Iterative Joint Source-Channel Decoding of Variable-Length Codes Using Residual Source Redundancy

Jörg Kliewer, *Senior Member, IEEE,* and Ragnar Thobaben

***Abstract*—We present a novel symbol-based soft-input *a posteriori* probability (APP) decoder for packetized variable-length encoded source indexes transmitted over wireless channels where the residual redundancy after source encoding is exploited for error protection. In combination with a mean-square or maximum APP estimation of the reconstructed source data, the whole decoding process is close to optimal. Furthermore, solutions for the proposed APP decoder with reduced complexity are discussed and compared to the near-optimal solution. When, in addition, channel codes are employed for protecting the variable-length encoded data, an iterative source-channel decoder can be obtained in the same way as for serially concatenated codes, where the proposed APP source decoder then represents one of the two constituent decoders. The simulation results show that this iterative decoding technique leads to substantial error protection for variable-length encoded correlated source signals, especially, when they are transmitted over highly corrupted channels.**

***Index Terms*—Iterative decoding, joint source-channel coding, residual source redundancy, variable-length codes (VLCs).**

I. Introduction

THE mobile access of multimedia data has recently emerged as an attractive application for current and future wireless systems. Since many source-coding standards employ variable-length codes (VLCs) for increasing the source compression performance, a reliable transmission of such variable-length encoded source signals over wireless channels has become an active area of research during the last years. Especially joint-source channel coding techniques seem to be attractive because the classical separation between source and channel coding appears not to be justified in practical systems with limited block lengths. Some of these approaches focus on robust decoding of such variable-length encoded streams using joint source-channel decoding techniques.

In [1], a soft-input dynamic programming approach is presented, which uses the probability distribution of the source-encoded indexes and the number of source symbols as *a priori* information in the decoding process. However, in [2] and [3], the authors follow a different strategy and present a trellis representation for VLCs, which allows them to utilize the Bahl–Cocke–Jelinek–Raviv (BCJR) algorithm [4] for soft-input *a posteriori* probability (APP) decoding. This has the advantage that reliability information for the source indexes can be obtained at the output of the decoder. A similar trellis is used in [5], where maximum *a posteriori* probability (MAP) and sequential decoding for MPEG 4 reversible VLCs are compared.

Whereas these approaches only consider memoryless sources, the methods introduced in [6]–[10] use a first-order Markov model for the source and exploit the correlation of the source symbols in the decoding process. In [6], an extension of the list Viterbi algorithm to VLCs is presented, where the source data is protected by a nonbinary convolutional code prior to Huffman encoding. On the other hand, being similar to [1], the method in [9] uses a MAP sequence estimation with dynamic programming based on an unidirectional graph. In [10], the VLC decoding problem is treated within the framework of Bayesian networks.

In this paper, we present a joint source-channel decoding approach for packetized variable-length encoded source indices, which is based on the VLC trellis representation from [2]. As a new result, we show for the first-order Markov source model that by modification of the BCJR algorithm the source correlation can be included in the VLC trellis [11]. This leads to a novel soft-input APP source decoding algorithm, which uses implicit residual source redundancy after source encoding to increase the robustness of the variable-length encoded source indexes against channel errors. For an additional error protection by channel codes, one important issue in this paper is to utilize the proposed source decoder as constituent decoder in an iterative source-channel decoding scheme. Therefore, we focus on symbol-by-symbol VLC decoding throughout this paper since in contrast to a sequence estimation approach (e.g., as in [7]) a symbol-by-symbol decoding approach (as the BCJR algorithm) is especially well suited to generate soft output in the form of *true* APPs. These APPs can then be used as soft input for subsequent (iterative) decoding steps. Additionally, since the computational complexity of the proposed VLC decoding scheme may become quite high, especially if the source indexes are encoded with a large number of bits, we also present methods for complexity reduction.

The outline of the paper is as follows. Section II introduces the overall transmission scenario and specifies the model for the transmission channel. In Section III, the soft-input APP VLC

Manuscript received March 22, 2002; revised May 6, 2003, February 9, 2004, and February 25, 2004. The editor coordinating the review of this paper and approving it for publication is A. U. H. Sheikh. This paper was presented in part at the IEEE Data Compression Conference, Snowbird, UT, USA, April 2002, and at the IEEE International Symposium on Information Theory, Lausanne, Switzerland, July 2002.

The authors are with the University of Kiel, Institute for Circuits and Systems Theory, 24143 Kiel, Germany (e-mail: jkl@tf.uni-kiel.de; rat@tf.uni-kiel.de).

Digital Object Identifier 10.1109/TWC.2005.847032

1536-1276/$20.00 © 2005 IEEE

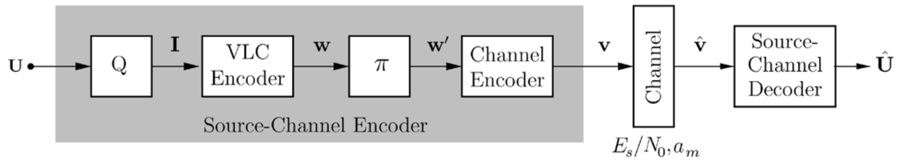


Fig. 1. Model of the transmission system.

decoder is derived, where the near-optimal behavior in combination with a mean-square (MS) or MAP estimation is shown. Section IV discusses possibilities for reducing the computational complexity of the proposed VLC decoding approach. In Section V, we address the case when an additional forward error correction is used and present the corresponding iterative decoder. Finally, Section VI refers to the simulation results.

II. Transmission System

Let us consider the block diagram of the transmission system depicted in Fig. 1. One packet $\mathbf{U} = [U_1, U_2, \ldots, U_K]$ of the source signal consists of K source symbols U_k, where k denotes the time instant. After subsequent (vector-) quantization, the resulting indexes $I_k \in \mathcal{I}$ from the finite alphabet $\mathcal{I} = \{0, 1, \ldots, 2^M - 1\}$ are represented with M bits. We can generally assume that there is a certain amount of redundancy in the index vector $\mathbf{I} = [I_1, I_2, \ldots, I_K]$ due to delay and complexity constraints for the quantization stage. In the following, the correlation between the indexes I_k is modeled by a first-order stationary Markov process with index transition probabilities $P(I_k = \lambda \mid I_{k-1} = \mu)$ for $\lambda, \mu \in \mathcal{I}$.

The VLC encoder in Fig. 1 now maps each fixed-length index I_k to a variable-length bit vector $\mathbf{c}^{(\lambda)} = \mathcal{C}(I_k = \lambda)$ using the prefix code $\mathcal{C}$. Concatenating these bit vectors leads to a binary sequence $\mathbf{w} = [w_1, w_2, \ldots, w_N]$ of length N, where w_n represents a single bit at bit index n. The interleaved bitstream $\mathbf{w}'$ is then encoded with a rate-R channel code. The resulting codewords form the bit vector $\mathbf{v}$ and are eventually transmitted over the communication channel, where both Rayleigh and additive white Gaussian noise (AWGN) channels are employed in the sequel. The Rayleigh channel serves as a good model for a mobile radio channel without a direct line-of-sight component. For the sake of simplicity, we consider a narrowband system and assume that the channel is flat and fully interleaved, i.e., artificially made statistically independent. In addition, we assume coherently detected binary phase-shift keying (BPSK) for the modulation. Then, the conditional pdf of the received soft-bit $\hat{v}_m \in \mathbb{R}$, $m = 1, 2, \ldots, N/R$, at the channel output given the transmitted bit $v_m \in \{0, 1\}$ is Gaussian-distributed with mean $a_m \bar{v}_m$ where $\bar{v}_m = 1 - 2 \cdot v_m$ and can be written as [1]

$$p(\hat{v}_m \mid v_m) = \frac{1}{\sqrt{2\pi}\sigma_e} \cdot e^{-(1/2\sigma_e^2)(\hat{v}_m - a_m \cdot \bar{v}_m)^2}. \qquad (1)$$

Herein, $\sigma_e^2 = N_0/(2E_s)$ denotes the channel noise variance, where E_s is the energy to transmit each bit and N_0 the one-sided power spectral density of the noise. The Rayleigh-distributed random process associated with the amplitude variation factors a_m is white due to the assumed statistical independence of the channel. Equation (1) can also be expressed using the conditional log-likelihood ratios (L-values)

$$\begin{aligned} L(\hat{v}_m \mid v_m) &= \ln\left(\frac{p(\hat{v}_m \mid v_m = 0)}{p(\hat{v}_m \mid v_m = 1)}\right) = 4\frac{E_s}{N_0} a_m \hat{v}_m \\ &= L_c a_m \hat{v}_m \text{ with } L_c = 4\frac{E_s}{N_0}. \end{aligned} \qquad (2)$$

If $a_m = 1$ for all m, we have the special case of an AWGN channel.

