

Research Article

Joint Source and Channel Decoding for Variable Length Encoded Turbo Codes

Jianjun Liu,¹ Guofang Tu,¹ Can Zhang,² and Yang Yang¹

¹ School of Information Science and Engineering, Graduate University of Chinese Academy of Sciences, Beijing 100049, China

² State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing 100049, China

Correspondence should be addressed to Jianjun Liu, jjliu@mails.gucas.ac.cn

Received 29 November 2006; Revised 19 April 2007; Accepted 16 September 2007

Recommended by Huaiyu Dai

Joint source and channel decoding (JSCD) has been proved to be an effective technique which can improve decoding performance by exploiting residual source redundancy. Most previous publications on this subject focus on a traditional coding scheme in which the source variable-length coding (VLC) is serially concatenated with a channel code. In this paper, a parallel concatenated coding scheme for the VLC combined with a turbo code is presented. By merging a symbol-level VLC trellis with a convolutional trellis, we construct a symbol-level joint trellis with compound states. Also, a solution of the symbol-by-symbol a posteriori probability (APP) decoding algorithm based on this joint trellis is derived, which leads to an iterative JSCD approach in the similar way to the classical turbo decoder. The simulation results show that our joint source-channel en/decoding system achieves some gains at the cost of increasing decoding complexity, when compared to the joint iterative decoding based on the bit-level super trellis for the separate coding system.

Copyright © 2008 Jianjun Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

Variable-length coding (VLC) is an effective technique to remove source redundancy. It is in turn essential for many communication applications, including text, voice, images, and video. Unfortunately, it is very sensitive to even a single binary error, which leads to error propagation, thus channel coding is always employed after source coding. In the classical communication system, the two coding parts are usually optimized separately, which has been theoretically justified by Shannon's source-channel separation theory [1]. But the separation theory holds only under asymptotic conditions, where both codes are allowed infinite length and complexity. If the practical system is heavily constrained by complexity or delay, the separate source-channel coding can be largely suboptimal. These arguments have motivated the active research areas of joint source and channel coding/decoding (JSCC/JSCD).

Several recent studies on the JSCD technology focus on iterative decoding for the VLC concatenated with a convolutional code through an interleaver. Obvious performance gains can be obtained via iterations between two soft-input

soft-output (SISO) modules. Bauer and Hagenauer [2, 3] proposed an iterative scheme based on a symbol-level VLC trellis and derived a symbol-based *a posteriori* probability (APP) algorithm by modifying the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [4]. They also studied a bit-level iterative decoding [5] based on the Balakirsky's trellis [6], which was well suited for long data packets. The extended work on the VLC serially concatenated [7, 8] or parallel concatenated [9] with a single convolutional code for the first-order Markov source was considered by Kliewer et al. [7, 9] Due to the outstanding decoding performance at low signal-to-noise ratio (SNR) level, turbo codes [10] have entered into service in many communication applications; naturally, the new subject of JSCD for the VLC with a turbo code attracted increasing research interests in recent years. Laković and Villasenor [11] suggested a model of iterative decoding between two SISO modules, in which the first constituent code of a turbo code was decoded based on the bit-level super trellis. Also, Guivarch et al. [12] and Jeanne et al. [13, 14] proposed a method that *a priori* source information of Huffman codes in the form of bit transition probabilities was introduced at a bit level. Peng et al. [15] studied a feedback approach

by modifying the extrinsic information. Additionally, an iterative scheme with three SISO decoding modules was presented by Jaspas and Vandendorpe [16, 17].

We can observe that most of the above joint source-channel decoding methods focus on the classical coding system with serial concatenation. Moreover, the performance of the VLC parallel concatenated with a single convolutional code [9] is not optimistic without the protection for the VLC systematic information. Additionally, in the above cases combined with turbo codes [11–17], residual source redundancy is usually utilized at the decoder side with the bit-level joint decoding, while no improvement has been done at the encoder side. The main contribution of this paper is to present a different parallel joint source-channel coding scheme by combining the VLC with a turbo code, and then suggest its JSCD method based on the proposed symbol-level joint trellis. In this paper, we do not present a rigorous proof of stability and convergence of iterative decoding; however, simulation results indicate that the proposed scheme achieves some gains at the cost of increasing decoding delay, when compared to the bit-level joint iterative decoding scheme [11].

