_l). A Binary Phase Shift Keying (BPSK) modulation is considered, then r_j can be written as $r_j = h_j + n_j$, j = 1,..., l, where h_j is the BPSK modulated value of b_j given by $h_j = \sqrt{E_b}\left(1 - 2.b_j\right)$ where E_b is the energy per information bit, and n_j is a Gaussian noise sample with zero mean and variance σ^2.

At the receiver, the arithmetic decoder is based on the MAP criterion which corresponds to the search of the best sequence s^i satisfying:

$$P(\mathbf{s}^i|\mathbf{r}) \geq P(\mathbf{s}^k|\mathbf{r})\, \mathit{forall}\, k \neq i \tag{3}$$

Therefore, the problem is equivalent to the maximization of the following metric

$$P(\mathbf{s}^k|\mathbf{r}) = \frac{P(\mathbf{r}|\mathbf{s}^k)\,.\,P(\mathbf{s}^k)}{P(\mathbf{r})} = \frac{P(\mathbf{r}|\mathbf{b}^k)\,.\,P(\mathbf{s}^k)}{P(\mathbf{r})} \tag{4}$$

where $\mathbf{b}^k = (b_1^k, \ . \ . \ . \ , b_1^k)$ is the bit-stream resulting from the arithmetic encoding of the source symbol sequence $\mathbf{s}^k = (s_1^k, \ . \ . \ . \ , s_L^k)$.

The metric (4) includes the channel transition probability $P(\mathbf{r}|\mathbf{b}^k)$, the *a priori* source probability P (s^k), and the term P(r). The latter can be ignored when comparing candidate sequences since it is constant for all candidates. In the case of an AWGN channel, the channel transition probability can be written

$$P(\mathbf{r}|\mathbf{b}^k) = \prod_{j=1}^{l} P(r_j|b_j^k) \tag{5}$$

where

$$P(r_j\,|\,b_j^k) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{d_E(r_j,\, h_j^k)^2}{2\sigma^2}\right) \tag{6}$$

and the function $d_E(x, y)$ denotes the Euclidean distance between x and y.

When taking the logarithm of (4) and using the equations (5) and (6), the problem can be reformulated as maximizing the following metric

$$M^k = \sum_{j=1}^{l} -\frac{d_E(r_j,\, h_j^k)^2}{2\sigma^2} + \log(P(\mathbf{s}^k)). \tag{7}$$

The exhaustive decoding approach calculates the metric M^k for all possible pairs (s^k, b^k) in order to select the best binary stream. In this study, we assume that the source symbol sequence length L and the compressed bit-stream length l are known at the decoder side. This information is used to detect invalid binary sequences in the search space. Therefore, the search space is limited to streams of l bits that yield exactly L symbols by arithmetic decoding. It is obvious that evaluating the MAP metric (7) for all possible candidates is infeasible. On the one hand, for practical values of l, it is impossible to store all the combinations of l bits, in order to select the valid sequences in terms of length. On the other hand, the evaluation of all valid sequences requires a huge decoding delay because of the large cardinality of the search space.

Thus, it is necessary to use a suboptimal decoding algorithm to reduce the search space size. In the following, we describe a suboptimal soft-input arithmetic decoder, inspired from the Chase algorithm [21] which was initially proposed for soft-input decoding of linear block codes.

3.2 The proposed decoding algorithm

The proposed decoding algorithm aims to find the compressed bit-stream that is length-valid, and whose corresponding decoded sequence has the best metric M^k. We use a Chase II-type algorithm [21] in order to achieve a low-complexity soft-input suboptimal arithmetic decoding. In fact, this algorithm reduces complexity by restricting the search space to only the Q most probable sequences.

We recall that the arithmetic decoding machine is based on a recursive procedure which terminates when all l bits are processed or when L symbols are obtained in the decoded sequence. However, the proposed decoder has to use information about L to detect erroneous sequences and improve decoding performance. To address this requirement, a proper AC termination strategy is implemented. In fact, the arithmetic encoder terminated each input sequence with an End-of-Block (EoB) symbol. The same rule is enforced at the decoder and only sequences that decode exactly L symbols and whose EoB symbol is determined by the last two bits is considered to be correct. This supplementary error detection tool can ameliorate the arithmetic decoder performance since it reduces the size of the search space and consequently, increases the Hamming distance between candidates.

A classical decoding scheme consists in applying an arithmetic decoding to binary sequence $\mathbf{y} = (y_1, \ldots, y_l)$ obtained by hard decision applied on the received values $\mathbf{r}$. The soft-input decoder uses the extra information given by r called reliability. The reliability of a component y_j is defined using the Log Likelihood Ratio (LLR) of decision y_j, which, for an AWGN channel, can be written

$$\Lambda(y_i) = \log \frac{P(b_j = 0 \mid r_j)}{P(b_j = 1 \mid r_j)} = \frac{2}{\sigma^2} r^j \tag{8}$$

For a stationary channel, the LLR can be normalized and the reliability of y_j is given by $|r_j|$. The proposed soft-input arithmetic decoding is as follows:

1. Evaluate the hard decision vector y = $(y_1, \ldots, y_l)$ and the corresponding reliability vector $\Lambda(\mathbf{y}) = (|y_1|, \ldots, |y_l|)$,
2. determine the positions of the q least reliable binary elements of y based on $\Lambda(\mathbf{y})$,
3. form test patterns $\mathbf{t}^i$, $0 < i \leq Q$ where $Q = 2^q$: all the l-elements binary vectors t_j^i, $0 < j \leq l$ having a maximum weight q with all the possible bit combinations in the least reliable positions,
4. form test sequences $\mathbf{z}^i$, $0 < i \leq Q$ with $z_j^i = y_j \oplus t_j^i$, for $0 < j \leq l$, where $\oplus$ is the XOR function,
5. decode all the test sequences $\mathbf{z}^i$ using classical arithmetic decoding. If a sequence $\mathbf{z}^i$ decode exactly L symbols and its EoB symbol is correct, we compute its metric using (7), and append it to the subset of competitor valid sequences ψ.
6. Finally, the decoded bit-stream corresponds to the sequence having the best metric M^k in the subset ψ.