III. Soft-Input Soft-Output Source Decoding

In this section, we only deal with the error correcting capabilities of the implicit residual source redundancy after quantization and derive a soft-input APP decoder for the resulting variable-length encoded sequence. Thus, in this special case, our general source-channel encoder in Fig. 1 just contains the quantizer and the VLC encoder, and we therefore have $\mathbf{v} = \mathbf{w}$.

A. Trellis Representation

While in the absence of transmission errors, variable-length encoded sequences can easily be decoded by parsing the bitstream, the decoding of corrupted VLC sequences is rather difficult due to a possible loss of synchronization between bit and symbol time. In this case, the position of a codeword $\mathbf{c}^{(\lambda)}$ corresponding to a source index $I_k = \lambda$ at symbol time k may not be exactly localized in the received variable-length encoded sequence $\hat{\mathbf{w}}$. Therefore, all possible bit positions $n \in \mathcal{N}_k$ for a given time instant k, which are represented by the index set $\mathcal{N}_k$, must be considered as the starting point for the codeword $\mathbf{c}^{(\lambda)}$. This can be realized by setting up an appropriate trellis for the VLC sequence, where we utilize the VLC trellis representation derived by Bauer and Hagenauer [2] for the proposed APP VLC source decoding approach. The trellis states are given by $S_k = n$, where S_k refers to the state at time instant k, and $n \in \mathcal{N}_k$ represents a hypothesis for the actual bit position in the encoded bit sequence $\mathbf{w}$.

An example for this trellis representation is given in Fig. 2 for $K = 5$, $N = 10$, and the Huffman code

$$\mathcal{C} = \{\mathbf{c}^{(0)} = [1], \mathbf{c}^{(1)} = [0, 1], \mathbf{c}^{(2)} = [0, 0, 0], \mathbf{c}^{(3)} = [0, 0, 1]\}.$$

As we can observe from Fig. 2, a transition from state $S_{k-1} = n_1$ to $S_k = n_2$ is caused by the source symbol $I_k = \lambda$, $\lambda \in \mathcal{I}$, and the corresponding variable-length codeword $\mathbf{c}^{(\lambda)}$ with length $l(\mathbf{c}^{(\lambda)}) = n_2 - n_1$, respectively. A code table which contains two or more codewords having the same length l leads to parallel transitions between the states $S_{k-1} = n_1$ and $S_k = n_2$ in the VLC trellis. In addition, Fig. 2 shows that the packetized nature of our transmission scheme leads to a time-varying trellis

[1] In the following, we assume that perfect channel state information is available at the decoder.

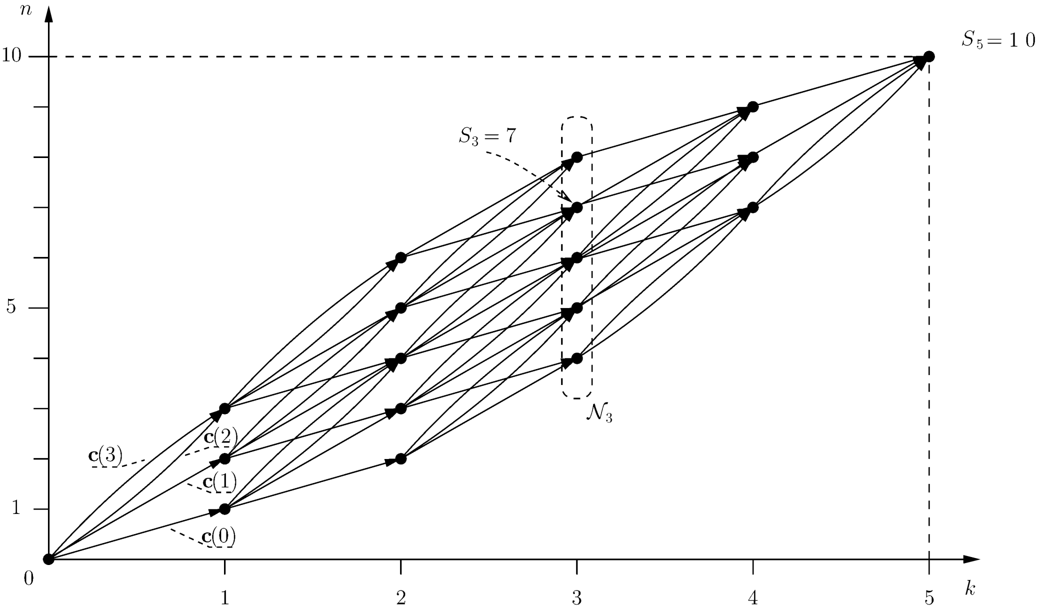


Fig. 2. Trellis representation for $K = 5, N = 10$, and $\mathcal{C} = \{\mathbf{c}^{(0)} = [1], \mathbf{c}^{(1)} = [0,1], \mathbf{c}^{(2)} = [0,0,0], \mathbf{c}^{(3)} = [0,0,1]\}$ [2].

with a diverging, a stationary, and a converging section due to the termination of the trellis after K source symbols and N codebits in the source packet.

B. Soft-Input APP Source Decoder

In the following, we are interested in soft-input soft-output VLC source decoding which generates, instead of an optimal sequence, reliability information for the source indexes $I_k = \lambda$, $\lambda \in \mathcal{I}$, in form of index-based APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$. These APPs can be obtained by the following soft-input decoding algorithm operating on the VLC trellis from above. As a new result, we also include the source correlation modeled by the index transition probabilities $P(I_k = \lambda \mid I_{k-1} = \mu)$ of the Markov source model into the decoding algorithm.

In the first step, the APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$ can be decomposed analogous to the BCJR algorithm [4] by using the Bayes theorem as

$$\begin{aligned} &P(I_k = \lambda \mid \hat{\mathbf{v}}) \\ &\quad = \sum_{n_2 \in \mathcal{N}_k} \sum_{n_1 \in \mathcal{N}_{k-1}} P(S_k = n_2, I_k = \lambda, S_{k-1} = n_1 \mid \hat{\mathbf{v}}) \\ &\quad = \frac{1}{C} \sum_{n_2 \in \mathcal{N}_k} \sum_{n_1 \in \mathcal{N}_{k-1}} \underbrace{p\left(\hat{\mathbf{v}}_{n_2+1}^{N} \mid S_k = n_2\right)}_{=:\beta_k(n_2)} \\ &\qquad \cdot \underbrace{p\left(\hat{\mathbf{v}}_{n_1+1}^{n_2}, I_k = \lambda, S_k = n_2 \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)}_{=:\gamma_k(\lambda, n_2, n_1)} \\ &\qquad \cdot \underbrace{p\left(S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)}_{=:\alpha_{k-1}(n_1)} \end{aligned} \tag{3}$$

where a subsequence from bit position a to b of the vector $\hat{\mathbf{v}}$ is denoted as $\hat{\mathbf{v}}_a^b = [\hat{v}_a, \hat{v}_{a+1}, \ldots, \hat{v}_b]$. The constant $C = p(\hat{\mathbf{v}})$ in (3) ensures that the $P(I_k = \lambda \mid \hat{\mathbf{v}})$ are true probabilities.

