Hindawi Publishing Corporation
EURASIP Journal on Image and Video Processing
Volume 2007, Article ID 69805, 9 pages
doi:10.1155/2007/69805

Research Article

Progressive Image Transmission Based on Joint Source-Channel Decoding Using Adaptive Sum-Product Algorithm

Weiliang Liu[1, 2] and David G. Daut[1]

[1] *Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey, Piscataway, NJ 08854, USA*
[2] *Qualcomm Inc., San Diego, CA 92121, USA*

Received 13 August 2006; Revised 12 December 2006; Accepted 5 January 2007

Recommended by Béatrice Pesquet-Popescu

A joint source-channel decoding method is designed to accelerate the iterative log-domain sum-product decoding procedure of LDPC codes as well as to improve the reconstructed image quality. Error resilience modes are used in the JPEG2000 source codec making it possible to provide useful source decoded information to the channel decoder. After each iteration, a tentative decoding is made and the channel decoded bits are then sent to the JPEG2000 decoder. The positions of bits belonging to error-free coding passes are then fed back to the channel decoder. The log-likelihood ratios (LLRs) of these bits are then modified by a weighting factor for the next iteration. By observing the statistics of the decoding procedure, the weighting factor is designed as a function of the channel condition. Results show that the proposed joint decoding methods can greatly reduce the number of iterations, and thereby reduce the decoding delay considerably. At the same time, this method always outperforms the nonsource controlled decoding method by up to 3 dB in terms of PSNR.

1. INTRODUCTION

Progressive coded images, such as those compressed by wavelet-based compression methods, have wide application including image communications via band-limited wireless channels. Due to the embedded structures of the corresponding compressed codestreams, transmission of such images over noisy channels exhibits severe error sensitivity and always experiences error propagation. Forward error correction (FEC) is a typical method used to ensure reliable transmission. Powerful capacity-achieving channel codes such as turbo codes and low-density parity-check (LDPC) codes have been used to protect the JPEG2000 codestream using various methods [1–3]. The typical idea of these schemes is to assign different channel protection levels via joint source-channel coding (JSCC) based on a rate distortion method. In addition to JSCC systems that are designed at the transmitter/encoder side, researchers also find that joint source-channel decoding (JSCD) can be achieved at the receiver/decoder side. The concept of utilizing source decoded information to aid the channel decoding procedure, and hence, improve the overall performance of the receiver can be traced back to early work by Hagenauer [4]. He proposed a modification to the Viterbi decoding algorithm that used additional *a priori* or *a posteriori* information about the source bit probability. A generalized framework which is suitable to any binary channel code was introduced in [5]. The iterative decoding procedure of turbo codes, implemented by exchanging the extrinsic information from one constituent decoder to another, makes it quite natural to use the information that comes from the source decoder as an additional extrinsic message, and thereby generate better soft-output data during each iteration. The iterative decoding behavior of the turbo codes can be found in [6, 7]. The JSCD methods using turbo codes have been studied in [8–11]. Image transmission based on turbo codes using a JSCD method was studied in [12] where vector quantization, JPEG, and MPEG coded images were tested and a wide range of improvements in turbo decoding computational efficiency was shown. After the rediscovery of the low-density parity-check

(LDPC) codes [13, 14], they had been quickly adopted for many applications including image transmission. LDPC iterative decoding behavior has been studied in [15–17]. In [18], a JSCD method for JPEG2000 images has been proposed using a modification algorithm similar to that in [12].

In this paper, we develop a JSCD method for JPEG2000 image transmission on both AWGN and flat Rayleigh fading channels. Fading channels wherein the receiver either has, or does not have additional channel state information (CSI) are considered. A regular LDPC code is used as the error correcting code. Log-domain iterative sum-product algorithm is chosen as the channel decoding method. After each iteration of the log-domain sum-product algorithm, the source decoder provides useful information as feedback that is based on the error resilience modes employed in the source codec. The information is then used to modify the log-likelihood ratio (LLR) of the corresponding bit nodes. The new modification factor presented in this paper extends the idea previously investigated in [12, 18]. Results show that the new scheme can accelerate the iterative sum-product decoding process as well as improving the overall reconstructed image quality.

The outline of this paper is as follows. Section 2 presents the sum-product algorithm and some observations about its iterative behavior. JPEG2000 and its error resilience capability are first described in Section 3 followed by the design of the joint source-channel decoding algorithm. Section 4 presents selected simulation results. Conclusions are given in Section 5.

2. SUM-PRODUCT ALGORITHM AND ITS ITERATIVE BEHAVIOR

The iterative sum-product algorithm for LDPC decoding in the log-domain is first introduced in this section. Both the AWGN channel and the flat Rayleigh fading channels with and without CSI are considered. The corresponding behaviors of the iterative algorithm are described in the second part of this section.

2.1. Log-domain sum-product algorithm

Consider an $M \times N$ sparse parity check matrix H, where $M = N - K$. N is the length of a codeword, and K is the length of the source information block. An example of H is shown as

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{bmatrix}. \tag{1}$$

The sparse matrix H has an equivalent bipartite graph description called a Tanner graph [19]. Figure 1 shows the Tanner graph corresponding to (1). In the graph, each column (row) of H corresponds to a bit node (a check node). Edges connecting check and bit nodes correspond to ones in H. In this example, each bit node is connected by 3 edges and each check node is connected by 6 edges. Therefore, each column of H corresponds to a bit node with weight 3 and each row of

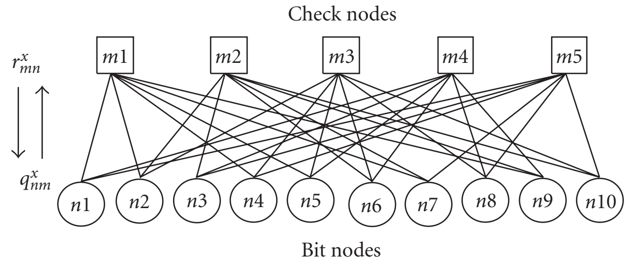


FIGURE 1: An example of Tanner graph corresponding to the matrix H in (1).