3.3 Chase-like arithmetic decoding performance

The performances of the proposed soft-input arithmetic decoder are evaluated in terms of Packet Error Rate (PER) with different values of test pattern weight q = 1, 2, 4, and 8 bits. A memoryless quaternary source is considered with packets of L = 128 symbols according to the probabilities presented in Table 1. Each packet is encoded by a static arithmetic encoder. In all the experiments, the residual redundancy at the output of the arithmetic encoder will be considered as a coding rate R_{ac}. We consider the case where the source statistics are assumed to be transmitted to the decoder without error. The mean length achieved by the arithmetic encoder is $\bar{l} = 1.93$ bits per symbol resulting in coding rate R_{ac} = 0.9. As a reference, we evaluated the performance of the classical arithmetic decoder. The PER obtained for the described schemes as a function of the signal-to-noise ratio $\frac{E_b}{N_0}$ are depicted in Figure 2.

Table 1 A memoryless quaternary source with entropy = 1

Symbol k	Probability p_k
0	0.5
1	0.25
2	0.125
3	0.125

Figure 2 shows that the proposed soft-input arithmetic decoder improves the performance compared to the classical arithmetic decoder. In fact, with one additional AC decoding test sequence (q = 1 bit), we achieve a gain of 1.2 dB at a PER = 10^{-3}. Increasing the value of the test pattern weight q induces an improvement in performance reaching 2 dB at PER = 10^{-2}.

We also notice that for low to medium values of signal-to-noise ratio, increasing q enables remarkable performance amelioration. For high values of $\frac{E_b}{N_0}$, a low value of q is sufficient to achieve the maximum performance. In this case, increasing q results in increasing the complexity without improving the performance. In the following, we will use the value of q = 4 bits since it represents a good trade-off between performance and complexity. For a complexity of 16 AC hard decoding, q = 4 bits includes an improvement of 1.6 dB at PER = 10^{-3}.

3.4 Comparison with trellis-based arithmetic decoding

In this section, we propose to compare the proposed soft-input arithmetic decoder with a trellis-based arithmetic decoding scheme using the forbidden symbol technique with probability ε.

We consider a binary source delivering source sequences of L = 512 symbols according to the probabilities p_0 = 0.2 and p_1 = 0.8. The entropy of this source is H = 0.72 bits per symbol. The mean length achieved by a static arithmetic encoder is $\bar{l}$ = 0.87 bits per symbol inducing a coding rate R_{ac} = 0.93. The same coding rate is obtained when considering the system based on Chase-like decoding (R_{ac} = 0.93). However, the forbidden symbol probability ε causes more redundancy in the compressed bit-stream. Therefore, the R_{ac} value varies according to ^{2}. The R_{ac} values corresponding to ε = 0.0, and 0.2 are, respectively, 0.96 and 0.67. The trellis-based

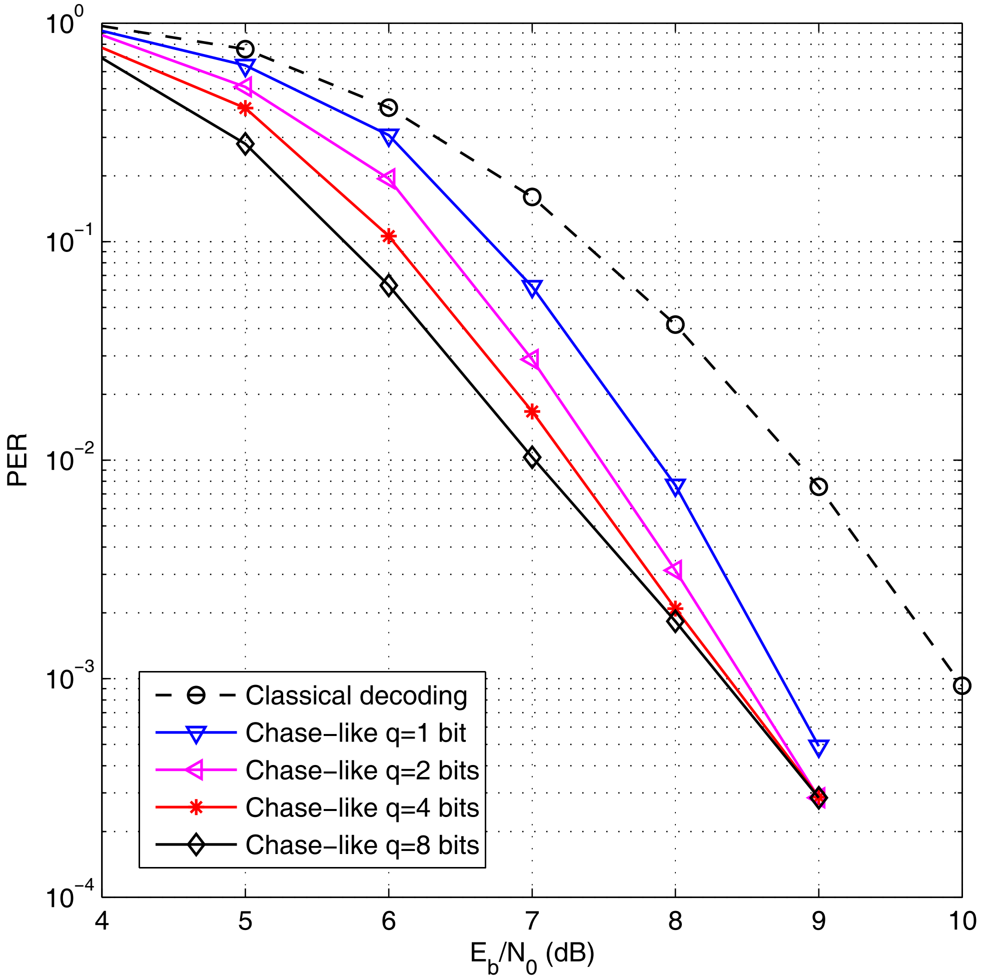


Figure 2 Performance in terms of PER as a function of $\frac{E_b}{N_0}$ of the proposed soft-input arithmetic decoding for different *q* values.

decoding uses the soft-input Viterbi algorithm based on the Maximum Likelihood (ML) criterion. The trellis-based reference scheme is presented in [15] where the finite precision static AC is described by a bit clock trellis. In Figure 3, the PER of the described schemes are reported versus signal-to-noise ratio $\frac{E_b}{N_0}$. The value of q = 4 bits is considered for Chase-like decoding.

