

Bit-Based Joint Source-Channel Decoding of Huffman Encoded Markov Multiple Sources

Weiwei Xiang

Department of communication engineering, Zhengzhou Information Science and Technology Institute,
Zhengzhou, P. R. China

Southwest Institute of Electron & Telecom. Techn., Shanghai, P. R. China

Email: xiang_ww423@yahoo.com.cn

Peizhong Lu

Southwest Institute of Electron & Telecom. Techn, Shanghai, P. R. China

Email: pzlu@fudan.edu.cn

Abstract—Multimedia transmission over time-varying channels such as wireless channels has recently motivated the research on the joint source-channel technique. In this paper, we present a method for joint source-channel soft decision decoding of Huffman encoded multiple sources. By exploiting the *a priori* bit probabilities in multiple sources, the decoding performance is greatly improved. Compared with the single source decoding scheme addressed by Marion Jeanne, the proposed technique is more practical in wideband wireless communications. Simulation results show our new method obtains substantial improvements with a minor increasing of complexity. For two sources, the gain in SNR is around 1.5dB by using convolutional codes when symbol-error rate (SER) reaches 10^{-2} and around 2dB by using Turbo codes.

Index Terms—Joint source-channel decoding (JSCD), variable-length codes (VLCs), Convolutional codes (CCs), Turbo codes (TCs)

I. INTRODUCTION

Shannon's source and channel separation theorem demonstrates that optimum performance can be achieved by designing source and channel code independently with infinite block length and unlimited complexity [1]. However, it doesn't consider the time-varying channels such as wireless channels. With the development of the modern communication systems, the separation style can no longer meet the needs of transmitting information efficiently and reliably. Therefore, the joint source-channel technique has recently gained considerable attentions, since the residual redundancy in the source and channel can be sufficiently explored to improve the decoding performance. In this paper, we focus on the joint source-channel decoding (JSCD) of variable-length encoded Markov multiple sources.

The first attempt in the JSCD focus on the fixed-length codes[2]-[5]. Then, the high compression efficiency of

variable-length codes (VLCs), such as Huffman codes, arithmetic codes, has motivated the consideration of JSCD on variable-length coded streams. However, VLCs have a major weakness that they are more sensitive to error propagation. This noteworthy problem has first been addressed in [6], where the entropy encoded bit stream was modeled as a semi-Markov process. The resulting dependency structures are well adapted for Maximum a posteriori (MAP) estimation, making use of the soft-input soft-output Viterbi algorithm [7] or the BCJR algorithm [8]. In [9], a joint VLC decoding method relying on the residual redundancy of the Markov source has been considered to reduce this effect. By combining the separate models for the Markov source, the source coder and the channel coder, a global stochastic automaton model of the transmitted bit stream is derived to perform a MAP decoding with the Viterbi algorithm. The approach provides an optimal joint decoding scheme but remains intractable for realistic applications because of the state explosion phenomenon.

Shannon mentioned in his 1948 paper that "any redundancy in the source will usually help if it is utilized at the receiving point." For reasons of delay and complexity, many source coding schemes still contain redundancy, which can be used at the channel decoder to improve the performance. A source-controlled channel decoding method is proposed in [10], where the source statistics *a priori* information is directly used at the channel decoder. Since it can be applied to the current communication system without any change for the transmission scheme, this method is more valuable in the practical applications. This approach has been attempted for fixed-length codes in convolution channel encoder in [10] and turbo channel encoder in [11] [12]. Recently, the method was extended to VLC encoded convolutional channel encoder and turbo channel encoder in [13] and [14].

In [14], Marion Jeanne presents a method by which the remaining redundancy in a single sequence encoded by a Huffman encoder is well used to improve the performance at the channel decoder.

The paper is supported by the National Natural Science Foundation of China under Grant No. 60673082, and Special Funds of Authors of Excellent Doctoral Dissertation in China under Grant No. 200084.

In this paper, we generalize M. Jeanne’s method to the multiple Huffman encoded sources. Our generalization is nontrivial because we overcome the following difficult problem: after the multiplexing of Huffman encoded multiple sources, the source encoding paths have complex relationships with the branches of trellis used by channel decoding, which results in intractable to obtain *a priori* bit probabilities of the code tree.