H corresponds to a check node with weight 6. A detailed iterative sum-product decoding algorithm is presented in [20]. In order to reduce the computation complexity and the numerical instability, a log-domain algorithm is preferred. It is introduced briefly as follows.

The message $r^{x,l}_{mn}$, the probability that bit node n has the value x given the information obtained via all the check nodes connected to it other than check node m for the lth iteration, is passed from check nodes to bit nodes. Similarly, the dual message $q^{x,l}_{nm}$ is passed from bit nodes to check nodes. Here, x is either 1 or 0. We define a set of bits n that participate in check m as $\mathcal{N}(m) = \{n : H_{mn} = 1\}$ and define a set of checks m in which bit n participates as $\mathcal{M}(n) = \{m : H_{mn} = 1\}$. Notation $\mathcal{N}(m) \setminus n$ denotes a set $\mathcal{N}(m)$ with bit n excluded and notation $\mathcal{M}(n) \setminus m$ denotes a set $\mathcal{M}(n)$ with check m excluded. The algorithm produces the LLR of the *a posteriori* probabilities for all the codeword bits after a certain number of iterations.

Consider an AWGN channel with BPSK modulation that maps the source bit c to the transmitted symbol x according to $x = 1 - 2c$. The received signal is modeled as $y = x + n_w$ with the conditional distribution

$$p(y \mid x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-x)^2}{2\sigma^2}\right), \tag{2}$$

where n_w is white Gaussian noise with variance $\sigma^2 = 1/2 \cdot R \cdot (E_b/N_0)$, and R is the channel code rate. At the initial step, bit nodes n have the values given by

$$Lc_n = Lq^0_{nm} = \log\left(\frac{P(c_n = 0 \mid y_n)}{P(c_n = 1 \mid y_n)}\right) = \frac{2}{\sigma^2} \cdot y_n. \tag{3}$$

Denote the corresponding LLR of the messages $q^{x,l}_{nm}$ and $r^{x,l}_{mn}$ as $Lq^l_{nm} = \log(q^{0,l}_{nm}/q^{1,l}_{nm})$ and $Lr^l_{mn} = \log(r^{0,l}_{mn}/r^{1,l}_{mn})$, respectively. Before the first iteration, Lq^0_{nm} is set to Lc_n. By denoting $Lq^l_{nm} = \alpha^l_{nm} \cdot \beta^l_{nm}$, where $\alpha^l_{nm} = \text{sign}(Lq^l_{nm})$ and $\beta^l_{nm} = \text{abs}(Lq^l_{nm})$, the first and the second parts of one iteration are

$$Lr^l_{mn} = \left(\prod_{n' \in \mathcal{N}(m)\setminus n} \alpha^l_{n'm}\right) \cdot \Phi\left(\sum_{n' \in \mathcal{N}(m)\setminus n} \Phi(\beta^l_{n'm})\right), \tag{4}$$

$$Lq^l_{nm} = Lc_n + \sum_{m' \in \mathcal{M}(n)\setminus m} Lr^l_{m'n}, \tag{5}$$

where $\Phi(x) = -\log(\tanh(x/2)) = \log((e^x+1)/(e^x-1))$. The LLR of "pseudoposteriori probability" defined as $LQ_n^l = \log(Q_n^{0,l}/Q_n^{1,l})$ is then computed as

$$LQ_n^l = Lc_n + \sum_{m \in \mathcal{M}(n)} Lr_{mn}^l. \tag{6}$$

The following tentative decoding is made: $\hat{c}_n^l = 0$ (or 1) if $LQ_n^l > 0$ (or < 0). When $LQ_n^l = 0$, $\hat{c}_n^l$ is set to 0 or 1 with equal probability. In theory, when $H\hat{c}_n^l = 0$, the iterative procedure stops.

2.2. *Decoding in the case of fading channels*

For wireless communication, Rayleigh fading channel is typically a good channel model. Consider an uncorrelated flat Rayleigh fading channel. Assume that the receiver can estimate the phase with sufficient accuracy, then coherent detection is feasible. The received signal is now modeled as $y = ax + n_w$, where n_w is white Gaussian noise as described in the previous subsection. The parameter a is a normalized Rayleigh random variable with distribution $P_A(a) = 2a \cdot \exp(-a^2)$ and $E[a^2] = 1$. Assume that the fading coefficients are uncorrelated for different symbols. BPSK modulation maps the source bit c to the transmitted symbol x according to $x = 1 - 2c$. At the initial step, the bit nodes n take on the values

$$Lc_n = \log\left(\frac{P(c_n = 0 \mid y_n)}{P(c_n = 1 \mid y_n)}\right) = \frac{2a}{\sigma^2} \cdot y_n. \tag{7}$$

The message definition above implies that the receiver has perfect knowledge of the CSI. For the case when CSI is not available at the receiver, $E[a] = 0.8862$ can be used instead of the instantaneous value a in (3). Thus, the bit nodes n take on the values

$$Lc_n = \log\left(\frac{P(c_n = 0 \mid y_n)}{P(c_n = 1 \mid y_n)}\right) = \frac{2E[a]}{\sigma^2} \cdot y_n. \tag{8}$$

For each iteration thereafter, the relationships given in (4)–(6) are used once again without any changes.