The simulation results show that for a very noisy channel ($\frac{E_b}{N_0}$ < 7 dB) the arithmetic trellis-based decoding with ε = 0.2 is more efficient than the Chase-like arithmetic decoding. But, for a medium to high $\frac{E_b}{N_0}$ values, the Chase-like soft-input arithmetic decoder presents the best performance. In fact, it exhibits a considerable gain of about 1.1 dB over the best configuration of the trellis-based Viterbi decoding with ε = 0.2.

This gain is essentially due to the additional information used by the proposed Chase-like algorithm to perform arithmetic decoding. In fact, in the trellis proposed by Ben-Jamaa et al. [15], all binary paths of length l are valid candidates; the constraint on L is not considered. However, all possible candidates for Chase-like decoding are length-valid sequences, which results in a greater Hamming distance between competitors, and consequently better performance.

On the other hand, the Chase-like arithmetic decoding complexity is constant for increasing source sequence length L, and alphabet cardinal U. The decoding complexity can be approximated to 2^q arithmetic classical decoding operations and depends only on q. The states number representing the arithmetic encoding machine of [15] increases for bigger values of L and U.

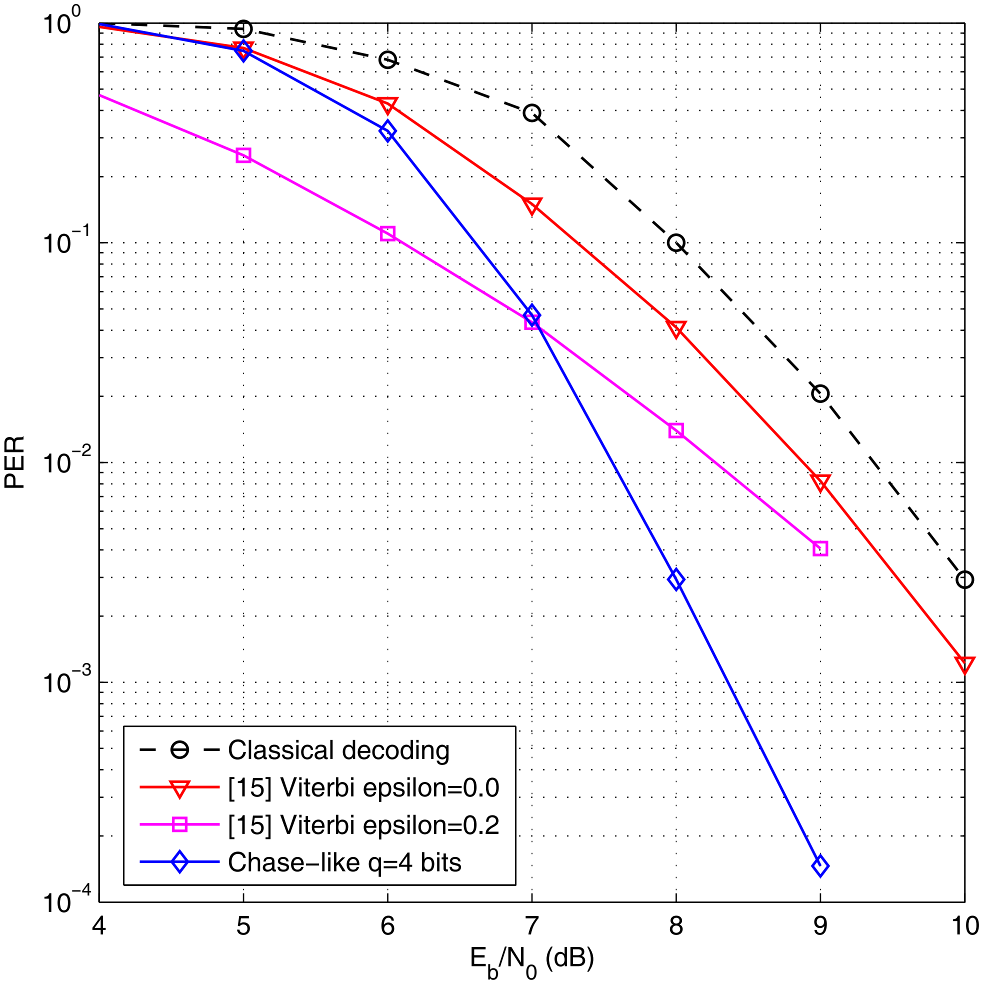


Figure 3 Performance in terms of PER as a function of $\frac{E_b}{N_0}$ for the proposed soft-input arithmetic decoder and trellis-based arithmetic decoding of [15].

Furthermore, the trellis construction needs the transmission of the source statistics as side information. Consequently, trellis-based decoding is very hard to apply with adaptive AC (the trellis changes with symbol probabilities). The proposed soft-input arithmetic decoder is very simple and can easily be extended to the adaptive context-based ACs.

4 Iterative decoding method for the serially concatenated ACs

In the previous section, we showed that the proposed soft-input arithmetic decoder takes profit from the bits' reliability at the output of the channel, which resulted in reducing the PER with respect to classical arithmetic decoding. In this section, it is modified in order to offer additional information concerning the compressed bit-stream components' reliability. Then, it is applied in an iterative decoding scheme composed of an AC and an RSCC. The system performance are evaluated in terms of PER.