1) Derivation of the Forward Recursion: Let us consider the term $\alpha_{k-1}(n_1)$ from (3), which can be written as a *forward recursion* according to [4] as

$$\begin{aligned} \alpha_k(n_2) &= \sum_{n_1 \in \mathcal{N}_{k-1}} \sum_{\lambda=0}^{2^M-1} \gamma_k(\lambda, n_2, n_1) \cdot \alpha_{k-1}(n_1), \\ \alpha_0(0) &= 1 \end{aligned} \tag{4}$$

with

$$\begin{aligned} \gamma_k(\lambda, n_2, n_1) = &\underbrace{p\left(\hat{\mathbf{v}}_{n_1+1}^{n_2} \mid I_k = \lambda\right)}_{\text{channel term}} \\ &\cdot \underbrace{P\left(I_k = \lambda, S_k = n_2 \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)}_{\text{source correlation}}. \end{aligned} \tag{5}$$

The channel term in (5) can be calculated from the conditional channel pdf in (1) by utilizing the memoryless property of the channel as

$$p\left(\hat{\mathbf{v}}_{n_1+1}^{n_2} \mid I_k = \lambda\right) = \prod_{\ell=1}^{l(\mathbf{c}^{(\lambda)})} p\left(\hat{v}_{n_1+\ell} \mid c_\ell^{(\lambda)}\right). \tag{6}$$

Herein, $c_\ell^{(\lambda)}$ denotes the ℓth bit in the variable-length codeword $\mathbf{c}^{(\lambda)} = \left[c_1^{(\lambda)}, \ldots, c_\ell^{(\lambda)}, \ldots, c_{l(\mathbf{c}^{(\lambda)})}^{(\lambda)}\right]$. The second term on the right-hand side of (5) takes the source correlation into account and can be written as

$$\begin{aligned} &P\left(I_k = \lambda, S_k = n_2 \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right) \\ &\quad = \sum_{\mu=0}^{2^M-1} P(I_k = \lambda, S_k = n_2 \mid I_{k-1} = \mu, S_{k-1} = n_1) \\ &\qquad \cdot P\left(I_{k-1} = \mu \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right). \end{aligned} \tag{7}$$

The conditional probability $P(I_k = \lambda, S_k = n_2 \mid I_{k-1} = \mu, S_{k-1} = n_1)$ represents the transition probability of the stationary first-order Markov source adapted to the time-varying VLC trellis according to

$$P(I_k = \lambda, S_k = n_2 \mid I_{k-1} = \mu, S_{k-1} = n_1) = \frac{1}{C_1(n_1, \mu)} \cdot \begin{cases} P(I_k = \lambda \mid I_{k-1} = \mu), & \text{for } n_2 - n_1 = l(\mathbf{c}^{(\lambda)}) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

with the normalization factor

$$C_1(n_1, \mu) = \sum_{n_2' \in \mathcal{N}_k} \sum_{\lambda \in \mathcal{I}: l(\mathbf{c}^{(\lambda)}) = n_2' - n_1} P(I_k = \lambda \mid I_{k-1} = \mu). \tag{9}$$

The normalization in (9) takes into account that in both the stationary and the converging region of the VLC trellis, not all transitions from $S_{k-1} = n_1$ to $S_k = n_1 + l(\mathbf{c}^{(\lambda)})$ for $\lambda = 0, 1, \ldots, 2^M - 1$ are possible anymore. This can be observed, for instance, in the example from Fig. 2 for all states with $k = 4$.

Furthermore, the conditional probability $P\left(I_{k-1} = \mu \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)$ in (7) can be expressed with the quantities $\alpha_{k-2}(\cdot)$ and $\gamma_{k-1}(\cdot)$ for previous time instants, as we will show in the following. First, applying the Bayes theorem yields the decomposition

$$P\left(I_{k-1} = \mu \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right) = \frac{p\left(I_{k-1} = \mu, S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)}{\sum_{\mu'=0}^{2^M-1} p\left(I_{k-1} = \mu', S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)}. \tag{10}$$

Since the index $n_0 \in \mathcal{N}_{k-2}$ is uniquely determined by $n_0 = n_1 - l(\mathbf{c}^{(\mu)})$ for the hypotheses $I_{k-1} = \mu$ and $S_{k-1} = n_1$, the pdf $p\left(I_{k-1} = \mu, S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)$ from the right-hand side of (10) can be stated with the definitions of $\alpha_k(\cdot)$ and $\gamma_k(\cdot)$ in (3) as

$$\begin{aligned} p\left(I_{k-1} = \mu, S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right) &= p(I_{k-1} = \mu, S_{k-1} = n_1, \\ &\qquad S_{k-2} = n_0, \hat{\mathbf{v}}_1^{n_1}) \\ &= \gamma_{k-1}(\mu, n_1, n_0) \\ &\qquad \cdot \alpha_{k-2}(n_0). \end{aligned} \tag{11}$$

Inserting (11) into (10), we finally have

$$\begin{aligned} P\left(I_{k-1} = \mu \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right) &= \frac{\gamma_{k-1}(\mu, n_1, n_0) \cdot \alpha_{k-2}(n_0)}{\sum_{\mu'=0}^{2^M-1} \gamma_{k-1}(\mu', n_1, n_0) \cdot \alpha_{k-2}(n_0)} \\ \text{with } n_0 &= n_1 - l(\mathbf{c}^{(\mu)}). \end{aligned} \tag{12}$$

By combining (5), (7), and (12), we now obtain a relation for $\gamma_k(\lambda, n_2, n_1)$ according to

$$\begin{aligned} \gamma_k(\lambda, n_2, n_1) = &\underbrace{p\left(\hat{\mathbf{v}}_{n_1+1}^{n_2} \mid I_k = \lambda\right)}_{\text{channel term}} \\ &\cdot \sum_{\mu=0}^{2^M-1} \underbrace{P(I_k = \lambda, S_k = n_2 \mid I_{k-1} = \mu, S_{k-1} = n_1)}_{\text{transition probability on the trellis (Markov source)}} \\ &\cdot \underbrace{\frac{1}{C_2(n_1)} \gamma_{k-1}(\mu, n_1, n_0) \alpha_{k-2}(n_0)}_{\text{``forward'' APP } P\left(I_{k-1}=\mu \mid S_{k-1}=n_1, \hat{\mathbf{v}}_1^{n_1}\right)} \end{aligned} \tag{13}$$

with

$$C_2(n_1) = \sum_{\mu'=0}^{2^M-1} \gamma_{k-1}(\mu', n_1, n_0) \alpha_{k-2}(n_0)$$

$$\text{and } n_0 = n_1 - l(\mathbf{c}^{(\mu)}).$$

We can see from (13) that $\gamma_k(\lambda, n_2, n_1)$, which can be regarded as a weighting term for the trellis branches, also contains *a priori* knowledge about the correlation between adjacent source indexes and *index-based* reliability values $P\left(I_{k-1} = \mu \mid S_{k-1} = n_1, \hat{\mathbf{v}}_1^{n_1}\right)$ for time instant $k-1$ in addition to the bit reliability information from the channel. This can be regarded as an extension of the forward recursion (4) of the BCJR algorithm to variable-length encoded first-order Markov sources.

2) Derivation of the Backward Recursion: The term $\beta_k(n_2)$ in (3) is obtained with the so called *backward recursion* from [4]

$$\begin{aligned} \beta_k(n_1) &= \sum_{n_2 \in \mathcal{N}_{k+1}} \sum_{\lambda=0}^{2^M-1} \beta_{k+1}(n_2) \cdot \gamma'_{k+1}(\lambda, n_2, n_1), \\ \beta_K(N) &= 1 \end{aligned} \tag{14}$$

where due to the memoryless property of the transmission channel we have

$$\begin{aligned} \gamma'_{k+1}(\lambda, n_2, n_1) = &\underbrace{p\left(\hat{\mathbf{v}}_{n_1+1}^{n_2} \mid I_{k+1} = \lambda\right)}_{\text{channel term}} \\ &\cdot \underbrace{P(I_{k+1} = \lambda, S_{k+1} = n_2 \mid S_k = n_1)}_{\text{source probability distribution}}. \end{aligned} \tag{15}$$

In contrast to (13), the term $P(I_{k+1} = \lambda, S_{k+1} = n_2 \mid S_k = n_1)$ only contains information about the distribution of the source indexes due to the causal definition of the transition probabilities for the first-order Markov source. It can be specified analogous to (8) as

$$P(I_{k+1} = \lambda, S_{k+1} = n_2 \mid S_k = n_1) = \frac{1}{C_3(n_1)} \cdot \begin{cases} P(I_{k+1} = \lambda), & \text{for } n_2 - n_1 = l(\mathbf{c}^{(\lambda)}) \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

with

$$C_3(n_1) = \sum_{n_2' \in \mathcal{N}_{k+1}} \sum_{\lambda \in \mathcal{I}: l(\mathbf{c}^{(\lambda)}) = n_2' - n_1} P(I_k = \lambda) \tag{17}$$

where the normalization in (17) again takes the special structure of the VLC trellis into account.