2.3. *Behavior of the sum-product algorithm*

As mentioned above, once $H\hat{c}_n = 0$, the iterative procedure stops. However, a large number of iterations may be needed to meet this criteria. Also, there is no guarantee that the iterative procedure converges unless the codeword length is infinite. In real-world applications, there exist three implementation problems: (1) finite block lengths (e.g., 10^3–10^4) are used; (2) the sum-product algorithm is optimal in the sense of minimizing the bit error probability for a cycle-free Tanner graph. For finite length codes, the influence of cycles cannot be neglected; and (3) the maximum number of iterations is always preselected before communication takes place. The preselected iteration number is usually smaller (e.g., 40–60) compared to the number that is needed to satisfy the strict stopping criteria. Examples are presented in the following to illustrate the iterative behavior of LDPC codes. A regular (4096,3072) LDPC code with rate 3/4 is selected. The

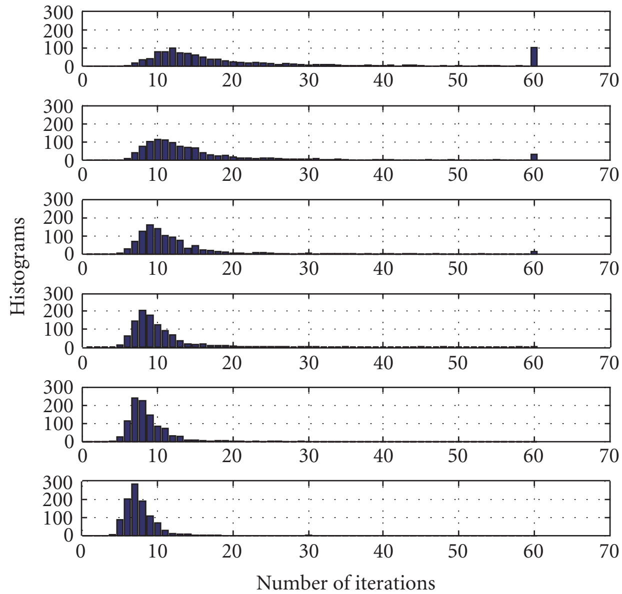


FIGURE 2: Histogram for number of iterations for the log-domain sum-product algorithm over AWGN channel. ($\gamma = 2.50$ to 3 dB in increments of 0.1 dB, from top to bottom.)

log-domain decoding procedure is performed for a total of 1000 transmission trials. The maximum number of channel decoder iterations is set to 60 for each trial. Two channels are tested, one is the AWGN channel and the other is the flat fading channel with CSI. Figures 2 and 3 show the histograms of the iteration numbers versus $\gamma = E_b/N_0$ and the average E_b/N_0, $\overline{\gamma}$. The x-axis represents the number of iterations needed for each LDPC decoding trial. The y-axis represents the number of occurrences (out of 1000 experiments) of a certain number of iterations. The figures illustrate that with increasing γ and $\overline{\gamma}$, the overall histogram becomes more and more narrow. This means that the decoding time reduces when better channel conditions are realized. Another point of observation obtained from these figures is that of the bars located at the maximum number of iterations, 60. For an AWGN channel, operating at $\gamma = 2.5$ dB, there are about 100 out of 1000 times that the decoding procedure does not satisfy $H\hat{c}_n = 0$, and has to abruptly stop. With increased channel SNR, this number becomes 31, 12, and 2 at $\gamma = 2.6$, 2.7, and 2.8 dB, respectively. The number of times the maximum is needed becomes zero as the channel condition continues to improve. Similar observations are also found for the fading channel. In Figure 3, operating at $\overline{\gamma} = 6.55$ dB, there are about 36 decoding procedures that do not satisfy $H\hat{c}_n = 0$ and have to stop. This number becomes 6 at $\overline{\gamma} = 6.75$ dB. Reducing the number of decoding failures indicates that the performance of the code becomes increasingly better.

In addition to the histogram of iteration numbers, Figures 4 and 5 present two meaningful statistics, the mean and the median of the number of iterations, for the AWGN channel and the fading channels with and without CSI. It is shown that the mean number of iterations is a monotonically decreasing function of the channel conditions. The discrete

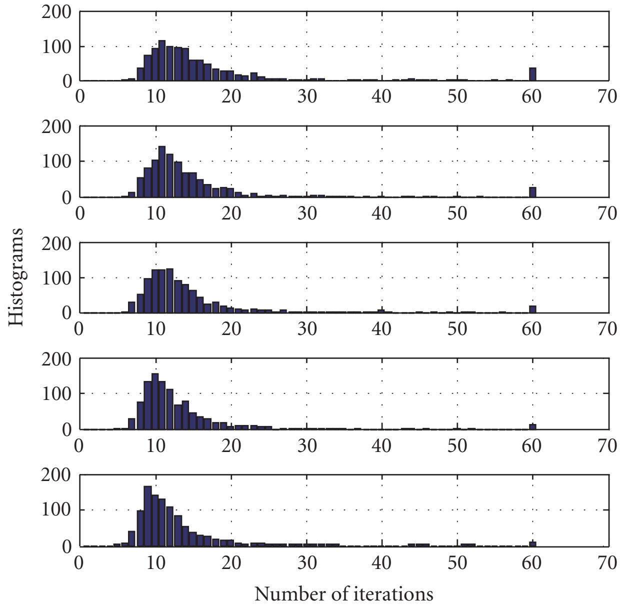


Figure 3: Histogram for number of iterations for the log-domain sum-product algorithm over flat fading channel with CSI. ($\overline{\gamma}$ = 6.55 to 6.75 dB in increments of 0.05 dB, from top to bottom.)

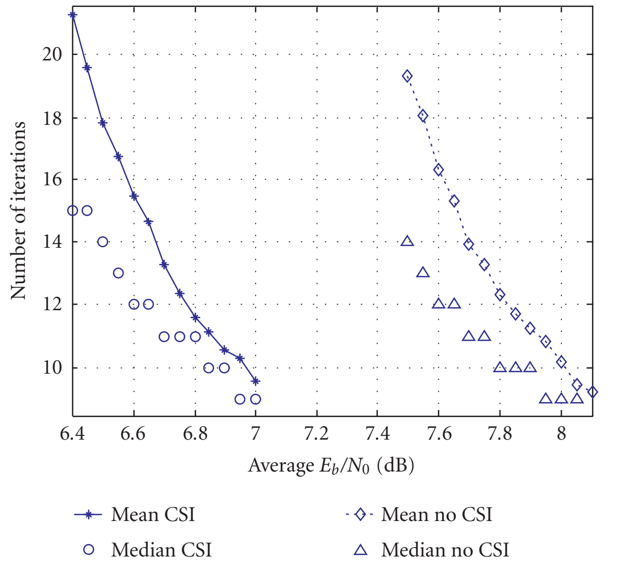


Figure 5: Mean and median number of iterations over flat Rayleigh fading channel with and without CSI.