4.1 Concatenated AC and RSCC transmission system description

As shown in Figure 4, in the considered system the source generates sequences $\mathbf{s}^h = (s_1^h, \ldots, s_L^h)$ of L symbols. An arithmetic encoder encodes the source symbol sequences. The bit-streams obtained $\mathbf{b}^h = (b_1^h, \ldots, b_{l_h}^h)$ are assembled to form the binary sequence $\mathbf{b} = (b_1,..., b_K)$, which is, then, scrambled by a

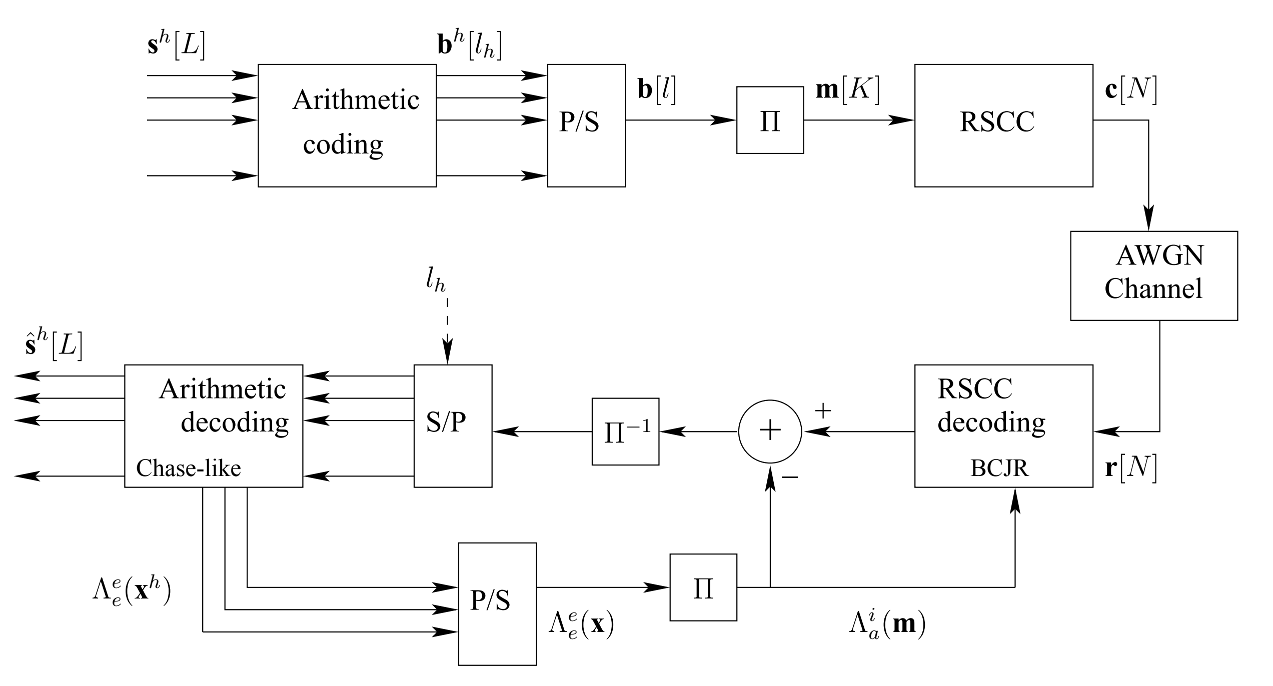


Figure 4 Block diagram of the studied concatenated AC and RSCC transmission scheme with iterative decoding.

random interleaver π. The interleaved sequence is encoded by a RSCC. The obtained sequence is transmitted over an AWGN channel by means of BPSK modulation.

At the receiver, we apply an iterative decoding based on information exchange between a low-complexity SISO Chase-like arithmetic decoder, detailed below, and the RSCC decoder using the optimal MAP algorithm. Performances of the iterative decoding scheme involving the Chase-like algorithm are evaluated and compared to tandem decoding results. As mentioned, major JSC iterative decoding contributions consider trellis-based algorithms for arithmetic SISO decoding. To evaluate the efficiency of our decoder with respect to such schemes, a comparison to an iterative decoding scheme with an arithmetic trellis-based de-coder is proposed. The reference decoder was presented in [17,18], and the authors used a bi-dimensional bit-clock trellis to model the arithmetic encoding machine. Then, to generate soft bit-reliability estimates, a modified SOVA [22] algorithm was proposed.

4.2 Low-complexity SISO arithmetic decoding

As mentioned in the previous section, the proposed soft-input arithmetic decoder does not use a finite-state machine to model the AC. Thus, BCJR [23] or SOVA [22] algorithms are not applicable. The main idea is not to generate bits *a posteriori* LLRs exact estimation but to define bits reliability factor.

The reliability of a component $\hat{b}_j$ of the decoded sequence $\hat{\mathbf{b}}$ is defined using the *a posteriori* LLR of the transmitted bit according to

$$\Lambda(\hat{b}_j) = \log\frac{P(b_j = 0|r_j)}{P(b_j = 1|r_j)} = \log\frac{P(r_j|b_j = 0)\cdot P(b_j = 0)}{P(r_j|b_j = 1)\cdot P(b_j = 1)}. \quad (9)$$

Often, contributions dealing with channel decoding assume an AWGN channel and equal *a priori* probabilities $P\ (b_j = 1) = P\ (b_j = 0)$, and use the expression given in Equation (8). However, in the case of iterative channel and arithmetic decoding, all the components of the decoded sequence do not have the same reliability and consequently the *a priori* term can be refined.

We have seen in the previous section that the proposed Chase-like arithmetic decoder calculates the J ($J \leq Q$) most likely length-valid compressed bit-streams $\hat{\mathbf{b}}^i$. The decoded sequence $\hat{\mathbf{b}}$ corresponds to the best sequence, among the J candidates, according to the metric given in Equation (7). It is clear that the positions j where we have the same bit for all possible candidate sequences $(\hat{b}_j^{i_1} = \hat{b}_j^{i_2}\ \forall i_1 \neq i_2)$ are the most reliable positions. In the following, we assume such bits as reliably decoded, and assign to them a constant extrinsic information $w_i = \beta\cdot(1 - 2.\hat{b}_j)$. We notify that a similar reliability definition was proposed in [24] for SISO decoding of linear block codes and that the value of β is determined experimentally. All the other bits are supposed non-reliable and we assign to them an extrinsic information equal to zero. It is worth noticing that in some cases, especially with a relatively noisy channel, the Chase-like arithmetic decoder is not able to find a length-valid codeword among the Q test sequences. In this case, we propose the decoding rule $\hat{\mathbf{b}} = \mathbf{y}$ and to assign to all its components an extrinsic information equal to 0.