In the process of our new decoding, not only the influence of the previous state to the current state, but also the one of the current state to the future state should be considered. Since we efficiently use the *a priori* source information, our novel approach obtains the gain in SNR around 1.5dB and 2dB when the sources are protected by a CC and TC respectively, and channel output signal is at $SER=10^{-2}$. Though the performances of our method will decrease when the number of source branches is increased, there is still a gain around 0.5 dB with a CC and 0.7dB with a TC when the number of source branches is 16.

Since the transmission of multiple sources is a popular technique used in wideband wireless digital communications, the method proposed in this paper is more useful in practical applications.

The rest of the paper is organized as follows. In Section II we first discuss the computation technique to obtain the bit probabilities from the source symbol probabilities, and then we present how to use the *a priori* bit probabilities at the channel decoder. In section III, we extend the JSCD method from single source to multiple sources for CC, and in section IV for TC. Simulation results for multiple branches of Markov sources are presented.

II. A PRIORI SOURCE INFORMATION

In the decoding process of the VLC encoded sources, the probability distribution of source symbols is assumed to be known. This knowledge of the *a priori* information can be used to improve the decoding performance [15]. When using the convolutional code, binary trellis decoding method is generally adopted. So we first need to derive the source probabilities at a bit level.

Huffman encoding is realized by building a binary tree, as showed in Fig. 1. Each bit in a codeword associates with an edge of the tree. The probability for each edge is just the probability of all codewords passing this edge. Then probability of each bit in a codeword is defined as the probability of the corresponding edge. Since the probability distribution of the source is known, we can obtain the “bit level” *a priori* information, namely the probability of a bit in codewords. The computation of the bit probabilities from the known codeword probabilities can be described as follows.

We denote by $\{c^1, c^2, \dots, c^C\}$ the set of source symbols with C being the size of the symbol set and $p(c^k)$ being the probability of symbol c^k . Each symbol is encoded as a codeword $c^k=[c^k(1), \dots, c^k(l_k)]$, where $c^k(i) \in \{0,1\}$ and l_k denotes the length of the codeword c^k , $1 \leq k \leq C$. For example, in Fig.1, $c^1 = [0,0]$ and l_1 is equal to 2. In the

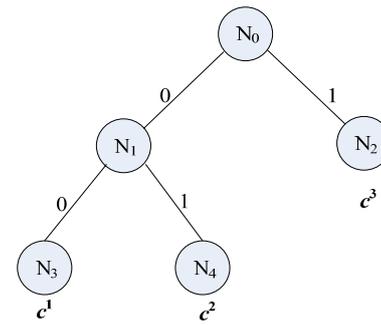


Fig.1 VLC tree representation

process of Huffman encoding, all nodes are numbered from the root node to the leaf nodes in sequence. All nodes in the tree form a set denoted by $N = \{N_0, N_1, \dots, N_n\}$, where $n = 2C - 1$ obviously. Each node belongs to at least one codeword. For instance, N_3 only belongs to codeword c^1 , while N_1 belongs to both c^1 and c^2 . The root node N_0 belongs to all codewords obviously. We define $N_i \in c^k$ if node N_i belongs to the codeword c^k .

Let $S_i = \{c^k \mid N_i \in c^k, 1 \leq k \leq C\}$, $0 \leq i \leq n$ be the set of all codewords with path passing through the node N_i . The probability of each node is the accumulated probability of all the codewords that pass through it, namely

$$P_N(N_i) = \sum_{c^k \in S_i} p(c^k). \quad (1)$$

Obviously, the node probability of leaf node is the one of that codeword it belongs to, e.g., $P_N(N_3) = p(c^1)$. If a node has two child nodes, the corresponding node probability is the sum of the child nodes probabilities, for instance, $P_N(N_1) = P_N(N_3) + P_N(N_4)$.

The probability of an edge with the start node N_i and the end node N_j is defined as follows

$$P_b(N_i \rightarrow N_j) = \frac{P_N(N_i)}{P_N(N_j)} = \frac{\sum_{c^k \in S_i} p(c^k)}{\sum_{c^k \in S_j} p(c^k)}. \quad (2)$$

Thus $P_b(N_i \rightarrow N_j)$ is the bit probability corresponding to the edge $N_i \rightarrow N_j$, which can be computed from the codeword probabilities.