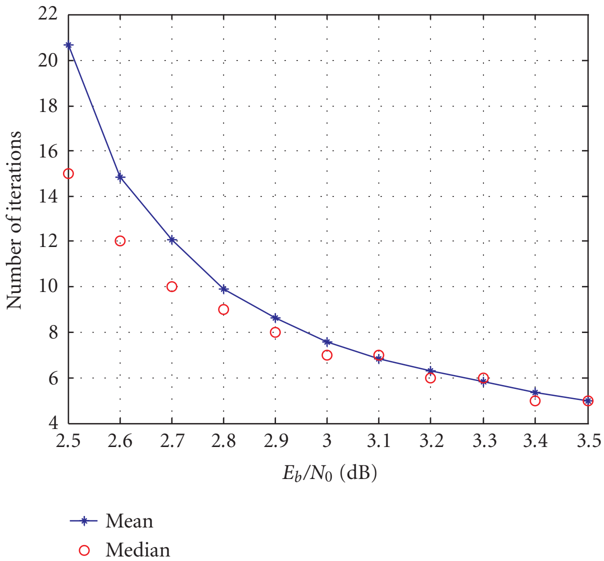


Figure 4: Mean and median number of iterations over AWGN channel.

values of the median have a property similar to a nonincreasing function. The mean and median are two important statistics that better measure the number of iterations needed during the decoding process.

The decoder iteration behaviors described above provide some insight for practical design considerations. It is desired to establish a JSCD methodology that has the capability to update the messages that are passing back and forth between the bit and check nodes during the iterations. Furthermore, such updated information should come from outside of the LDPC decoder as extrinsic information similar to that which is exchanged between the constituent convolutional decoders within an iterative turbo decoder.

3. JOINT SOURCE-CHANNEL DECODER DESIGN

A natural choice for the provider of the extrinsic information is the source decoder that follows the channel decoder. In this paper, the JPEG2000 decoder after the LDPC decoder can provide such extrinsic information. The error resilience tools provided in the JPEG2000 standard are discussed in the first part of this section. In the second part, the details of the JSCD design are provided.

3.1. *Error resilience methods in JPEG2000*

In the JPEG2000 standard, several error resilience methods are defined to deal with the error sensitivity and error propagation resulting from its embedded codestream structure. Among them, a combined use of "RESTART" and "ERTERM" tools provides a mechanism such that if there exists at least one bit error in any given coding pass, the remaining coding passes in the same codeblock will be discarded since the rest of bits in this codeblock have strong dependency on the error bit. The mechanism is illustrated in Figure 6. In this example, a codeblock in the LH subband of the second resolution (corresponding to the second packet in each quality layer) has 15 coding passes. They are distributed into 3 quality layers. After transmission, assume that a bit error occurred at the 10th coding pass. Thus, the JPEG2000 decoder will only use the first 9 error-free coding passes of this codeblock for reconstruction. Since the last 6 coding passes are discarded, errors are thereby limited to only one codeblock

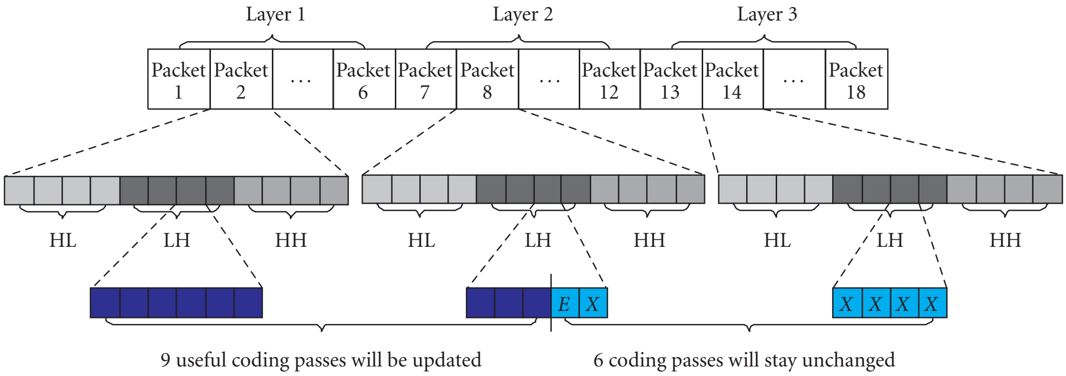


FIGURE 6: Error resilience mechanism used in JPEG2000 to prevent error propagation.

and will not be propagated to other codeblocks in the transmitted data stream.

3.2. *Adaptive modification in the joint design*

From the channel decoder point of view, the error resilience mechanism implemented in the source decoder may provide potential feedback information that makes it possible to design a joint source-channel decoder. In [12, 18], two different modification methods have been proposed. The former one either enlarges or reduces the extrinsic information in turbo codes by the mappings $x' = x \cdot t$ or $x' = x/t$, respectively, where t is the modification factor whose value depends on the channel conditions. The latter one uses a simple plus or minus operation to modify the LLR values in LDPC codes as $x' = x + t$ or $x' = x - t$. We note here, most importantly, that t is channel-independent. As discussed in Section 2, the behavior of the iterative decoding algorithm is channel-dependent. Hence, a channel-adaptive modification algorithm is expected to be more beneficial both in the reduction of computation time and the improvement in overall image quality. Since the log-domain is used in the sum-product algorithm, using plus and minus operations to increase and decrease the LLR values coincides with the product and division algorithms in the probability domain.