4.3 Iterative arithmetic decoding performance

We consider a memoryless binary source with $p_0 = 0.2$ and $p_1 = 0.8$ ($H = 0.72$ bits per symbol) delivering packets of $L = 512$ symbols. Each packet is encoded by a static arithmetic encoder. The bit-streams delivered by the static arithmetic encoding of $P = 4$ packets are concatenated, then encoded by a 8-state RSCC with rate $R = \frac{1}{2}$. In Figure 5, the PER of the described scheme is reported versus signal-to-noise ratio $\frac{E_b}{N_0}$. We also depict the simulation results obtained with an earlier proposed iterative decoder based on a trellis representation of the AC which is SISO decoded using SOVA and List-SOVA algorithms. The trellis decoder may use the forbidden symbol technique with probability ε to insert a controlled redundancy in the compressed bit-stream that improves the arithmetic decoding performance. The curves in Figure 5 correspond to the results obtained at the fifth iteration. The tandem scheme involves the soft-input Viterbi algorithm for RSC decoding and the classical hard-input arithmetic decoder.

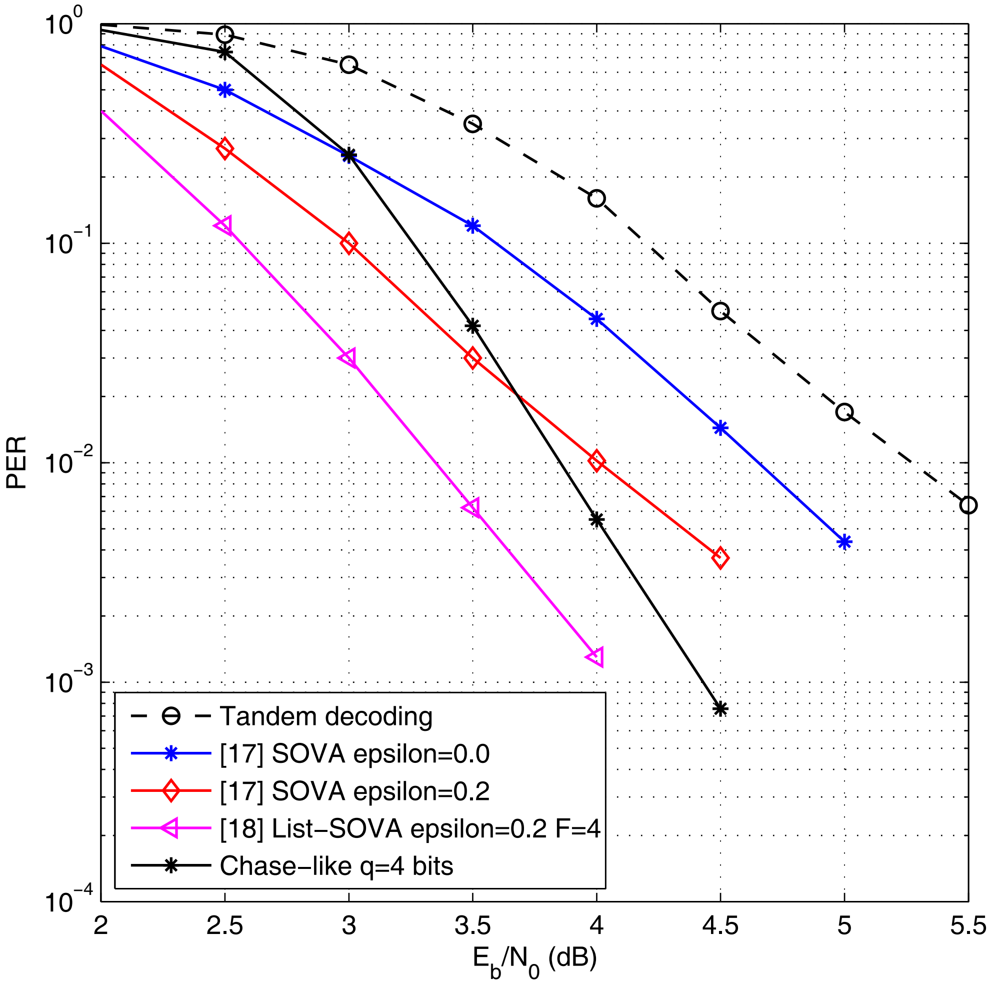


Figure 5 Performance in terms of PER as a function of $\frac{E_b}{N_0}$ for the tandem scheme and the iterative scheme using the proposed SISO arithmetic decoder, and the trellis-based SISO arithmetic decoding algorithms [17].

Simulation results show that significant gains are obtained with respect to the tandem scheme, for the different considered decoding algorithms. However, the gain varies with the channel signal-to-noise ratio $\frac{E_b}{N_0}$. For low $\frac{E_b}{N_0}$ values, the Chase-like arithmetic decoder is less efficient than the trellis-based decoder with $\varepsilon = 0.2$. However, for medium to high $\frac{E_b}{N_0}$. values, the Chase-like algorithm performance are better than the best configuration of trellis-based decoding using the modified SOVA algorithm. At a PER = 10^{-3}, an improvement of 0.3 dB in favor of the List-SOVA algorithm is obtained compared to Chase-like decoding.

Note that the main advantage of Chase-like decoding is that its complexity remains constant for increasing source sequence length L, and alphabet cardinal U. In fact, as mentioned in the previous section, the decoding complexity is determined by the number of hard decoding operations given by 2^q. In the case of the evaluated scheme we have $q = 4$ bits, and 16 classical arithmetic decoding operations are required, and which results in a reasonable complexity. On the other hand, the Chase-like algorithm is very simple and can be easily implemented for different types of ACs such as the contextual adaptive AC. However, the trellis-based arithmetic decoder is very hard to implement with such type of ACs.

In the next section, we will valid the Chase-like arithmetic decoding advantages in the context of an image transmission system using the JPEG 2000 standard which considers a contextual adaptive AC.

5 Application to JPEG 2000 coded images transmitted over a noisy channel

The objective, in this section, is to improve the performance of an image transmission system using JPEG 2000 standard for compression and a Convolutional Code as a channel code.