We now have all quantities available in order to calculate the APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$ for $\lambda = 0, 1, \ldots, 2^M - 1$ according to (3), where compared to the classical BCJR algorithm the forward and backward recursion contain additional normalizations due to the time-varying structure of the VLC trellis. Furthermore, the forward recursion [(4) with (13)] now utilizes the first-order Markov source model for obtaining the α values. However, this also leads to an increased computational complexity for the forward recursion and therefore, suboptimal approaches for a complexity reduction will be discussed in Section III-C.

C. Optimal Estimation

The APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$ can now be used for optimally estimating the source symbols U_k in such a way that the value of the MS distortion or the symbol error rate (SER) is minimized in the reconstructed data packet $\hat{\boldsymbol{U}} = [\hat{U}_1, \hat{U}_2, \ldots, \hat{U}_K]$ at the decoder output. In the following, we therefore consider an MS and MAP estimation, respectively.

An MS estimation can be obtained by minimizing the average distortion over $\hat{U}_k$, i.e.,

$$E\left\{\left(U_k - \hat{U}_k\right)^2 \mid \hat{\mathbf{v}}\right\} = \min_{\hat{U}_k} \tag{18}$$

which directly corresponds to the maximization of the reconstruction SNR (RSNR). From (18), we obtain [12]

$$\hat{U}_k^{(\mathrm{MS})} = \sum_{\lambda=0}^{2^M-1} U_q(\lambda) \cdot P(I_k = \lambda \mid \hat{\mathbf{v}}) \tag{19}$$

with $U_q(\lambda)$ denoting the entry of the quantization table corresponding to the index $\lambda \in \mathcal{I}$. This estimator is especially useful for waveform-like signals where an SNR maximization is desired for the reconstructed signal.

The MAP estimation leads to classical symbol-by-symbol decoding and minimizes the symbol error probability, where the estimated values are given as

$$\begin{aligned} \hat{U}_k^{(\mathrm{MAP})} &= U_q(I_k = \lambda_{\mathrm{map}}) \\ P(I_k = \lambda_{\mathrm{map}} \mid \hat{\mathbf{v}}) &\geq P(I_k = \lambda \mid \hat{\mathbf{v}}), \\ &\text{for all } \lambda \in \mathcal{I} \backslash \lambda_{\mathrm{map}}. \end{aligned} \tag{20}$$

Note that using these two estimators in combination with true APPs generated by the soft-input VLC decoder from the previous section leads to a near-optimal decoding process in the MS or MAP sense.

IV. Complexity Reduction

In the following, we consider techniques for reducing the computational complexity of the proposed VLC APP source decoder.

A. Max-Log-Map Algorithm

The Max-Log-MAP algorithm assumes that the APP calculation in (3) is carried out completely in the logarithmic domain. Then, the following approximation [13], based on the Jacobian logarithm, is used for computing the logarithmic APPs $\ln(P(I_k = \lambda \mid \hat{\mathbf{v}}))$ and the corresponding logarithmic values of the α-, β-, and γ-terms in (3):

$$\ln\left(\sum_{\nu=1}^{q} e^{a_\nu}\right) \approx \max(a_1, a_2, \ldots, a_q). \tag{21}$$

For example, for the forward recursion in (4), the variables a_ν, $\nu = 1, \ldots, q$, correspond to $\ln(\gamma_k(\lambda, n_2, n_1)\alpha_{k-1}(n_1))$, where the summation over ν in (21) refers to both summations over n_1 and λ in (4). Clearly, by applying the right-hand side of (21) the computational complexity is reduced, since now a simple maximum operation replaces the computational costly calculation of the exponential terms and the final logarithm. An alternative interpretation is that by applying the approximation in (21), the number of trellis paths used in the decoding process is reduced [13]. Whereas the BCJR-based decoder considers all possible paths for obtaining the APPs, the Max-Log-MAP algorithm in our case only takes 2^M paths for each time instant k into account. Here, each path corresponds to a hypothesis $I_k = \lambda$, $\lambda \in \mathcal{I}$.

B. Reduction of the Number of Trellis States

Since the quantities $\alpha_k(n_2)$ and $\beta_k(n_2)$ can be regarded as reliability information for every state $n_2 \in \mathcal{N}_k$, a reduction of the number of trellis states and thus also a reduction of the computational complexity may be carried out depending on the actual values of both $\alpha_k(n_2)$ and $\beta_k(n_2)$. In the following, we consider two different strategies [14].

1. The first method (BCJR-T) utilizes a trellis state reduction based on a threshold, where in the first step, the forward recursion is carried out with (4). All states n_2 for which the corresponding logarithmic α values are smaller than a given threshold T, that is, if

   $$\ln(\alpha_k(n_2)) < T \text{ for all } k = 1, \ldots, K, \qquad n_2 \in \mathcal{N}_k$$

   holds, are removed from the list of trellis nodes for each k. Then, the backward recursion in (14) is calculated on this reduced trellis, where the trellis may be further truncated, if

   $$\ln(\beta_k(n_2)) < T \text{ for all } k = 1, \ldots, K, \qquad n_2 \in \mathcal{N}_k.$$

2. In the second approach (BCJR-Q), we do not specify a certain threshold, instead we only consider the Q largest values $\alpha_k(n_2)$ with

   $$\begin{aligned} \ln\left(\alpha_k(n_2^{(1)})\right) &\geq \ln\left(\alpha_k(n_2^{(2)})\right) \\ &\geq \cdots \geq \ln\left(\alpha_k(n_2^{(Q)})\right) \text{ for all} \\ n_2^{(\nu)} &\in \mathcal{N}_k, \quad \nu = 1, \ldots, Q \end{aligned}$$

   for every time instant k. The remaining α values are neglected and the corresponding states are removed from the VLC trellis. The resulting truncated trellis is then utilized for the backward recursion. However, compared to the BCJR-T method, the BCJR-Q approach has a higher complexity, since for every k the Q states with the highest reliability have to be determined by sorting the $\ln(\alpha_k(\cdot))$ values instead of a single comparison with a threshold value.

V. Iterative Decoding With Channel Codes

Due to the high sensitivity of the variable-length encoded bitstream-to-channel noise, using the residual source redundancy for error protection may not be enough in many cases. Therefore, we now assume that the interleaved output of the VLC encoder is protected by a systematic convolutional channel code prior to transmission, as it is depicted in Fig. 1. The interleaver is responsible for the decorrelation of the individual bits before the channel encoding is carried out. Since this encoding scheme is highly similar to a serially concatenated code, however, with the difference that the redundancy provided by the first channel encoder is replaced with the residual source redundancy, we can apply a similar iterative decoding strategy [15]. In this decoding scheme, the outer constituent decoder is replaced by the soft-input APP source decoder from the last section.