This paper is organized as follows. Section 2 gives the basic transmission system. Section 3 presents the proposed symbol-level joint trellis, and a symbol-level APP decoding algorithm suited for the new joint trellis is illustrated in Section 4. Iterative decoding of our system is discussed in Section 5. The simulation results are presented in Section 6, and conclusions of our work are made in Section 7.

2. TRANSMISSION SYSTEM

The model of our transmission system is depicted in Figure 1. Different from the traditional turbo coding system with serial concatenation, the VLC is integrated with the upper recursive systematic convolutional (RSC) code of a turbo code into a single constituent code. We assume the VLC is implemented on a K -symbol source sequence $\mathbf{U} = [U_1, U_2, \dots, U_K]$, where each $U_k, k = 1, 2, \dots, K$ from a finite source alphabet \mathcal{U} , must be mapped to a variable length codeword $c(U_k)$. The output of the source variable length encoder can be denoted as $\mathbf{C}(\mathbf{U}) = [c(U_1), c(U_2), \dots, c(U_K)]$, which is composed of K variable length codewords, or denoted as a binary sequence $\mathbf{w}_s = [w_{s1}, w_{s2}, \dots, w_{sN_1}]$ with the total bit length N_1 . The VLC sequence \mathbf{w}_s is then protected by the RSC1 code, which produces a parity check sequence $\mathbf{w}_p = [w_{p1}, w_{p2}, \dots, w_{pN_1}]$. Utilizing a Q -bit quantizer, the symbol sequence \mathbf{U} is converted into a bit sequence $\mathbf{U}' = [U'_1, U'_2, \dots, U'_K]$, where $U'_k = [u'_{k1}, u'_{k2}, \dots, u'_{kQ}]$, $k = 1, 2, \dots, K$. This bit sequence \mathbf{U}' is interleaved by Π , and then channel coded by another RSC2 encoder. However, only the parity check sequence $\mathbf{v}'_p = [v'_{p1}, v'_{p2}, \dots, v'_{pN_2}]$ is reserved since the systematic information has already been included in \mathbf{w}_s . In our case, the RSC2 encoder with the memory length μ_2 is terminated, so the sequence length N_2 equals $(K \cdot Q + \mu_2)$. Note that quantizing the source symbols directly and then coding the sequence \mathbf{U}' with the RSC2 encoder increases some redundancy, while the higher channel code rate can be

achieved by puncturing the parity check sequence \mathbf{v}'_p to \mathbf{v}_p . Finally, \mathbf{w}_s , \mathbf{w}_p and \mathbf{v}_p are passed through a multiplexer and then sent to the wireless channel.

We assume the coherently detected binary phase-shift keying (BPSK) modulation, and signals are transmitted over the additive white Gaussian noise (AWGN) channel. After the channel output is received, an iterative JSCD between two SISO modules is carried out in order to obtain the decoding output $\hat{\mathbf{U}} = [\hat{U}_1, \hat{U}_2, \dots, \hat{U}_k]$.

3. REPRESENTATION OF JOINT TRELLIS

As mentioned above, in our joint turbo coding scheme, the serial concatenated VLC and an RSC code are treated as a single constituent code, it should be appropriate to deal with decoding of these two parts as a single module, which has inspired us to construct a symbol-level joint trellis.

We assume a K -symbol source sequence is variable-length encoded to an N -bit binary sequence. As suggested by Bauer and Hagenauer [3], this process can be denoted by a symbol-level VLC trellis. Figure 2 gives an example of the trellis representation, which corresponds to a four-symbol code table $\mathcal{C} = \{c(0) = [1], c(1) = [0, 1], c(2) = [0, 0, 0], c(3) = [0, 0, 1]\}$, with the constraint of $K = 5$ and $N = 10$. In the trellis, the state index n denotes the bit length of the sequence after k source symbols have been variable-length encoded (e.g., $n_3 = 7$), and the state transition from $n_{k-1} = n_1$ to $n_k = n_2$ is caused by a variable length codeword $c_k \in \mathcal{C}$, with the bit length $l(c_k) = n_2 - n_1$. Especially, all available states at the symbol instant k belong to a subset \mathcal{R}_k (e.g., \mathcal{R}_3). With the transform

$$v = n - k \cdot l_{\min}, \quad (1)$$

where l_{\min} is the minimum codeword length in the table \mathcal{C} , the symbol-level VLC trellis in Figure 2 can be transformed to another VLC KV-trellis [3] with a different state index v instead of n .