The proposed joint decoder block diagram is illustrated in Figure 7. It operates as follows: the parts in the dashed line frame represent a typical log-domain iterative sum-product LDPC decoder. After the ith iteration, the JPEG2000 decoder receives the tentative decoded bits $\hat{c}_n^i$. Only several initial JPEG2000 decoding steps will be executed. The aim is to find which coding pass contains the first bit error within a codeblock. The whole JPEG2000 decoding procedure will not be applied at this time. Compared to an iteration of LDPC decoding, such an operation is very quick. In the example shown in Figure 6, the 10th coding pass contained the first bit error. The JPEG2000 decoder then feeds the positions of bits P^i, which belong to the useful coding passes (the first 9 useful coding passes in Figure 6), back to the channel decoder. The Lc_n values corresponding to those positions will be updated and denoted as $Lc_n^{i,\text{new}}$. At the same time, the LLR values of the last 6 coding passes will remain unchanged. The adaptive

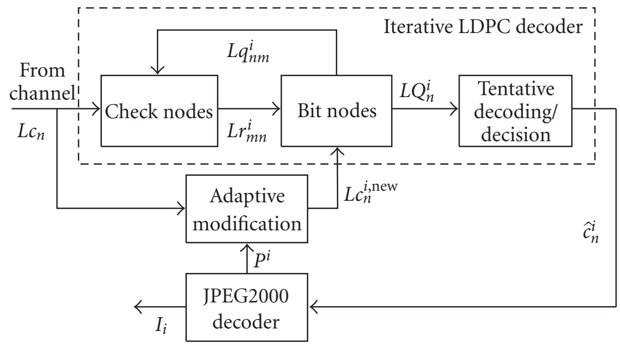


FIGURE 7: Block diagram of the joint source-channel decoder.

modification methods will be discussed later. At the initial step, Lc_n is calculated using (3), (7), or (8), and after that, for each iteration, it will be updated as $Lc_n^{i,\text{new}}$ and sent to the bit nodes. Bit nodes then use $Lc_n^{i,\text{new}}$ to compute the second part of the iteration corresponding to (5) and the tentative decision. When the iterative procedure stops, the JPEG2000 decoder reconstructs the entire image I_i as the system output. Thus, the modification factor $t(\cdot)$ used in the algorithm is designed so as to be a function of the channel condition. Hence, the desired parameter is $t(\gamma)$, with γ being the channel SNR in terms of E_b/N_0. A similar approach can be used in connection with flat fading channels. Using the average channel SNR, $\overline{\gamma}$, $t(\cdot)$ is designed to be $t(\overline{\gamma}, a)$ and $t(\overline{\gamma}, E[a])$ for fading channels with CSI and without CSI, respectively. Then, the modification algorithm after each iteration is defined as

$$Lc_n^{i,\text{new}} = \begin{cases} Lc_n^{i-1,\text{new}} + t(\cdot) & \text{if } \hat{c}_n^{i-1} = 0;\ n \in P^{i-1}, \\ Lc_n^{i-1,\text{new}} - t(\cdot) & \text{if } \hat{c}_n^{i-1} = 1;\ n \in P^{i-1}, \\ Lc_n^{i-1,\text{new}} & \text{if } n \notin P^{i-1}, \end{cases} \quad (9)$$

where

$$t(\cdot) = \begin{cases} t(\gamma) & \text{for AWGN Channel}, \\ t(\overline{\gamma}, a) & \text{for flat fading channel with CSI}, \\ t(\overline{\gamma}, E[a]) & \text{for flat fading channel without CSI}. \end{cases} \quad (10)$$

At the initial iteration, $Lc_n^{0,\text{new}} = Lc_n$. P^i is a set of bits that belongs to the correct coding passes for the ith iteration obtained from the JPEG2000 decoder. The $Lc_n^{i-1,\text{new}}$ values associated with the P^i bits are either plus or minus a modification factor $t(\cdot)$ so as to generate new LLR values. Bits that are not in the set P^i hold onto their last iteration values without any update. Further, since the fading coefficient a attenuates the transmitted symbol x, it is worthwhile to compensate for a in the case of those bits that belong to P^i. Thus, the modification factors can be written as $t(\overline{\gamma})/a$ and $t(\overline{\gamma})/E[a]$, respectively. Both $t(\gamma)$ and $t(\overline{\gamma})$ can be tabulated empirically before beginning real-time transmission of compressed image data.

4. SELECTED SIMULATION RESULTS

The proposed JSCD method and the associated modification algorithm have been simulated. The 8-bit gray-scale Lena image was used. Three source coding rates 1.0, 0.5, and 0.1 bpp were selected. For each rate, three quality layers were generated. A (4096, 3072) regular LDPC code with rate 3/4 was employed in the system. The maximum number of iterations was set to 60. For AWGN and flat fading channels (assume uncorrelated Rayleigh fading), different sets of γ or $\overline{\gamma}$ were selected so that the performances of the LDPC code are close to each other under these channel conditions for different channel models. For each channel condition, the corresponding BER performance is presented in Table 1.

For a source coding rate of 0.5 bpp, Tables 2–4 present the simulation results for the AWGN channel and flat fading channels with and without CSI. In each table, the second column shows the values of $t(\gamma)$ and $t(\overline{\gamma})$. The quantity $t(\overline{\gamma})$ is divided by either a or $E[a]$ for channels with or without CSI to form the modification factors, respectively. The last two columns show the PSNR (dB) and mean number of iterations in pairs corresponding to without/with use of a joint decoding strategy.

Data in the three tables are plotted in Figures 8 and 9. Figure 8 illustrates the mean number of iterations for systems employing a JSCD design as well as for systems not using a joint decoding design. It is obvious that for all the channel models, the JSCD system requires less decoder iteration, which means that the overall decoding time can be reduced. For an AWGN channel, the decoding time can be reduced by as much as 2.16% to 16.93%. The decoding time is reduced by 2.43% to 15.42% for the fading channel case. Figure 9 shows the qualities of the reconstructed images both with a JSCD design and without a joint decoding design employed at the receiver. In all the channel models, the PSNR gain becomes smaller with an increase in the channel SNR. Also, Figure 9 shows that the JSCD method is more effective for the fading channel with CSI than that for the fading channel without CSI. That is due to the fact that $E[a]$ is not a sufficient statistic compared to the instantaneous fading coefficient a. It has been found that a gain of 1.24 dB to 3.04 dB can be obtained on an AWGN channel employing a JSCD design for image transmission, while for a fading channel, the gain in PSNR is up to 2.52 dB when CSI is available. Simulation results illustrating the PSNR gain for the other two source coding rates 0.1 bpp and 1.0 bpp are presented in Figures 10 and 11. The results are similar to the case of 0.5 bpp.