The considered coding scheme uses the JPEG 2000 encoder which compresses the source image at D_s bits per pixel (bpp). In JPEG 2000 coding, the 9-7 filters are used for wavelet transform and the number of resolution levels is equal to five. The wavelet domain is divided into rectangular regions called code-blocks. The JPEG 2000 encoding machine defines multiple quality layers that help image reconstruction at different rates (scalability). In the following experiment, we consider a single quality layer for simplicity and the scheme can be generalized to multiple quality layers. The values of D_s = 0.4 bpp and D_s = 1 bpp are considered. The bit-stream generated by the JPEG 2000 encoder is composed of headers describing the coding parameters followed by a sequence of packets containing the encoded data. We assume that the data contained in the headers are transmitted without error. The JPEG 2000 uses a context-based binary adaptive AC called the MQ-coder. The code-blocks are independently encoded by the AC [3].

The experimental setup is as follows. The test image Lena 512 × 512 initially coded at 8 bpp is considered. By analogy to the system proposed in the previous section, each bit-stream resulting from the compression of $P = 4$ code-blocks form the message **b**. The latter is scrambled then, coded by an 8-state RSCC with rate $R = \frac{1}{2}$. Finally, the coded image is transmitted over an AWGN channel by means of the BPSK modulation. In our simulation, we used an open source implementation of JPEG 2000 called OpenJPEG (J2000 library). More details about the implementation are available in http://www.openjpeg.org.

At the receiver side, we apply the iterative decoding, described in the previous section, based on information transfer between the JPEG 2000 decoder and the channel decoder. The JPEG 2000 decoder uses the proposed SISO arithmetic decoder described in Section 4.2. The test pattern weight q is fixed to 4 bits. The value of the extrinsic information β given by the SISO arithmetic decoder is experimentally optimized and we used, for $\frac{E_b}{N_0} = 4$ dB a value of $\beta = 2.5$, and for $\frac{E_b}{N_0} \geq 4.5$ dB a $\beta = 4.0$. The BCJR [23] algorithm is applied for RSCC decoding with soft-inputs and outputs.

Performance are measured in terms of average PSNR over 500 independent image transmissions. As a reference, we consider a system with the same parameters and classical JPEG 2000 decoding. The simulation results are reported in Figures 6 and 7 for the respective bit rates $D_s = 0.4$ and $D_s = 1$ bpp at the output of JPEG 2000 encoder.

It can be seen that remarkable gains are obtained in terms of average PSNR. In fact, for $D^s = 0.4$ bpp and at $\frac{E_b}{N_0} = 4.5$ dB the proposed algorithm exhibits a PSNR gain of 5 dB. The iterative decoding yields a significant gain of 6.2 dB over the tandem scheme. On the other hand, for the same bit rate (D_s = 0.4 bpp), the studied system reaches approximately an average PSNR value of 35 dB at $\frac{E_b}{N_0} = 5.0dB$. Whereas, the tandem scheme achieves this value at $\frac{E_b}{N_0} = 6.5$ dB. Therefore, the iterative decoding allows a gain of 1.5 dB in terms of $\frac{E_b}{N_0}$.

For $D_s = 1$ bpp and at $\frac{E_b}{N_0} = 4.5$ dB, an average PSNR gain of 3.8 dB is obtained at the first iteration with respect to the tandem scheme. This gain increases with iterations to reach the value of 8.5 dB over the tandem scheme. Moreover, from $\frac{E_b}{N_0} = 5.0$ dB the value of the average PSNR obtained at the third iteration is

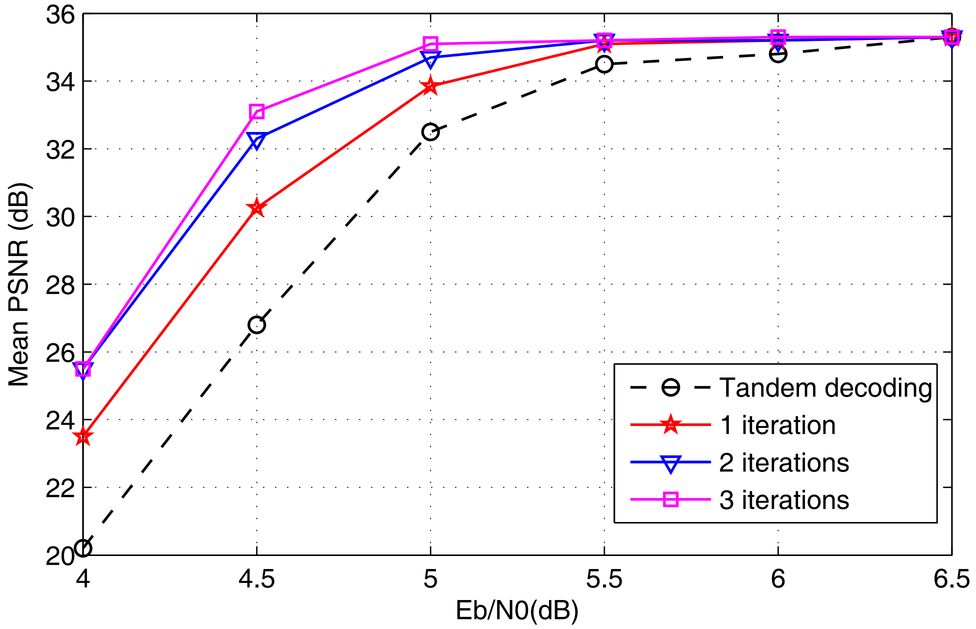


Figure 6 **The average PSNR versus $\frac{E_b}{N_0}$ for the classical decoding and iterative decoding applied to JPEG 2000 standard, rate = 0.4 bpp.**

approximately 39 dB. However, this value is reached by the tandem scheme at $\frac{E_b}{N_0}$ = 6.5 dB. Therefore, the iterative decoding enables a gain of 1.5 dB in terms of $\frac{E_b}{N_0}$.

In Figure 8, we give examples of reconstructed images in order to illustrate the improvement obtained by the proposed iterative decoder in terms of visual quality. The figure shows improvement in image quality over iterations, which results in increasing the PSNR value.