We can further represent the process of the VLC and the convolutional coding by a single joint KT-trellis. At the symbol instant k , if a state in the VLC KV-trellis is denoted as v_k , and a state in the convolutional trellis is denoted as S_k (the value of shift registers), then a certain state in our joint KT-trellis can be written as $T_k = (v_k, S_k)$, which actually consists of two substates. An example under the constraint of $K = 5$ and $v_{\max} = 5$ (the maximal v caused by $N = 10$ and $l_{\min} = 1$) is shown in Figure 3, which is derived from the KV-trellis for the code table \mathcal{C} and a two-state RSC encoder with code polynomials $G_1 = (3, 1)_8$. Similarly to Figure 2, all available states at the symbol instant k belong to a set \mathcal{R}_k (e.g., \mathcal{R}_3). Especially, each transition (T_{k-1}, T_k) between two state nodes must correspond to a pair of variable-length input/output codewords (e.g., $c(1)/0010$). In Figure 3, there are two terminating states in the joint trellis since a two-state RSC code is considered. This time-varying joint trellis can be utilized for the symbol-level APP decoding.

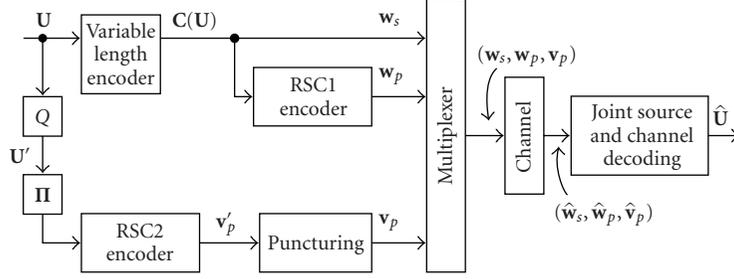
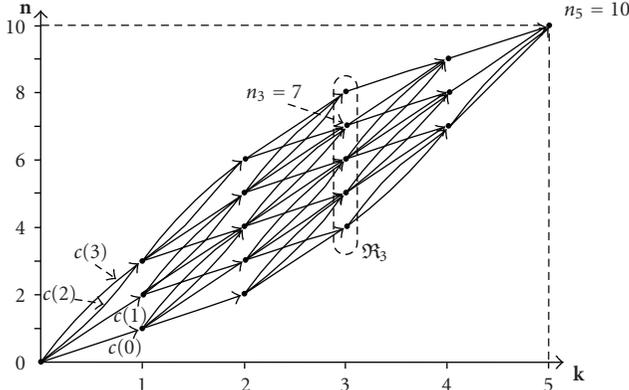
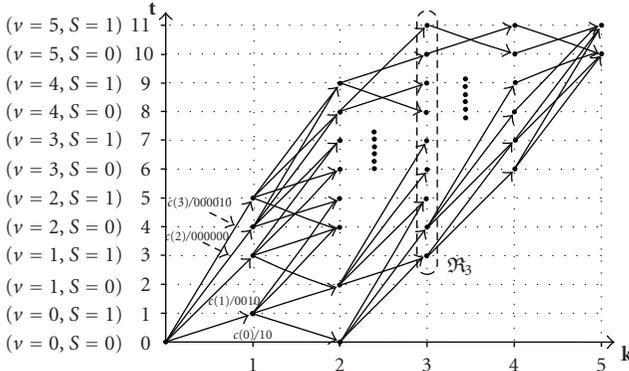


FIGURE 1: The model of the transmission system.

FIGURE 2: Symbol-level VLC trellis for $\mathcal{C} = \{c(0) = [1], c(1) = [0, 1], c(2) = [0, 0, 0], c(3) = [0, 0, 1]\}$, with $K = 5$ and $N = 10$.FIGURE 3: Joint KT-trellis representation with compound states derived from the VLC KV-trellis for \mathcal{C} and a two-state RSC code.