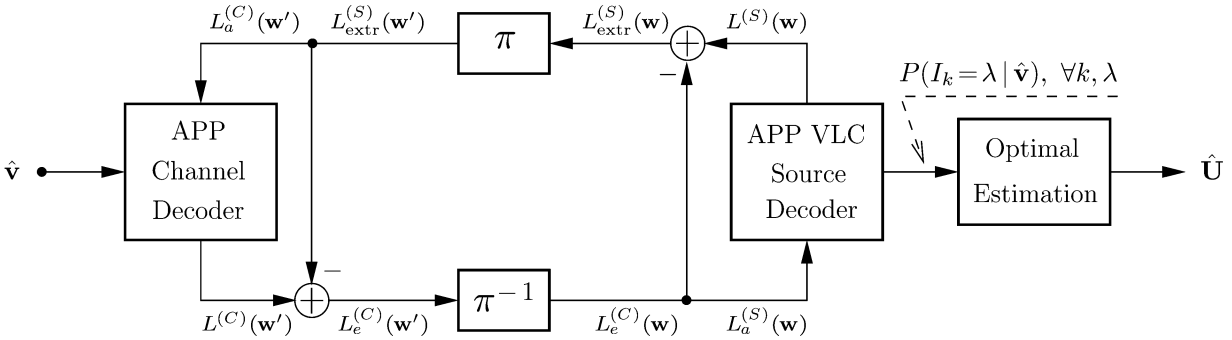


Fig. 3. Iterative joint source-channel decoder.

Fig. 3 shows the structure of the proposed decoder. The APP channel decoder calculates reliability information for the variable-length encoded index bits using the conditional L-values [16]

$$\begin{aligned} L^{(C)}(w'_n) &= \ln\left(\frac{P(w'_n = 0 \mid \hat{\mathbf{v}})}{P(w'_n = 1 \mid \hat{\mathbf{v}})}\right) \\ &= L_c a'_n \hat{w}'_n + L_a^{(C)}(w'_n) + L_{\text{extr}}^{(C)}(w'_n) \end{aligned} \tag{22}$$

for $n = 1, 2, \ldots, N$. $L_c {a'}_n \hat{w}'_n$ is defined analogous to (2) for the interleaved information bits w'_n and the corresponding amplitude variation factors a'_n. The term $L_a^{(C)}(w'_n)$ also refers to w'_n and denotes the *a priori* information which is available to the APP channel decoder and which is set to zero at the beginning of the decoding process. In (22), $L_{\text{extr}}^{(C)}(w'_n)$ refers to the extrinsic information [16]. After subtraction of the *a priori* term, we obtain the L-values $L_e^{(C)}(w'_n) = L_c {a'}_n \hat{w}'_n + L_{\text{extr}}^{(C)}(w'_n)$, which are used as *a priori* information $L_a^{(S)}(w_n) = L_e^{(C)}(w_n)$ for the VLC source decoder after deinterleaving. Note that this *a priori* term contains both the soft information at the channel output and the extrinsic information of the APP channel decoder for the information bit w_n. It can be converted back to an APP

$$\begin{aligned} P_e^{(C)}(w_n \in \{0,1\} \mid \hat{\mathbf{v}}) &= \frac{1}{C'} \cdot p(\hat{w}_n \mid w_n) \cdot P_{\text{extr}}^{(C)}(w_n \mid \hat{\mathbf{v}}) \\ &= \begin{cases} \frac{e^{L_e^{(C)}(w_n)}}{1+e^{L_e^{(c)}}(w_n)}, & \text{for } w_n = 0 \\ \frac{1}{1+e^{L_e^{(c)}}(w_n)}, & \text{for } w_n = 1 \end{cases} \end{aligned} \tag{23}$$

for each information bit, where C' denotes a normalization constant. When we now assume that all bits are independent, the index-based APP for a certain $\lambda \in \mathcal{I}$ can be obtained by multiplication of the individual bit APPs in (23) with C'' representing another normalization constant

$$\begin{aligned} P_a^{(S)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}}) &= P_e^{(C)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}}) \\ &= \prod_{\ell=1}^{l(\mathbf{c}^{(\lambda_{n_1})})} P_e^{(C)}\left(w_{n_1+\ell} = c_\ell^{(\lambda_{n_1})} \mid \hat{\mathbf{v}}\right) \\ &= \frac{1}{C''} \cdot p\left(\hat{\mathbf{w}}_{n_1+1}^{n_2} \mid I_k = \lambda_{n_1}\right) \\ &\quad \cdot P_{\text{extr}}^{(C)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}}). \end{aligned} \tag{24}$$

Since the index-based APPs $P_a^{(S)}(\cdot)$ and $P_e^{(S)}(\cdot)$ are calculated for all possible bit positions in the binary sequence $\mathbf{w}$, the source hypotheses now depend on the bit position n_1 and are denoted as λ_{n_1} for clarity.

The extrinsic information $P_{\text{extr}}^{(C)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}})$ in (24) contains extra information about the index I_k from the channel decoding process. This can be used in order to improve the reliability of the APPs at the output of the APP VLC source decoder by inserting $P_{\text{extr}}^{(C)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}})/C''$ as an extra weighting term in (5), where due to the use of systematic channel codes the received bit vector $\hat{\mathbf{v}}$ in (5) now has to be replaced with the corresponding vector $\hat{\mathbf{w}}$ for the information bits

$$\begin{aligned} &\gamma_k(\lambda_{n_1}, n_2, n_1) \\ &\quad = \underbrace{\frac{1}{C''} P_{\text{extr}}^{(C)}\left(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}}\right) p\left(\hat{\mathbf{w}}_{n_1+1}^{n_2} \mid I_k = \lambda_{n_1}\right)}_{P_a^{(S)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}})} \\ &\qquad \cdot P\left(I_k = \lambda_{n_1}, S_k = n_2 \mid S_{k-1} = n_1, \hat{\mathbf{w}}_1^{n_1}\right). \end{aligned} \tag{25}$$

The term in braces can be identified as $P_a^{(S)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}}) = P_e^{(C)}(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}})$ from (24). Likewise, we have for the γ'-term of the backward recursion in (15)

$$\begin{aligned} \gamma'_k(\lambda_{n_1}, n_2, n_1) = P_a^{(S)}&(I_k = \lambda_{n_1} \mid \hat{\mathbf{v}}) \\ &\cdot P(I_k = \lambda_{n_1}, S_k = n_2 \mid S_{k-1} = n_1). \end{aligned}$$

The APP source decoder now issues index-based APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$, where the bit-based APPs for the information bits w_n can be calculated according to (26), seen at the bottom of the page, where $\mathcal{K}_n$ denotes the set of all possible states for bit position n, and the constant C is the same as in (3). The conditional *a posteriori* L-values $L^{(S)}(w_n)$ are then obtained with

$$L^{(S)}(w_n) = \ln\left(\frac{P^{(S)}(w_n = 0 \mid \hat{\mathbf{v}})}{P^{(S)}(w_n = 1 \mid \hat{\mathbf{v}})}\right). \tag{27}$$

$$\begin{aligned} P^{(S)}(w_n = i \mid \hat{\mathbf{v}}) &= \sum_{\substack{(k, n_1, n_2, \lambda_{n_1}) \in (\mathcal{K}_n \times \mathcal{N}_{k-1} \times \mathcal{N}_k \times \mathcal{I}): \\ w_n = i}} P(S_{k-1} = n_1, I_k = \lambda_{n_1}, S_k = n_2 \mid \hat{\mathbf{v}}) \\ &= \frac{1}{C} \sum_{\substack{(k, n_1, n_2, \lambda_{n_1}) \in (\mathcal{K}_n \times \mathcal{N}_{k-1} \times \mathcal{N}_k \times \mathcal{I}): \\ w_n = i}} \beta_k(n_2) \gamma_k(\lambda_{n_1}, n_2, n_1) \alpha_{k-1}(n_1) \end{aligned} \tag{26}$$

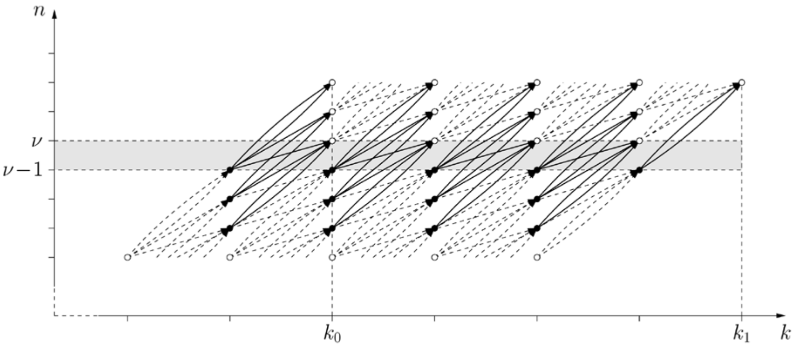


Fig. 4. On the calculation of the bit-based APP $P(w_\nu = i \mid \hat{\mathbf{v}})$.