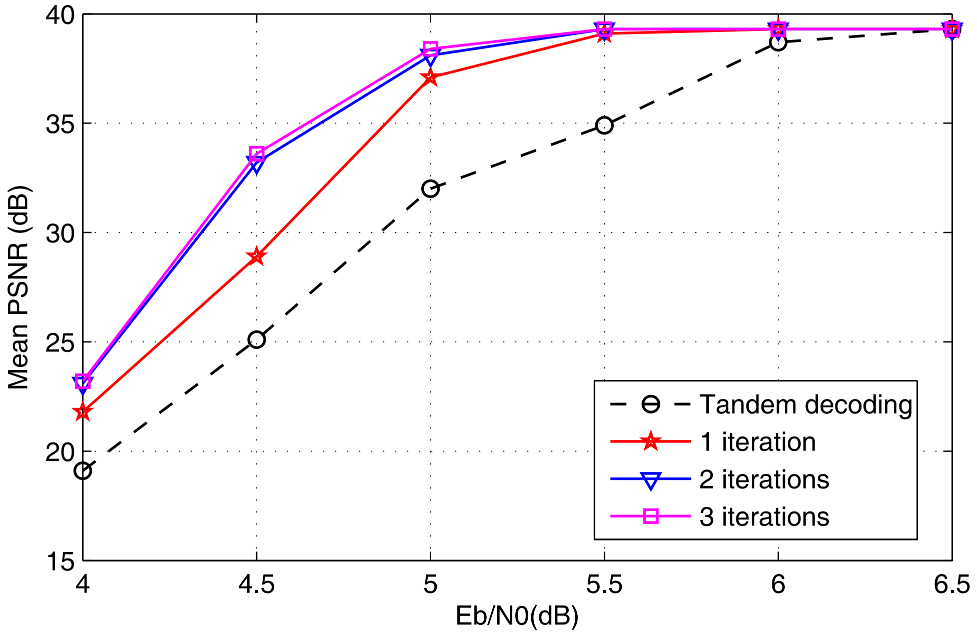
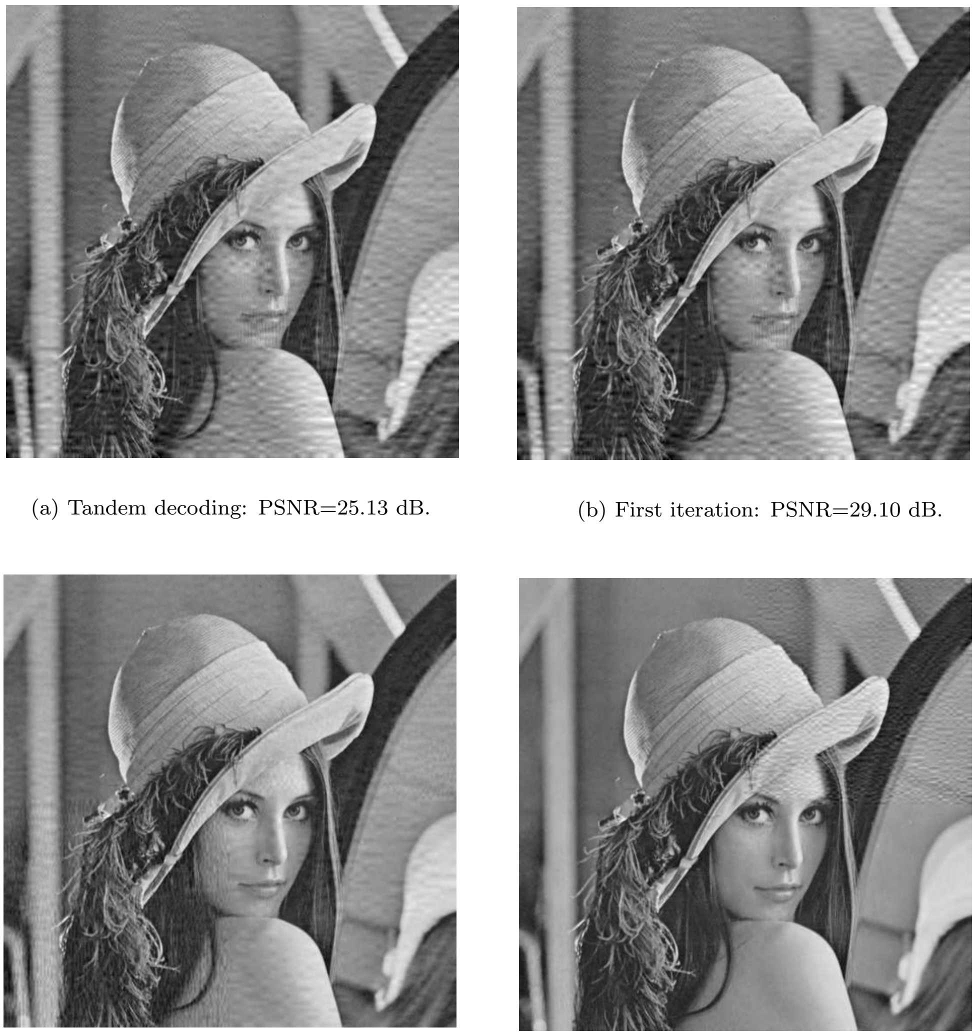


Figure 7 **The average PSNR versus $\frac{E_b}{N_0}$ for the classical decoding and iterative decoding applied to JPEG 2000 standard, rate = 1 bpp.**



(a) Tandem decoding: PSNR=25.13 dB.

(b) First iteration: PSNR=29.10 dB.

(c) Second iteration: PSNR=32.54 dB.

(d) Third iteration: PSNR=34.29 dB.

Figure 8 Examples of reconstructed images in the case of $\frac{E_b}{N_0}$ = 4.5 dB and D_s = 1 bpp. (b)-(d) Correspond to iterative decoding with the Chase-like algorithm.