4. SYMBOL-LEVEL APP ALGORITHM FOR JOINT TRELLIS

In this section, we give a description of the modified APP algorithm suitable for the symbol-level joint trellis, especially, an independent memoryless source is considered here. Codeword sequences \mathbf{w}_s and \mathbf{w}_p from the first constituent code are compounded and BPSK-modulated into a sequence $\mathbf{X}_1^{2N_1} = [x_1, x_2, \dots, x_{2N_1}]$ before being sent to a wireless channel. Let $\mathbf{Y}_1^{2N_1} = [y_1, y_2, \dots, y_{2N_1}]$ represent the channel observations of $\mathbf{X}_1^{2N_1}$, and its subsequence from the bit position

a to b is indicated as $\mathbf{Y}_a^b = [y_a, y_{a+1}, \dots, y_b]$. At the symbol instant k , a VLC codeword $c_k = c(i)$ with the length $l(c(i))$ is channel coded and then BPSK-modulated into a codeword $x_k(i, t', t)$ with the length $2 \cdot l(c(i))$, which is associated with the state transition ($T_{k-1} = t', T_k = t$). In addition, if the bit length of the VLC sequence \mathbf{w}_s associated with a compound state t is denoted as $n(t)$, we can represent the bit length of the channel sequence produced by the upper constituent code as $m(t) = 2 \cdot n(t)$, and the maximum of $m(t)$ should be $M = 2 \cdot N_1$. Note that, the transform between the parameter $m(t)$ and the substate $v(t)$ can be obtained from (1).

The key point of our decoding algorithm is to calculate symbol-based APPs for each VLC codeword $c_k = c(i)$ giving the observations \mathbf{Y}_1^M . Using Bayesian principles, we have

$$P(c_k = c(i)/\mathbf{Y}_1^M) = C \cdot \sum_{t \in \mathcal{R}_k} \sum_{t' \in \mathcal{R}_{k-1}} \underbrace{p(\mathbf{Y}_{m(t)+1}^M / T_k = t)}_{\beta_k(t)} \cdot \underbrace{p(\mathbf{Y}_{m(t')+1}^{m(t)}, c_k = c(i), T_k = t/T_{k-1} = t')}_{\gamma_k^i(\mathbf{Y}_{m(t')+1}^{m(t)}, t', t)} \cdot \underbrace{p(T_{k-1} = t', \mathbf{Y}_1^{m(t')})}_{\alpha_{k-1}(t')}, \quad (2)$$

where $C = 1/p(\mathbf{Y}_1^M)$ is a constant term. We name $\alpha_k(t)$ as the forward recursion, $\beta_k(t)$ as the backward recursion, and $\gamma_k^i(\mathbf{Y}_{m(t')+1}^{m(t)}, t', t)$ as the transition probability from t' to t associated with the input codeword $c_k = c(i)$, respectively. The forward recursion $\alpha_k(t)$ can be calculated from

$$\alpha_k(t) = \sum_{t' \in \mathcal{R}_{k-1}} \sum_i \gamma_k^i(\mathbf{Y}_{m(t')+1}^{m(t)}, t', t) \cdot \alpha_{k-1}(t'), \quad (3)$$

$$\alpha_0(0) = 1.$$

Similarly, the backward recursion can be calculated from

$$\beta_k(t) = \sum_{t' \in \mathcal{R}_{k+1}} \sum_i \gamma_{k+1}^i(\mathbf{Y}_{m(t)+1}^{m(t')}, t, t') \cdot \beta_{k+1}(t'). \quad (4)$$

If the memory length of the RSC1 encoder is μ_1 , the initial conditions for performing the backward recursion are

$$\beta_K(t) = \begin{cases} 1/2^{\mu_1} & \text{if } v(t) = v_{\max} \\ 0 & \text{else.} \end{cases} \quad (5)$$

Using Bayesian principles, the transition probability can be finally factorized into three terms as follows:

$$\begin{aligned} \gamma_k^i(\mathbf{Y}_{m(t')_+1}^{m(t)}, t', t) &= p(\mathbf{Y}_{m(t')_+1}^{m(t)}/x_k(i, t', t)) \\ &\cdot P(T_k = t/c_k = c(i), T_{k-1} = t') \quad (6) \\ &\cdot P(c_k = c(i)/T_{k-1} = t'). \end{aligned}$$

Note that, the implementation of the above symbol-level algorithm should be performed in logarithm domain [18].