Note that the summation in (26) is carried out over all trellis branches which correspond to bit $w_n = i \in \{0,1\}$ at bit position n. Fig. 4 shows an example for bit w_ν at bit position ν with $\mathcal{K}_\nu = \{k_0, k_0+1, \ldots, k_1\}$, where the corresponding branches included in the summation are marked with solid lines and the corresponding states S_{k-1} with black nodes, respectively.

By subtracting the source *a priori* information $L_a^{(S)}(w_n)$ from the L-values $L^{(S)}(w_n)$ in (27), we finally obtain the extrinsic information $L_{\text{extr}}^{(S)}(w_n)$, which is used after interleaving as *a priori* information $L_a^{(C)}(w'_n) = L_{\text{extr}}^{(S)}(w'_n)$ in the next channel decoding round. Note that only after the first half-iteration "exact" APPs $P(I_k = \lambda \mid \hat{\mathbf{v}})$ are obtained at the output of the VLC source decoder [16]. In subsequent iterations, only APP "approximations" are available since correlations between the *a priori* L-values $L_a^{(C)}(w'_n)$ and the extrinsic information $L_{\text{extr}}^{(C)}(w'_n)$ in (22) emerge.

VI. Simulation Results

In order to assess both the performance of the soft-input APP source decoding approach and the iterative source-channel decoding technique, including their complexity-reduced versions, simulations were carried out for AWGN and Rayleigh transmission channels. In all simulations, it was assumed that perfect channel state information is available for the channel signal-to-noise ratio (SNR) and the amplitude variation factors a_m. The source redundancy was modeled by an AR(1) source process with correlation coefficient $a = 0.9$ quantized with M bits by a uniform scalar quantizer. Besides, we used packets of length $K = 100$ over 50 different realizations of the source signal, where each packet was interleaved and transmitted independently. The source transition probabilities $P(I_k = \lambda \mid I_{k-1} = \mu)$ were obtained from a training set of 500 source realizations. Each of the 50 source packets was averaged over 100 channel realizations for a given channel noise variance. We furthermore assumed that the sensitive parameters N and K are protected by a strong channel code and thus are transmitted without errors to the decoder.

A. APP Source Decoder

In this section, we consider the performance of the full-state APP decoding approach from Section III and its complexity-reduced versions from Section IV. In addition to the simulation parameters mentioned above, a standard Huffman code with $M = 4$ and an AWGN channel was used.

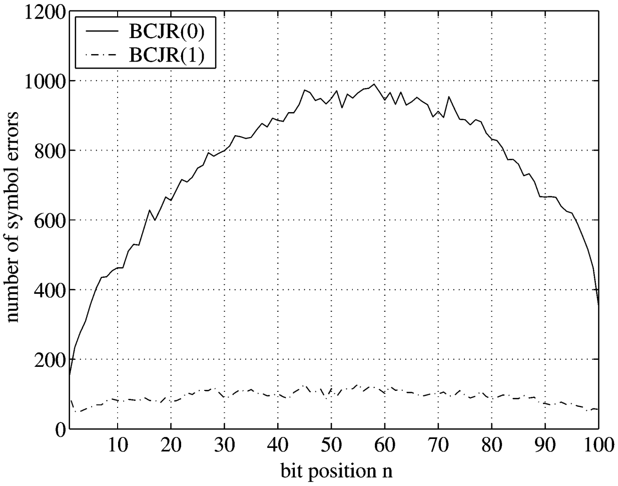


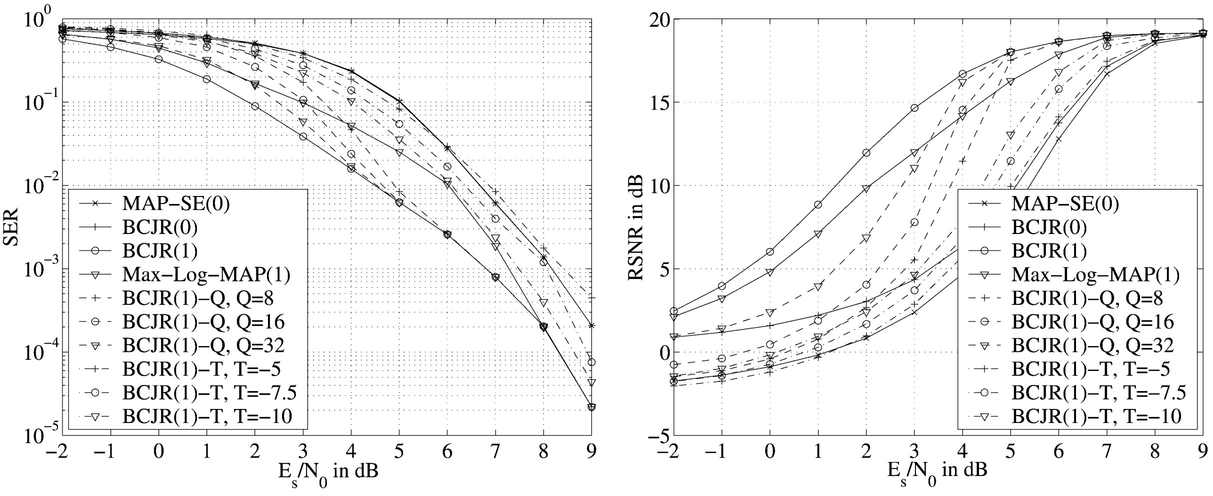
Fig. 5. Occurrence of symbol errors in data packets (AWGN channel with $E_s/N_0 = 4$ dB, one single channel realization, $K = 100$ for 5000 different realizations of the source signal).

First, we illustrate the influence of the source statistics on the distribution of symbol errors in a data packet. Fig. 5 shows a profile of occurring symbol errors for the AWGN channel with $E_s/N_0 = 4$ dB, where 5000 source realizations were simulated with the full-state APP VLC decoder. The BCJR(0) approach exploits the probability distribution of the source indexes corresponding to a zeroth-order Markov source and is essentially the method presented in [2] and [3], whereas the BCJR(1) decoder additionally considers the first-order Markov property of the source. It can be seen from Fig. 5 that for the BCJR(0) decoder the number of symbol errors increases to the middle of the packet and, due to the merging VLC trellis, decreases again, when we proceed further to the end. However, for the BCJR(1) decoder, taking the index correlation into account, a flat distribution of the symbol errors is obtained. This is due to the fact that by additionally exploiting the first-order Markov property of the source, the mean deviation of all used paths in the VLC trellis from the optimal one is reduced, leading to a smaller number of symbol errors.

We now address the complexity-reduced APP VLC decoding approaches in Section IV and estimate the complexity by comparing the required number of trellis states. Table I shows the

TABLE I
AVERAGE NUMBER OF TRELLIS STATES AT TIME INSTANT k IN THE STATIONARY TRELLIS SECTION FOR THE BCJR-T AND THE FULL-STATE VLC BCJR DECODER (AWGN CHANNEL, $K = 100$)

	$E_s/N_0 = -2\,\mathrm{dB}$		$E_s/N_0 = 2\,\mathrm{dB}$		$E_s/N_0 = 6\,\mathrm{dB}$	
	forward rec.	backw. rec.	forward rec.	backw. rec.	forward rec.	backw. rec.
BCJR-T, $T=-20$	22	19	22	15	4	2
BCJR-T, $T=-10$	11	9	7	6	1	1
BCJR-T, $T=-7.5$	6	5	4	2	1	1
BCJR-T, $T=-5$	3	2	2	2	1	1
full-state VLC BCJR	49	49	49	49	49	49



(a)