There are two trees, the binary trellis in channel decoding and the tree of source codewords. In a JSCD algorithm, the decoder may correspond to a branch of the binary trellis with the transition path of the edges in the tree of source codewords. If the channel decoder employs the difference between two edge probabilities and tries to keep a synchronization between the state path in the trellis with the code edges, the reliability of the path selection in channel decoding will be greatly improved, which is of great benefit to the decoding performance. The main advantage in paper [14] is just based on this idea.

If we have a first-order Markov relation between two successive symbols instead, the *a priori* bit probability

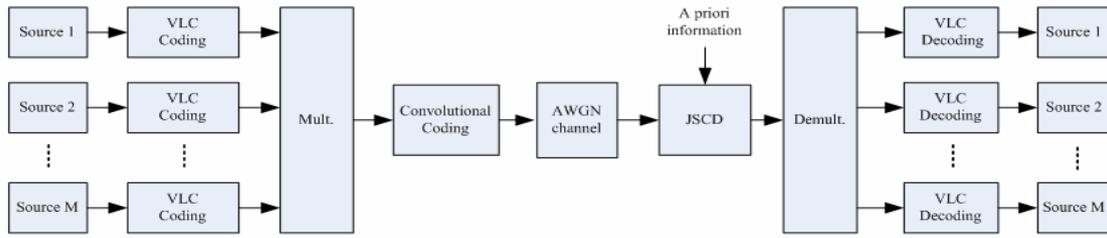


Fig.2 Transmission process for multiple VLC-Encoded sources with a CC

must be computed using conditional probability $p(c^k|c^m)$ as follows

$$P_b(N_i \rightarrow N_j | c^m) = \frac{P_N(N_i | c^m)}{P_N(N_j | c^m)} = \frac{\sum_{c^k \in S_i} p(c^k | c^m)}{\sum_{c^k \in S_j} p(c^k | c^m)}. \quad (3)$$

III. JSCD FOR MULTIPLE VLC-ENCODED SOURCES WITH CCs

In this section we generalize the method described in [14] to VLC-encoded multiple sources with convolutional codes. The transmission scheme satisfied by our method is shown in Fig. 2. The sources are first VLC encoded independently, multiplexed to a serial bit stream, and then fed to the convolutional encoder.

A. A priori Information of Multiple VLC-encoded Sources

If there is only a single source transmitted as in [14], the transition path between the trellis states in adjacent time can be directly associated with a certain edge in the source code tree, as Fig. 3 shows. Let s_t denote the state in the time t . When evaluating the transition path from time $t-1$ to time t , a priori information $P(s_t | s_{t-1})$ equals to a certain edge's probability $P_b(N_i \rightarrow N_j)$.

However, in the case of multiple sources, the sources from different branch are multiplexed in bit level, the adjacent bits belong to different sources respectively, and so the transition path does not belong to any edges. Therefore, unlike the single source, the a priori bit probabilities can not be computed directly. So, how to determine and apply the a priori information of multiple sources at the channel decoder is a main issue in this section.

Take two sources for instance, just as Fig.4 shows. The black points represent the states relevant to source 1, while the white points represent the states relevant to source 2. So the states in adjacent times are associated with the nodes in different source trees respectively. In the trellis decoding procedures, the state in time $t-1$ needs to accompany a node in the source tree 2, while the state in time t needs to accompany a node in the tree of source 1. For each transition path in the trellis between adjacent time, from time $t-1$ to t for example, we can not associate with one edge in a given source tree. But from time $t-2$ to t , we can accompany one edge $N_{1,t} \rightarrow N_{1,j}$ from source tree 1, and its probability $P_b(N_{1,t} \rightarrow N_{1,j})$ representing the influence from the past states. Similarly, there is a path with two edges, from time $t-1$ to $t+1$, we can accompany one edge $N_{2,m} \rightarrow N_{2,n}$ from source tree 2, and its

probability $P_b(N_{2,m} \rightarrow N_{2,n})$ representing the influence on the future states. So, the path selection should consider the influences not only from the past state, but also on the future state. This principle is also true for the cases more than two sources.