5. ITERATIVE DECODING FOR VARIABLE LENGTH ENCODED TURBO CODES

The basic iterative decoding model of the system is shown in Figure 4. We denote a priori information as \mathbf{L}_{ai} ($i = 1, 2$) and logarithm likelihood ratios as \mathbf{L}_i ($i = 1, 2$) for the two constituent decoders. The inner decoder for the RSC2 code first decodes the observations by a bit-level APP algorithm, and the bit-based extrinsic information \mathbf{L}_{e2} is passed to the outer decoder after being deinterleaved by Π^{-1} . Whereafter, the joint symbol-level APP decoder carries out a symbol-based APP algorithm as described in Section 4, however, the feedback information from the outer decoder to the inner decoder is the systematic extrinsic information $\mathbf{L}_{s\&e1}$ owing to the nonsystematic property of the VLC. After the last iteration, a symbol decision is made on \mathbf{L}_1 and we get the symbol sequence estimation $\hat{\mathbf{U}}$.

The function of \mathbf{T}^{-1} in Figure 4 is to transform a priori information from bit levels to symbol levels. Since the VLC is performed as a special one-to-one mapping from a source symbol $U_k = i$, $i \in \mathcal{U}$, to a VLC codeword $c_k = c(i)$, it is equivalent to represent a priori information for a codeword $c_k = c(i)$ by a priori information for a source symbol $U_k = i$, as

$$L_a(c_k = c(i)) \Leftrightarrow L_a(U_k = i) = \log \frac{P_a(U_k = i)}{P_a(U_k = 0)}. \quad (7)$$

We further assume all bits u'_{kl} , $l = 1, 2, \dots, Q$ within the quantized codeword U'_k are uncorrelated. Then symbol a priori probability $P_a(U_k = i)$ can be calculated from a multiplication of several bit a priori probabilities $P_a(u'_{kl} = i_l)$ as

$$P_a(U_k = i) = \prod_{l=1}^Q P_a(u'_{kl} = i_l), \quad (8)$$

where $i_l \in \{0, 1\}$ is the l th bit of the quantized symbol i . Utilizing (7) and (8), symbol a priori information $L_a(U_k = i)$ can be finally written as the summation of those bit a priori information $L_a(u'_{kl})$ with $i_l = 1$ as follows:

$$L_a(U_k = i) = \sum_{l=1}^Q \log \frac{P_a(u'_{kl} = i_l)}{P_a(u'_{kl} = 0)} = \sum_{l=1:i_l=1}^Q L_a(u'_{kl}). \quad (9)$$

Correspondingly, the function of \mathbf{T} in Figure 4 is to convert the symbol-based a priori information $L_a(U_k = i)$ to the bit-based a priori information $L_a(u'_{kl})$ according to

$$L_a(u'_{kl}) = \log \frac{\sum_{i:i_l=1} P_a(U_k = i)}{\sum_{i:i_l=0} P_a(U_k = i)}. \quad (10)$$

We further replace all probability terms $P_a(U_k = i)$ in (10) by $L_a(U_k = i)$ according to (7). Based on the Jacobian logarithm [18], the bit-based a priori information can be approximately calculated from

$$L_a(u'_{kl}) \approx \max_{i:i_l=1} (L_a(U_k = i)) - \max_{i:i_l=0} (L_a(U_k = i)). \quad (11)$$

6. SIMULATION RESULTS

In this section, simulations were performed over the AWGN channel with BPSK in order to access the performance of the proposed joint en/decoding approach. The VLC was carried out on the independent memoryless source with 4 symbols, and the corresponding Huffman codes and reversible VLCs (RVLCs) are listed in Table 1 [3]. In our system, the first constituent code was selected to be $G_{p1} = (3, 1)_8$ in order to reduce the decoding complexity, and the second one was selected to be $G_{p2} = (11, 12)_8$ according to [9]. However, both constituent codes were not optimal yet. The interleaver permuted the bit sequence pseudorandomly, where trellis termination of the RSC2 encoder was considered. Simulation comparisons were done between the proposed scheme and the joint iterative decoding scheme in [11], in which the VLC was serially concatenated with a turbo code and the upper constituent code of the turbo code was decoded based on the bit-level super trellis. Due to the higher complexity of the proposed symbol-based decoding, we selected both constituent codes to be $G_S = (11, 12)_8$ for the separate coding scheme, in which the memory length of the upper RSC code was enlarged to $\mu_1 = 3$.