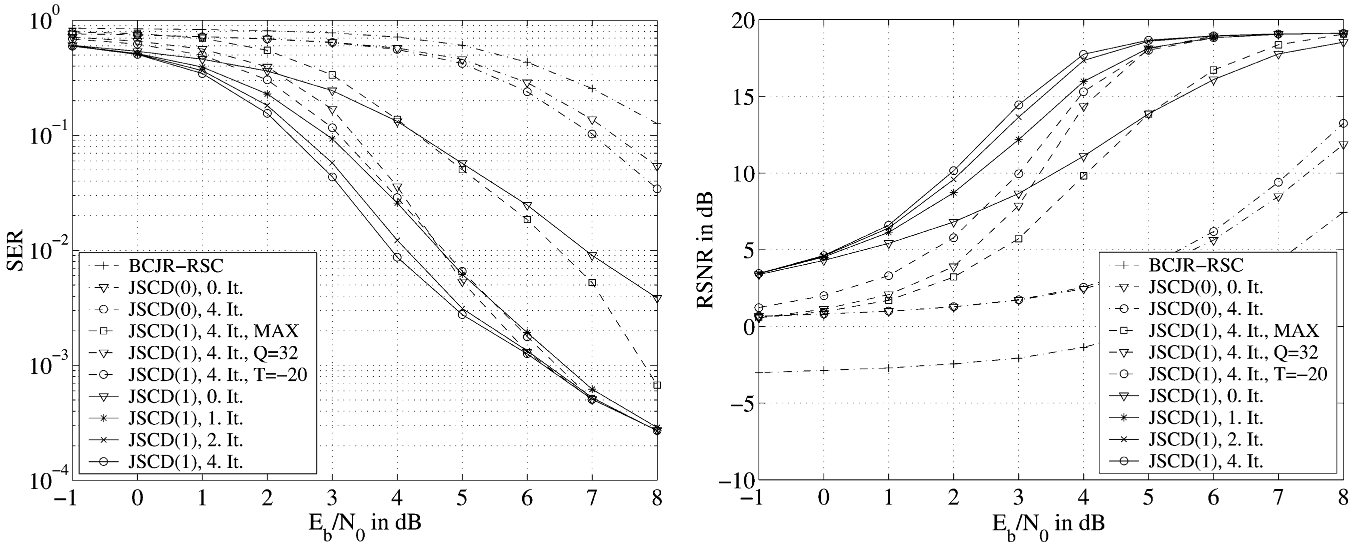
(b)

Fig. 6. Comparison of efficient VLC source decoding approaches, transmission over an AWGN channel [$K = 100$ for 50 different realizations of the source signal, averaged over 100 AWGN channel realizations; $M = 4$, AR(1) process with source correlation $a = 0.9$]. (a) MAP estimation and SER. (b) MS estimation [except for MAP-SE(0)] and reconstruction SNR (RSNR).

average number of trellis states at time index k in the stationary trellis section for the BCJR-T and the full-state BCJR-based APP decoder from Section III, where the number of states in the forward recursion (4) and backward recursion (14), resp., is considered separately. The BCJR-Q method is not shown in Table I, since it has a fixed complexity with always Q states per time instant in the stationary trellis section. It can be observed from Table I that the average number of states decreases with increasing E_s/N_0. This is due to the fact that a decreasing channel noise variance increases the bit reliability at the channel output, such that for a larger number of states the corresponding $\ln\alpha$- and $\ln\beta$-values are below the given threshold T.

Finally, the error-correction performance of the presented APP VLC source decoding approaches is addressed in Fig. 6. Here, the Max-Log-MAP algorithm as well as the BCJR-T and BCJR-Q techniques are only applied to the BCJR(1) decoder and thus are denoted with Max-Log-MAP(1), BCJR(1)-T, and BCJR(1)-Q, respectively. Fig. 6(a) refers to a MAP estimation, where the SER is plotted over the channel parameter E_s/N_0. The result for a MAP sequence estimation for the zeroth-order Markov case is denoted with MAP-SE(0), where its performance is almost identical to that of the BCJR(0) technique. We can also observe from Fig. 6(a) that all BCJR(1)-Q methods reach the performance of the full-state decoder for larger channel SNRs. Due to the high reliability of the received bits in this channel SNR region, only certain states contribute to the APP calculation in the BCJR algorithm. The other ones may be neglected and thus, may be removed from the trellis without any loss in performance. In our simulations, we have observed that the Max-Log-MAP algorithm has approximately the same complexity as the BCJR(1)-Q approach for $Q = 16$. The direct comparison in Fig. 6(a) shows that the Max-Log-MAP algorithm yields a lower SER only for highly distorted channels with $E_s/N_0 \leq 3$ dB. Due to the smaller number of states, the BCJR(1)-T approaches perform worse compared to all other approaches. For $T = -5$, the results are comparable to the BCJR(0) technique, however, the complexity is highly reduced as we can observe from Table I.

Fig. 6(b) shows the results for a MS estimation of the reconstructed source signal, where now the RSNR is used as error measure. Again, it can be observed that for highly corrupted channels with $E_s/N_0 \leq 3$ dB the Max-Log-MAP algorithm applied to the BCJR(1) technique represents the best complexity-reduced decoder in terms of RSNR.



(a) (b)

Fig. 7. Simulation results for the iterative source-channel decoder, transmission over a fully interleaved flat Rayleigh channel ($K = 100$ for 50 different realizations of the source signal, averaged over 100 channel realizations; $M = 4$, AR(1) process with source correlation $a = 0.9$). (a) MAP estimation and SER. (b) MS estimation (except for BCJR-RSC) and reconstruction SNR (RSNR).

B. Iterative Source-Channel Decoder

Next, we simulated the iterative source-channel decoder from Section V for the fully interleaved BPSK-modulated flat Rayleigh channel. After VLC encoding, the resulting bitstream was interleaved using an S-random interleaver [17]. The channel code used in the simulations was a terminated rate $R = 3/4$ recursive systematic convolutional (RSC) code, which was obtained from an optimal memory-4 rate-1/2 mother code with free distance $d_f = 7$ and the generator matrix [18]

$$G(D) = \begin{bmatrix} 1 & \dfrac{1+D^1+D^2+D^4}{1+D^3+D^4} \end{bmatrix}.$$

Puncturing was carried out with the pattern $[1,1,1;1,0,0]$ which was selected via a computer search. All other simulation parameters and constraints were chosen as stated at the beginning of this section.

Fig. 7 shows the simulation results for a Huffman code with $M = 4$, where the channel parameter is now E_b/N_0 with $E_b = E_s/R$. In the JSCD(0) technique, the APP VLC source decoder in Fig. 3 is based on the BCJR(0) approach from Section VI-A, whereas the source decoder for the JSCD(1) method also exploits the index correlations of the first-order Markov source. Protecting the variable-length encoded bitstream only with the rate-3/4 RSC channel code without considering any source statistics leads to the BCJR-RSC method, which employs the BCJR algorithm as channel decoder followed by a hard decision decoding for the variable-length encoded bitstream. Furthermore, efficient realizations by means of the BCJR(1)-T and BCJR(1)-Q approaches with the parameters $T = -20$ and $Q = 32$, respectively, are also included in the comparison. The "JSCD(1), MAX" technique uses the Max-Log-MAP algorithm in both channel and source decoder, whereas in all other approaches the channel decoder employs the full-state BCJR.

As we can see from the result of the MAP estimation in Fig. 7(a), for $E_b/N_0 \geq 5$ dB and the JSCD(1) approach, we achieve a SER being at least one order of magnitude lower than that for the JSCD(0) technique, even if only the first half-iteration ("zeroth" iteration) is carried out. However, increasing the number of iterations over two, a further gain is only observed in the "waterfall" region of the SNR-SER relation in Fig. 7(a). When the less complex BCJR(1)-T or BCJR(1)-Q source decoders are employed as outer constituent decoders in the iterative decoding scheme from the last section, we almost reach the performance of the full-state iterative decoder for moderately distorted channels. However, for worse channels, a performance degradation can be observed. Note that the performance of the BCJR-RSC approach is almost identical to APP channel decoding followed by a maximum-likelihood sequence estimation for the Huffman-encoded sequence. This is due to the fact that a Huffman-encoded binary sequence still corresponds to a valid code sequence after a single bit error *except* the final bits in the sequence which may not be valid Huffman codewords.