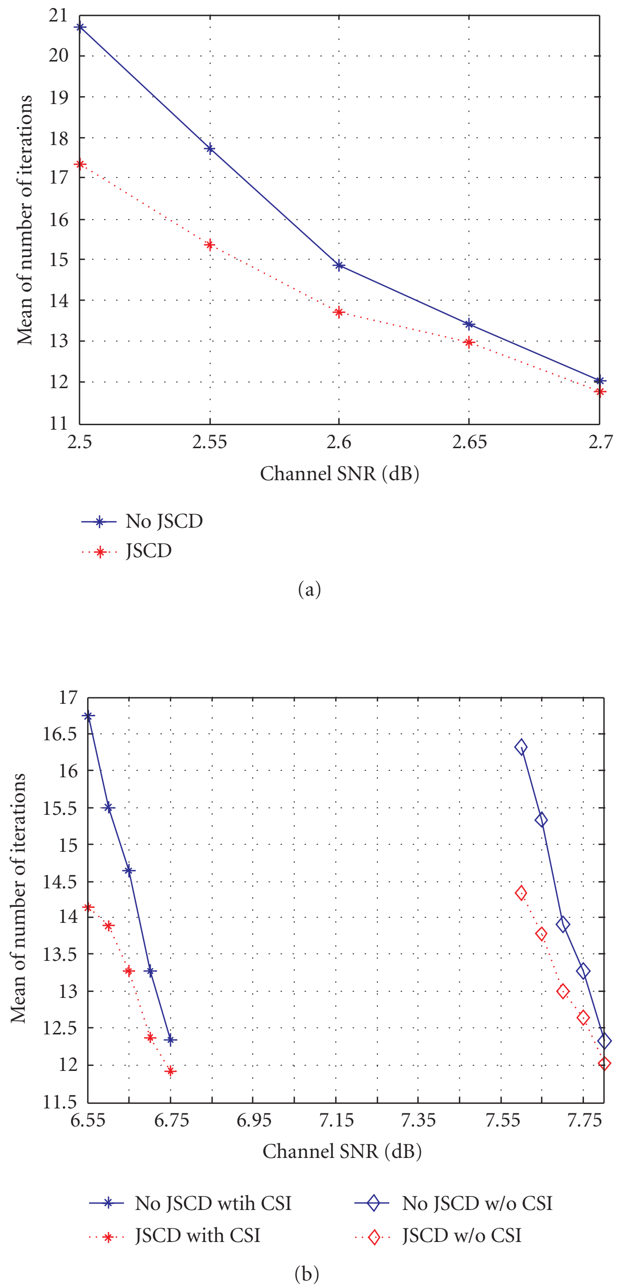


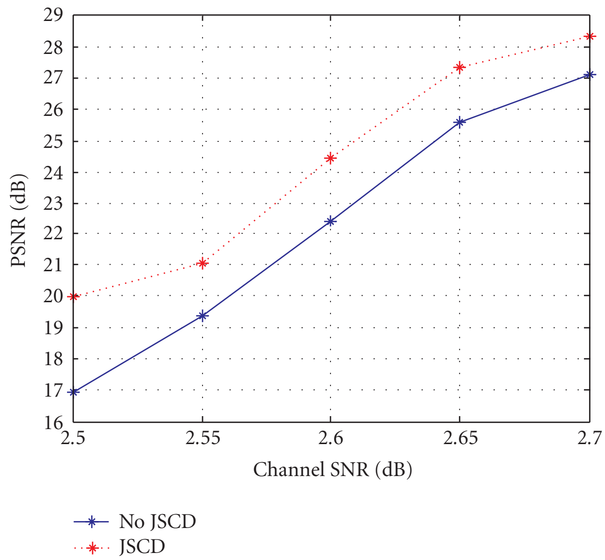
FIGURE 8: Mean number of iterations with and without using a JSCD design. Source coding rate at 0.5 bpp (a) for AWGN channel, (b) for flat fading channels with and without CSI.

5. CONCLUSION

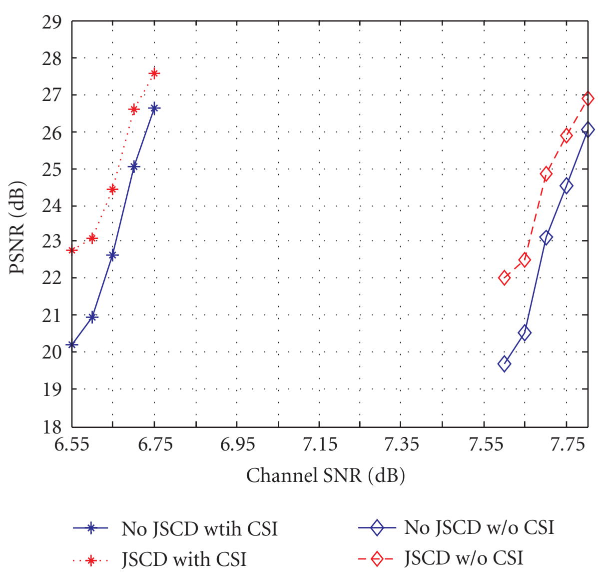
In this paper, we proposed a joint source-channel decoding method for transmitting a JPEG2000 codestream. The iterative log-domain sum-product LDPC decoding algorithm is

Table 1: Channel SNR sets and the corresponding BER performance.