In order to reduce the simulation delay, the bit stream was divided into several short packets with a fixed number of symbols ($K = 100$) as in [3, 7, 9]. Each packet was interleaved and transmitted independently. Furthermore, we assumed the parameters K and N_1 were protected by a strong channel code and thus obtained at the receiver side without errors. The overall code rates for the separate coding scheme (R_S) and the proposed coding scheme (R_P) are denoted as

$$\begin{aligned} R_S &= \frac{K \cdot H(\mathcal{U})}{(K \cdot L_{av} + \mu_1)/R_{c1} + (K \cdot L_{av} + \mu_2)/R_{c2}}, \\ R_P &= \frac{K \cdot H(\mathcal{U})}{K \cdot L_{av}/R_{c1} + (K \cdot Q + \mu_2)/R_{c2}}, \end{aligned} \quad (12)$$

where $H(\mathcal{U})$ is the source entropy, L_{av} is the average codeword length, and R_{ci} ($i = 1, 2$) is the code rate for the i th constituent code, respectively.

At the receiver side, the channel observations were decoded with the Max-LogMAP algorithm [18]. Simulations were carried out for each E_b/N_0 in dB, where E_b denotes the average energy per information bit, and N_0 is the single-sided noise power spectral density. The symbol error rates (SERs) were accounted using the Levenshtein distance [19]. Simulations at the 4th and the 8th iterations were performed. Figure 5 shows the results for the Huffman codes with the overall code rate 0.330, and Figure 6 shows the results for the reversible VLCs with the overall code rate 0.308. These overall code rates were obtained by setting all R_{c1} to be 1/2 and

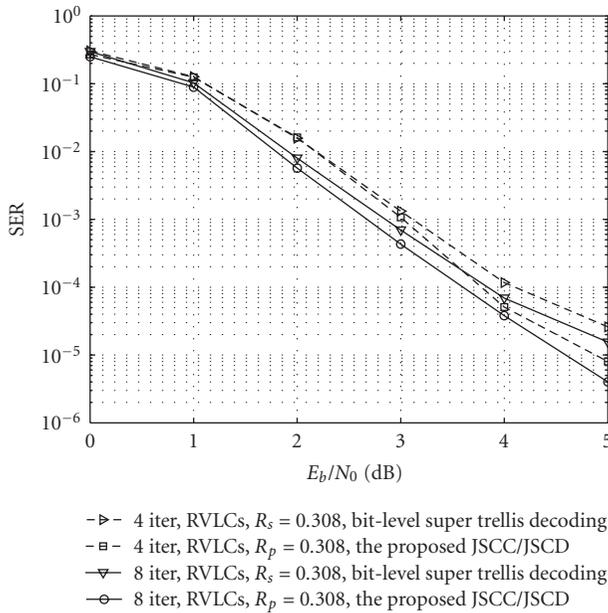


FIGURE 6: Simulation results for the RVLCs on the AWGN channel, $K = 100$, and the overall code rate is 0.308.

with the packet length. If J_{VLC} denotes the number of states in the stationary section, which generally takes on the maximal value ($v_{\max} + 1$), the state number at each symbol instant in our joint trellis is changeable but no more than $2^{J_{VLC}}$. Therefore, we are currently making efforts to research low-complexity decoding algorithms, which would be applicable to the practical system with higher memory or long data packets.

7. CONCLUSION

We have presented a joint source-channel coding scheme by parallel concatenating the VLC with a turbo code. To explain the basic concept of our idea, the two-state RSC code is considered as an example. A symbol-level joint trellis is derived through merging a symbol-level VLC trellis with a convolutional trellis, based on which, the symbol-by-symbol APP decoding algorithm can be implemented. A JSCD approach is obtained similarly to the classical turbo decoder. Simulation results show that our scheme can achieve some gains compared to the joint iterative decoding with the bit-level super trellis, at the cost of decoding complexity. The proposed scheme could be applied to robust transmission for variable-length coded image data.