B. JSCD Based on Multiple VLC-encoded Sources with CCs

Let the number of source branches being M . As in [14], source symbols are packetized into blocks for VLC encoding in order to limit the error propagation. Since the bit length of each packet encoded sources may not be equal, zero should be filled to ensure the equality of bit numbers for each parallel input. Denoting the bits number of each packet as L , the total bit numbers after multiplexing is then equal to ML . The input sequence of the convolutional code is denoted by $u = (u_1, u_2, \dots, u_{ML})$ and the output is $x = (x_1, x_2, \dots, x_{ML})$. We choose a CC of rate 1/2 in this paper, and then x_i represents the pair (x_{i1}, x_{i2}) . Denote the state sequence by $s^{(i)} = (s_0^{(i)}, s_1^{(i)}, \dots, s_{ML}^{(i)})$ and the received sequence $R = (r_1, r_2, \dots, r_{ML})$. Based on MAP estimation, the task of the decoder is to find an optimal path starting from state $s_0^{(i)} = 0$, passing through ML branches, and finally returning to all zero states, thus maximizing the posterior probabilities

$$\max_i P(s^{(i)} | R). \quad (4)$$

Since R is independent with i , (4) can be written as

$$\begin{aligned} & \max_i P(R | s^{(i)})P(s^{(i)}) \\ \Leftrightarrow & \max_i \sum_{t=1}^{ML} (\ln(P(r_t | s_t^{(i)}, s_{t-1}^{(i)})) + \ln(P(s_t^{(i)} | s_{t-1}^{(i)}))) \end{aligned} \quad (5)$$

Where, $P(r_t | s_t^{(i)}, s_{t-1}^{(i)})$ is the channel transition probability and $P(s_t^{(i)} | s_{t-1}^{(i)})$ is the a priori information. As the noise samples are independent, we have $P(r_t | s_t^{(i)}, s_{t-1}^{(i)}) = P(r_{t1} | x_{t1})P(r_{t2} | x_{t2})$, which depends upon the channel and the modulation used. In the case of an AWGN channel using binary phase-shift keying (BPSK) modulation type, we get

$$P(r_{t1} | x_{t1}) = \frac{1}{\sqrt{\pi N_0}} e^{-\frac{(r_{t1} - (2x_{t1}-1)\sqrt{E_b/2})^2}{N_0}}, \quad (6)$$

and a similar equation for $P(r_{t2} | x_{t2})$, with N_0 being the single-sided noise density, and E_b the energy per transmitted information bit.

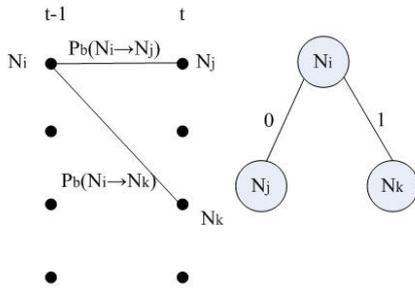


Fig.3 Relationship between the branches and states in case of single source

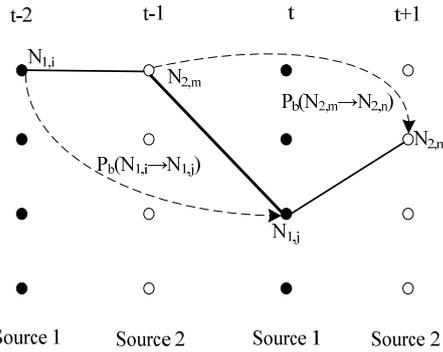


Fig. 4 Branches in state trellis in case of two sources

In the case of single source in [14], the *a priori* information $P(s_t^{(i)}/s_{t-1}^{(i)})$ equals to the edge probability of the code tree. For multiple sources with M channels, $s_t^{(i)}$ is in the same code tree with the previous state $s_{t-M}^{(i)}$ in time $t-M$, while the state $s_{t-1}^{(i)}$ is in the same code tree with the future state $s_{t+M-1}^{(i)}$ in time $t+M-1$. So the *a priori* information $P(s_t^{(i)}/s_{t-1}^{(i)})$ in (5) can be replaced with $P(s_t^{(i)}/s_{t-M}^{(i)})$ and $P(s_{t+M-1}^{(i)}/s_{t-1}^{(i)})$. Let $M(s_{t-1}^{(i)}, s_t^{(i)})$ denote the branch metric value $s_{t-1}^{(i)} \rightarrow s_t^{(i)}$, then

$$M(s_{t-1}^{(i)}, s_t^{(i)}) = \ln(P(r_t | s_t^{(i)}, s_{t-1}^{(i)})) + \ln(P(s_t^{(i)} | s_{t-M}^{(i)})) + \ln(P(s_{t+M-1}^{(i)} | s_{t-1}^{(i)})) \cdot (7)$$