AWGN	2.50	2.55	2.60	2.65	2.70
BER	2.4×10^{-3}	1.26×10^{-3}	5.90×10^{-4}	2.37×10^{-4}	1.90×10^{-4}
Fading CSI	6.55	6.60	6.65	6.70	6.75
BER	1.03×10^{-3}	7.94×10^{-4}	5.03×10^{-4}	2.71×10^{-4}	1.89×10^{-4}
Fading no CSI	7.60	7.65	7.70	7.75	7.80
BER	1.22×10^{-3}	7.87×10^{-4}	4.78×10^{-4}	3.10×10^{-4}	2.19×10^{-4}

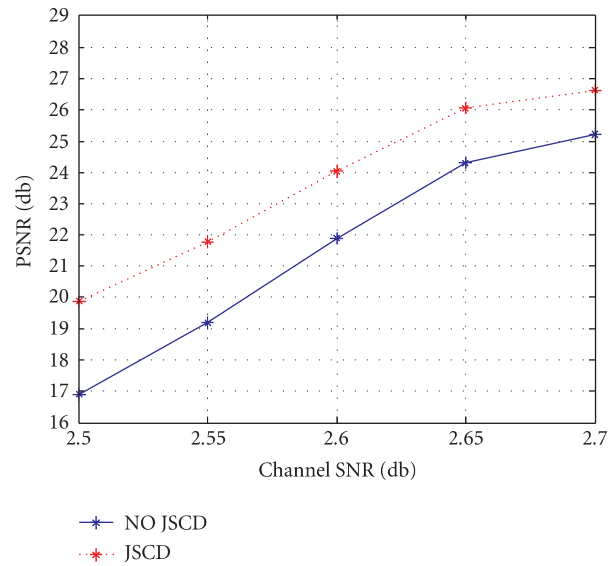


(a)

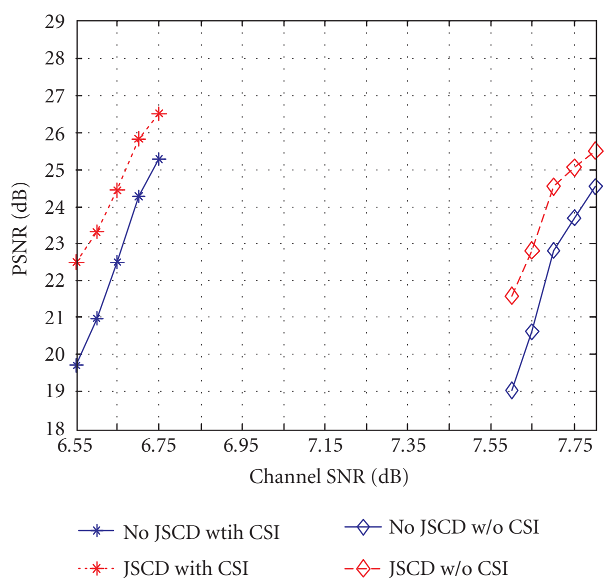


(b)

Figure 9: PSNR with and without using a JSCD design. Source coding rate at 0.5 bpp (a) for AWGN channel, (b) for flat fading channels with and without CSI.



(a)



(b)

Figure 10: PSNR with and without using a JSCD design. Source coding rate at 0.1 bpp (a) for AWGN channel, (b) for flat fading channels with and without CSI.

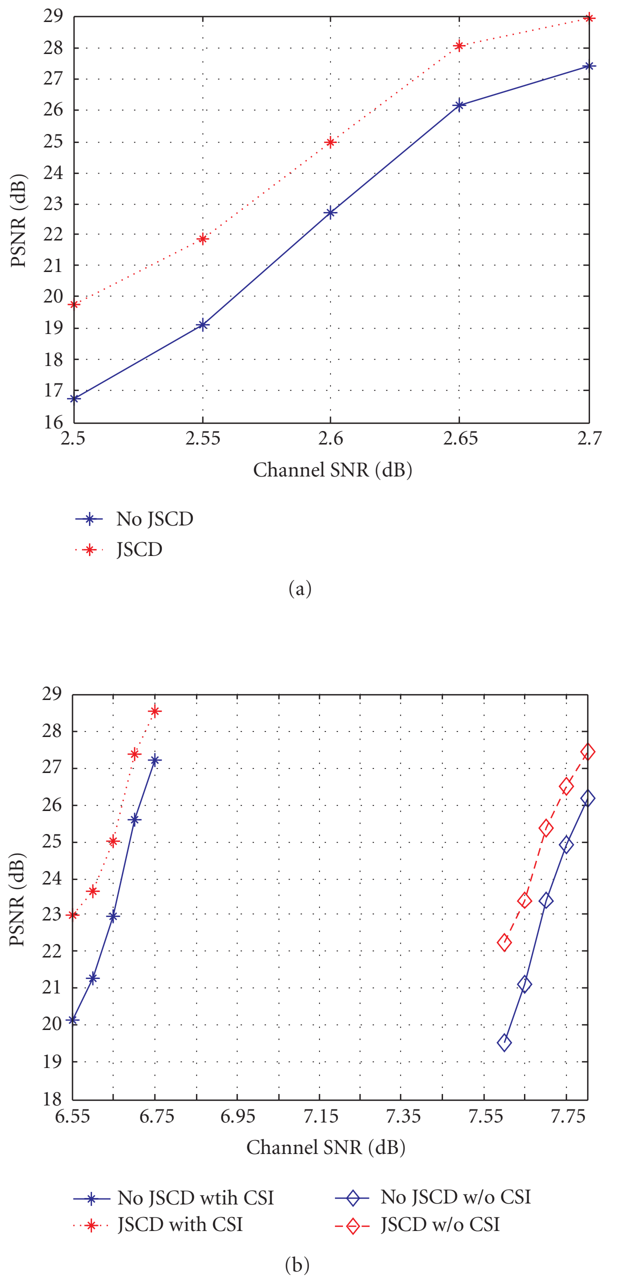


Figure 11: PSNR with and without using a JSCD design. Source coding rate at 1 bpp (a) for AWGN channel, (b) for flat fading channels with and without CSI.

used on both the AWGN and the flat fading channels. The correct coding passes are fed back to update the LLR values after each iteration. The modification factor is chosen to be channel-dependent. Thus, the feedback system adapts to channel variations. Results show that at lower SNR for all the channel models, the proposed method can improve the reconstructed image by approximately 2 to 3 dB in terms of PSNR. Also, the results demonstrate that the joint design method reduces the average number of iterations by up to 3, thereby considerably reducing the decoding time.