Fig. 7(b) shows the results for the MS estimation. We can see that with full-state iterative joint source-channel decoding we achieve almost clear-channel quality for $E_b/N_0 > 4$ dB. This verifies the good performance of the proposed transmission system.

Finally, a more practical example is considered, where the one-dimensionally scanned lowpass subband image obtained from a three-level wavelet decomposition of the "Lena" image is transmitted over a Rayleigh channel. Here, we used a symmetrical reversible VLC (RVLC) [19], which has additional inherent redundancy due to the symmetry requirements for the codewords. The image data was uniformly quantized with $M =$

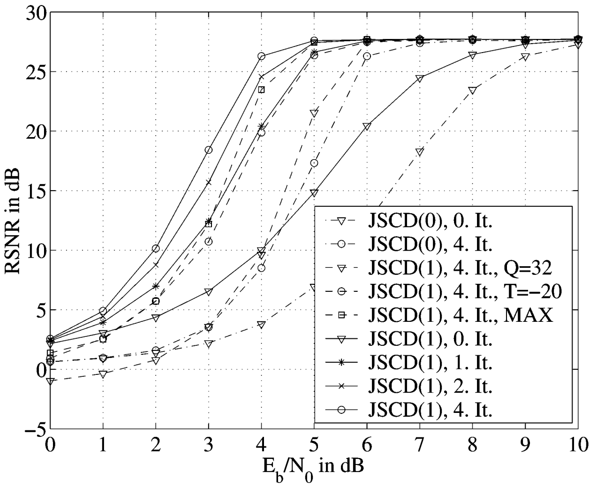


Fig. 8. MS estimation and reconstruction SNR (RSNR) for the lowpass subband signal of the wavelet-decomposed "Lena" image encoded with an RVLC (fully interleaved flat Rayleigh channel; $K = 64$, averaged over 100 channel realizations; $M = 5$).

5 bits and packetized with a block length of $K = 64$ corresponding to one row of the lowpass subband image. From the results for a MS estimation in Fig. 8, it can be seen that most JSCD(1) approaches lead to clear-channel quality for $E_b/N_0 >$ 5 dB. When considering moderate channel SNRs, the Max-Log-MAP algorithm applied to both source and channel decoder performs only slightly worse or even almost identically compared to the full-state iterative approach. The reason is that the additional redundancy of the RVLC improves the reliability of the states associated with the optimal trellis path in the source decoder. By exchanging extrinsic information, this reliability increase is also transferred to the channel decoder. Therefore, applying the Max-Log-MAP algorithm, where a lower number of trellis paths compared to the full-state approach is considered, only leads to a small performance degradation for moderately distorted transmission channels.

VII. Conclusion

We have shown that by starting from the classical BCJR algorithm, a soft-input APP decoding technique for packetized variable-length encoded source data can be derived by modification of the BCJR forward recursion and adaptation to the nonstationary VLC trellis. As a new result, the proposed APP VLC decoder considers residual source index correlations modeled by a first-order Markov model as *a priori* information and leads to a near-optimal decoding process, when a subsequent MS or MAP estimation is carried out. Besides the transition probabilities of the Markov model and some channel information, also the number of source symbols and the number of bits in the packet are used as *a priori* knowledge in the decoding process. A complexity reduction of the proposed APP source decoding approach may be carried out by reducing the number of trellis states or by using the Max-Log-MAP algorithm. For moderately distorted transmission channels, we obtain only a minor performance degradation, however, for stronger channel distortions a state reduction always leads to a decrease in performance.

When the variable-length encoded bitstream is additionally protected by channel codes, an iterative decoder consisting of an APP decoder for the channel code and the proposed APP VLC source decoder can be derived in the same manner as for serially concatenated codes. The simulation results show that for a variable-length encoded autoregressive source transmitted over a flat Rayleigh channel and a MS estimation of the reconstructed source symbols the full-state iterative source-channel decoder leads to almost clear-channel quality for an E_b/N_0 being larger than 4 dB. This result also holds for real image subband data being encoded with a symmetrical reversible VLC.

References

[1] J. Wen and J. D. Villasenor, "Utilizing soft information in decoding of variable length codes," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 1999, pp. 131–139.

[2] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in *Proc. 3rd ITG Conf. Source Channel Coding*, Munich, Germany, Jan. 2000, pp. 111–116.

[3] ——, "Iterative source/channel-decoding using reversible variable length codes," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2000, pp. 93–102.

[4] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. IT-20, no. 3, pp. 287–294, Mar. 1974.

[5] M. Bystrom, S. Kaiser, and A. Kopansky, "Soft source decoding with applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 10, pp. 1108–1120, Oct. 2001.

[6] K. Sayood, H. H. Otu, and N. Demir, "Joint source/channel coding for variable length codes," *IEEE Trans. Commun.*, vol. 48, no. 5, pp. 787–794, May 2000.

[7] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," *IEEE Trans. Commun.*, vol. 48, no. 1, pp. 1–6, Jan. 2000.

[8] E. Fabre, A. Guyader, and C. Guillemot, "Joint source-channel turbo decoding of VLC-coded Markov sources," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Salt Lake City, UT, May 2001, pp. 2657–2660.

[9] M. Park and D. J. Miller, "Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Phoenix, AZ, May 1999, pp. 2451–2454.

[10] E. Fabre, A. Guyader, C. Guillemot, and M. Robert, "Joint source-channel turbo decoding of entropy-coded sources," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1680–1696, Sep. 2001.

[11] J. Kliewer and R. Thobaben, "Combining FEC and optimal soft-input source decoding for the reliable transmission of correlated variable-length encoded signals," in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Apr. 2002.

[12] T. Fingscheid and P. Vary, "Robust speech decoding: A universal approach to bit error concealment," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Munich, Germany, Apr. 1997, pp. 1667–1670.

[13] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun.*, Seattle, WA, Jun. 1995, pp. 1009–1013.

[14] V. Franz and J. B. Anderson, "Concatenated decoding with a reduced-search BCJR algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, no. 2, pp. 186–195, Feb. 1998.

[15] N. Goertz, "On the iterative approximation of optimal joint source-channel decoding," *IEEE J. Select. Areas Commun*, vol. 19, no. 9, pp. 1662–1670, Sep. 2001.

[16] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.

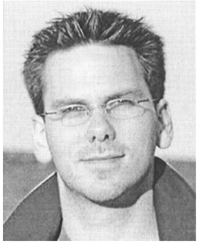
[17] D. Divsalar and F. Pollera, "Turbo codes for PCS applications," in *Proc. IEEE Int. Conf. Commun.*, Seattle, WA, Jun. 1995, pp. 54–59.

[18] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[19] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 158–162, Feb./Mar./Apr. 1995.

Jörg Kliewer (S'97–M'99–SM'04) received the Dipl.-Ing. degree in electrical engineering from the Hamburg University of Technology, Hamburg, Germany, in 1993 and the Dr.-Ing. degree (Ph.D.) in electrical engineering from the University of Kiel, Kiel, Germany, in 1998.

From 1993 to 1998, he was a Research Assistant at the Institute for Circuits and Systems, University of Kiel. In 1996, he spent five months at the Image Processing Lab, University of California, Santa Barbara, as a Visiting Researcher. Since 1999, he has been with the Faculty of Engineering, University of Kiel where he is a Senior Researcher and Lecturer. His current research interests are in the area of signal processing for communications applications and include joint source and channel coding, error correcting codes, and network coding.



Ragnar Thobaben was born in Hamburg, Germany, in 1977. He received the Dipl.-Ing. degree in electrical engineering from the University of Kiel, Kiel, Germany, in 2001.

Since October 2001, he has been a Research Assistant with the Institute for Circuits and Systems Theory, University of Kiel. His research interests are in joint source and channel coding and iterative source channel decoding of variable-length-encoded source signals.

Mr. Thobaben received the Prof. Dr. Werner Petersen Award for his diploma thesis and the Best Diploma Award of the Faculty of Engineering, University of Kiel, in 2001.