ACKNOWLEDGMENTS

The authors would like to thank the editors and the anonymous reviewers for their helpful comments. This work was supported by the National Natural Science Foundation of China under Grants nos. 90304003 and 60573112 and the "Eleventh Five-year" Project of China.

REFERENCES

- [1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, 1948.
- [2] R. Bauer and J. Hagenauer, "Iterative source/channel-decoding using reversible variable length codes," in *Proceedings of Data Compression Conference (DCC '00)*, pp. 93–102, Snowbird, Utah, USA, March 2000.
- [3] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in *Proceedings of the 3rd ITG Conference on Source and Channel Coding (CSCC '00)*, pp. 111–116, Munich, Germany, January 2000.
- [4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.
- [5] R. Bauer and J. Hagenauer, "On variable length codes for iterative source/channel decoding," in *Proceedings of Data Compression Conference (DCC '01)*, pp. 273–282, Snowbird, Utah, USA, March 2001.
- [6] V. B. Balakirsky, "Joint source-channel coding with variable length codes," in *Proceedings of IEEE International Symposium on Information Theory (ISIT '97)*, p. 419, Ulm, Germany, June–July 1997.
- [7] J. Kliewer and R. Thobaben, "Iterative joint source-channel decoding of variable-length codes using residual source redundancy," *IEEE Transactions on Wireless Communications*, vol. 4, no. 3, pp. 919–929, 2005.
- [8] R. Thobaben and J. Kliewer, "Low-complexity iterative joint source-channel decoding for variable-length encoded Markov sources," *IEEE Transactions on Communications*, vol. 53, no. 12, pp. 2054–2064, 2005.
- [9] J. Kliewer and R. Thobaben, "Parallel concatenated joint source-channel coding," *Electronics Letters*, vol. 39, no. 23, pp. 1664–1666, 2003.
- [10] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Transactions on Communications*, vol. 44, no. 10, pp. 1261–1271, 1996.
- [11] K. Laković and J. Villasenor, "Combining variable length codes and turbo codes," in *Proceedings of the 55th IEEE Vehicular Technology Conference (VTC '02)*, vol. 4, pp. 1719–1723, Birmingham, Ala, USA, May 2002.
- [12] L. Guivarch, J.-C. Carlach, and P. Siohan, "Joint source-channel soft decoding of Huffman codes with turbo-codes," in *Proceedings of Data Compression Conference (DCC '00)*, pp. 83–92, Snowbird, Utah, USA, March 2000.
- [13] M. Jeanne, J. C. Carlach, P. Siohan, and L. Guivarch, "Source and joint source-channel decoding of variable length codes," in *Proceedings of IEEE International Conference on Communications (ICC '02)*, vol. 2, pp. 768–772, New York, NY, USA, April–May 2002.
- [14] M. Jeanne, J.-C. Carlach, and P. Siohan, "Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes," *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 10–15, 2005.
- [15] 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.
- [16] X. Jaspard and L. Vandendorpe, "New iterative decoding of variable length codes with turbo-codes," in *Proceedings of IEEE International Conference on Communications*, vol. 5, pp. 2606–2610, Paris, France, June 2004.

-
- [17] X. Jaspas and L. Vandendorpe, "Three SISO modules joint source-channel turbo-decoding of variable length coded images," in *Proceedings of the 5th International ITG Conference on Source and Channel Coding (SCC '04)*, pp. 279–286, Erlangen, Germany, January 2004.
 - [18] P. Robertson, E. Villebrun, and P. Hoher, "Comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proceedings of the IEEE International Conference on Communications*, vol. 2, pp. 1009–1013, Seattle, Wash, USA, June 1995.
 - [19] T. Okuda, E. Tanaka, and T. Kasai, "A method for the correction of garbled words based on the levenshtein metric," *IEEE Transactions on Computers*, vol. 25, no. 2, pp. 172–178, 1976.
 - [20] S. Ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Transactions on Communications*, vol. 49, no. 10, pp. 1727–1737, 2001.