6 Conclusion

In this article, we have proposed novel low-complexity decoding algorithms for ACs based on the Chase algorithm. The schemes were tested for transmission across an AWGN channel with BPSK signaling and shows numerous significant advantages. First, we showed that the soft-input arithmetic decoder achieves good error correction performance, has low complexity and can be easily extended to adaptive ACs, unlike the trellis-based arithmetic decoders. Second, we showed that the Chase-like algorithm can be slightly modified to generate additional information regarding the reliability of the decoded bits. Such improvement allow iterative decoding in the case of the serial concatenation of an AC with a channel code. As a second experiment, the concatenation of an AC with a RSCC was considered and iterative decoding results were investigated. The scheme is a JSC decoding approach where AC embeds compression and error correction in a single stage. Simulation results shows significant performance improvement when compared to tandem decoding and to our previous iterative decoding scheme using a trellis-based SISO arithmetic decoder. Moreover, the presented iterative system has profitably been exploited in the case of JPEG 2000 image transmission. In fact, improvements in terms of average PSNR, and visual quality were observed compared to standard JPEG 2000 decoding.

Author details

[1]Syscom Laboratory, National Engineering School of Tunis, BP 37, Le belvedere, 1002, Tunis, Tunisia [2]Institut Telecom/Telecom Bretagne, Lab-STICC UMR CNRS 3192, Universite Europeenne de Bretagne, France

Competing interests

The authors declare that they have no competing interests.

Received: 2 February 2012 Accepted: 24 May 2012
Published: 24 May 2012

References

1. JJ Rissanen, GG Langdon, Arithmetic coding. IBM J Res Dev. **23**(2), 149–162 (1979)
2. IH Witten, RM Neal, JG Cleary, Arithmetic coding for data compression. Comm ACM. **30**(6), 520–540 (1987). doi:10.1145/214762.214771
3. DS Taubman, MW Marcellin, *JPEG 2000: image compression fundamentals, standards and practice*, (Kluwer Academic Publishers, 2002)
4. IEG Richardson, *H.264 and MPEG-4 video compression: video coding for next generation multimedia*, (John Wiley And Sons Ltd, 2003)
5. DW Redmill, NG Kingsbury, The EREC: an error-resilient technique for coding variable-length blocks of data. IEEE Trans Image Processing. **5**(4), 565–574 (1996). doi:10.1109/83.491333
6. C Boyd, JG Cleary, SA Irvine, I Rinsma-Melchert, IH Witten, Integrating error detection into arithmetic coding. IEEE trans Communications. **145**, 1–3 (1997)
7. J Chou, K Ramchandran, Arithmetic coding-based continuous error detection for efficient ARQ-based image transmission. IEEE J Sel Areas Commun. **18**(6), 861–867 (2000). doi:10.1109/49.848240
8. R Anand, K Ramchandran, IV Kozintsev, Continuous error detection (CED) for reliable communication. IEEE Trans Communications. **49**(9), 1540–1549 (2001). doi:10.1109/26.950341
9. G Elmasry, Y Shi, MAP symbol decoding of arithmetic coding with embedded channel coding. in *Proceedings of the Wireless Communications and Networking Conf.*: New Orleans, USA. **2**, 988–992 (1999)
10. BD Pettijohn, W Hoffman, K Sayood, Joint source/channel coding using arithmetic codes. IEEE Trans Communications. **49**, 826–836 (2001). doi:10.1109/26.923806
11. M Grangetto, P Cosman, G Olmo, Joint source/channel coding and MAP decoding of arithmetic coding. IEEE Trans Communications. **35**, 1007–1016 (2005)
12. J Sayir, Arithmetic coding for noisy channels. in *Proceedings of the IEEE Information Theory Workshop:* Kruger National Park, South Africa 69–71 (1999)
13. T Guionnet, C Guillemot, Soft decoding and synchronization of arithmetic codes: Application to image transmission over noisy channels. IEEE Trans on Image Processing. **12**, 1599–1609 (2003). doi:10.1109/TIP.2003.819307
14. B Dongsheng, W Hoffman, K Sayood, State machine interpretation of arithmetic codes for joint source and channel coding. in *Proceedings of the IEEE Data Compression Conference:* Snowbird, Utah, USA 143–152 (2006)
15. S Ben-Jamaa, C Weidmann, M Kieffer, Analytical tools for optimizing the error correction performance of arithmetic codes. IEEE Trans Communications. **56**(9), 1458–1468 (2008)
16. M Grangetto, B Scanavino, G Olmo, S Bendetto, Iterative decoding of serially concatenated arithmetic and channel codes with JPEG 2000 applications. IEEE Transactions on Image Processing. **16**(6), 1557–1567 (2007)
17. A Zribi, S Zaibi, R Pyndiah, A Bouallegue, Low-complexity joint source/channel turbo decoding of arithmetic codes with image transmission application. in *Proceedings of the IEEE Data Compression Conference*: Snowbird, Utah, USA 472 (2009)
18. A Zribi, S Zaibi, R Pyndiah, A Bouallegue, Low-complexity joint source/channel turbo decoding of arithmetic codes. in *Proceedings of the IEEE intl symp on turbo codes and related topics:* Lausanne, Switzerland 385–389 (2009)
19. X Pan, A Cuhadar, AH Banihashemi, Combined source and channel coding with JPEG2000 and rate-compatible low-density Parity-check codes. IEEE Transactions on Signal processing. **54**(3), 1160–1164 (2006)
20. L Pu, Z Wu, A Bilgin, MW Marcellin, B Vasic, LDPC-based iterative joint source-channel decoding for JPEG2000. IEEE Trans Image Process. **16**(2), 577–581 (2007)
21. D Chase, A class of algorithms for decoding block codes with channel measurement information. IEEE Transactions on Information Theory. **18**, 170–182 (1972). doi:10.1109/TIT.1972.1054746
22. J Hagenauer, P Hoeher, A Viterbi algorithm with soft-decision outputs and its applications. in *Proceedings of the IEEE Globecom:* Dallas, TX, USA 11–17 (1989)
23. L Bahl, J Cocke, F Jelinek, J Raviv, Optimal decoding of linear codes for minimizing symbol error rate. IEEE Transactions on Information Theory. **20**(2), 284–287 (1974)
24. R Pyndiah, Near-optimum decoding of product codes: block turbo codes. IEEE Transactions on Communications. **46**(8), 1003–1010 (1998). doi:10.1109/26.705396

doi:10.1186/1687-6180-2012-114
Cite this article as: Zaibi *et al.*: **Joint source/channel iterative arithmetic decoding with JPEG 2000 image transmission application.** *EURASIP Journal on Advances in Signal Processing* 2012 **2012**:114.