Table 2: Joint decoding results for AWGN channel.

$\overline{y}$	$t(\overline{y})$	PSNR	Mean
2.50	8	16.93/19.97/3.04	10.68/17.32/16.25%
2.55	8	19.36/22.03/2.67	17.71/15.33/13.44%
2.60	6	22.44/24.47/2.03	14.84/13.71/7.61%
2.65	5	25.59/27.35/1.76	13.40/12.98/3.13%
2.70	5	27.10/28.34/1.24	12.04/11.78/2.16%

Table 3: Joint decoding results for flat fading channel with CSI.

$\overline{y}$	$t(\overline{y})$	PSNR	Mean
6.55	7	20.21/22.73/2.52	16.73/14.15/15.42%
6.60	7	20.95/23.06/2.11	15.49/13.89/10.33%
6.65	6	22.61/24.45/1.84	14.64/13.27/9.36%
6.70	5	25.07/26.59/1.52	13.27/12.38/6.17%
6.75	4	26.62/27.56/0.94	12.35/11.93/3.40%

Table 4: Joint decoding results for flat fading channel without CSI.

$\overline{y}$	$t(\overline{y})$	PSNR	Mean
7.60	7	19.69/22.02/2.33	16.31/14.34/12.10%
7.65	7	20.51/22.48/1.97	15.33/13.77/10.18%
7.70	6	23.11/24.87/1.76	13.90/13.01/6.40%
7.75	5	24.55/25.89/1.34	13.27/12.65/4.67%
7.80	4	26.06/26.88/0.82	12.33/12.03/2.43%

REFERENCES

[1] B. A. Banister, B. Belzer, and T. R. Fischer, "Robust image transmission using JPEG2000 and turbo-codes," in *Proceedings of the International Conference on Image Processing (ICIP '00)*, vol. 1, pp. 375–378, Vancouver, BC, Canada, September 2000.

[2] Z. Wu, A. Bilgin, and M. W. Marcellin, "An efficient joint source-channel rate allocation scheme for JPEG2000 codestreams," in *Proceedings of Data Compression Conference (DCC '03)*, pp. 113–122, Snowbird, Utah, USA, March 2003.

[3] W. Liu and D. G. Daut, "An adaptive UEP transmission system for JPEG2000 codestream using RCPT codes," in *Proceedings of 38th Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 2265–2269, Pacific Grove, Calif, USA, November 2004.

[4] J. Hagenauer, "Source-controlled channel decoding," *IEEE Transactions on Communications*, vol. 43, no. 9, pp. 2449–2457, 1995.

[5] N. Görtz, "A generalized framework for iterative source-channel decoding," in *Turbo Codes, Error-Correcting Codes of Widening Application*, M. Jézéquel and R. Pyndiah, Eds., pp. 105–126, Kogan Pade Science, Sterling, Va, USA, 2003.

[6] S. T. Brink, "Convergence of iterative decoding," *Electronics Letters*, vol. 35, no. 10, pp. 806–808, 1999.

[7] S. T. Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.

[8] T. Hindelang, T. Fingscheidt, N. Seshadri, and R. V. Cox, “Combined source/channel (de-)coding: can a priori information be used twice?” in *Proceedings of IEEE International Symposium on Information Theory*, p. 266, Sorrento, Italy, June 2000.

[9] M. Adrat, P. Vary, and J. Spittka, “Iterative source-channel decoder using extrinsic information from softbit-source decoding,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’01)*, vol. 4, pp. 2653–2656, Salt Lake, Utah, USA, May 2001.

[10] K. Laković and J. Villasenor, “On reversible variable length codes with turbo codes, and iterative source-channel decoding,” in *Proceedings of IEEE International Symposium on Information Theory*, p. 170, Lausanne, Switzerland, June-July 2002.

[11] M. Adrat, U. von Agris, and P. Vary, “Convergence behavior of iterative source-channel decoding,” in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP ’03)*, vol. 4, pp. 269–272, Hong Kong, April 2003.

[12] Z. Peng, Y.-F. Huang, and D. J. Costello Jr., “Turbo codes for image transmission—a joint channel and source decoding approach,” *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 6, pp. 868–879, 2000.

[13] R. G. Gallager, “Low-density parity-check codes,” *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, 1962.

[14] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.

[15] G. Lechner and J. Sayir, “On the convergence of log-likelihood values in iterative decoding,” in *Proceedings of Mini-Workshop on Topics in Information Theory*, pp. 1–4, Essen, Germany, September 2002.

[16] G. Lechner, “Convergence of sum-product algorithm for finite length low-density parity-check codes,” in *Proceedings of Winter School on Coding and Information Theory*, Monte Verita, Switzerland, February 2003.

[17] M. Ardakani, T. H. Chan, and F. R. Kschischang, “EXIT-chart properties of the highest-rate LDPC code with desired convergence behavior,” *IEEE Communications Letters*, vol. 9, no. 1, pp. 52–54, 2005.

[18] L. Pu, Z. Wu, A. Bilgin, M. W. Marcellin, and B. Vasic, “Iterative joint source/channel decoding for JPEG2000,” in *Proceedings of the 37th Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1961–1965, Pacific Grove, Calif, USA, November 2003.

[19] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.

[20] D. J. C. MacKay and R. M. Neal, “Good codes based on very sparse matrices,” in *Cryptography and Coding: 5th IMA Conference*, C. Boyd, Ed., Lecture Notes in Computer Science, no. 1025, pp. 100–111, Springer, Berlin, Germany, 1995.