$P(s_t^{(i)}/s_{t-M}^{(i)})$ and $P(s_{t+M-1}^{(i)}/s_{t-1}^{(i)})$ are the corresponding edge probabilities. Therefore, in the process of multiple sources decoding, we should synchronize state of each time in the surviving path with the associated nodes in the code tree. Unlike single source decoding, the node in each code tree is alternative, there is a distance of $M-1$ for the neighboring branch nodes in a same code tree, and the adjacent code trees should be considered when calculating the branch metric value.

In order to use the channel information sufficiently and improve the reliability of the decoder, this paper adopts the soft decision decoding metric and the channel output signal is quantized with Q ($Q > 2$) levels. The soft decision decoder tries to find a path with minimal soft decision distance to the received sequence R . Let $d_s(r_t, x_t)$ represents the soft decision distance between the codeword received in time t and the possible output codeword x_t , (7) can be simplified as

$$M(s_{t-1}^{(i)}, s_t^{(i)}) = d_s(r_t, x_t) - w * (\ln(P(s_t^{(i)} | s_{t-M}^{(i)})) + \ln(P(s_{t+M-1}^{(i)} | s_{t-1}^{(i)}))) \cdot (8)$$

w is the weight of the *a priori* information and can be well chosen according to the channel quality.

The goal of the decoding process is to find a path with minimal accumulated metric value.

$$\min_i \sum_{t=1}^{ML} M(s_{t-1}^{(i)}, s_t^{(i)}) \cdot (9)$$

Since ML is quite large and the memory size is limited, dock-tailed code is always adopted, that means the decision result is output until the data of τ ($\tau \ll L$) time instants have been received and processed. Denote the storage length of the convolutional encoder by m and τ is generally chosen as $\tau = (5 \sim 10)m$.

Compared with the traditional Viterbi algorithm, two accessorial steps should be added:

a) Record the states with the nodes in the corresponding code tree, then compute the *a priori* bit probability of the competing path.

b) Add the *a priori* bit probability to the branch metric value as (8).

Because the *a priori* bit probabilities in code trees can be calculated in advance, so we only need to lookup table and need not to calculate in each step. For the CCs with the constraint length of m and the rate of $1/2$, the cost in a) is $2^m \times 2$, and b) is 2^m , so the computational complexity added is $ML(2^m \times 3)$. Compared with traditional Viterbi algorithm, the complexity increased is very little. While for the storage space, our algorithm only needs to increase the storage of the *a priori* bit probabilities in code tree, since the code tree is not very big in reality, the increased storage quantity is neglectable.

C. Simulation Results

To evaluate the performance of the joint decoding of multiple sources using the *a priori* information, experiments have been performed on the first-order Markov source given in [14] for comparison. As we know, when using VLCs, a single bit error can produce a loss of synchronization, which will lead to the error propagation. Since the bit-error rate (BER) is no longer the correct measurement, we also use the symbol-error rate (SER) for the evaluation of the whole transmission scheme in this paper. Because the SER of different sources may not be equal, we adopt the average SER of all sources to evaluate the performance of the transmission scheme.

For source symbols in a packet, the posterior the position of bit error is, the less the influence of the error propagation. Hence, for multiple sources, the reducing of the packet size can reduce the influence of the error propagation. The experiment results in the condition of different packet size for two sources are depicted on Fig.5. The convolutional coder is the commonly used (2,1,4) coder and the generator polynomials are $[1+D+D^4, 1+D+D^2+D^3+D^4]$. Obviously, the smaller the packet size, the better the decoding performance, as can be seen from Fig.5, the performance with the packet size 64 and 128 is better than 256. However, if the packet size is too small, the convolutional encoder needs to initialize frequently,

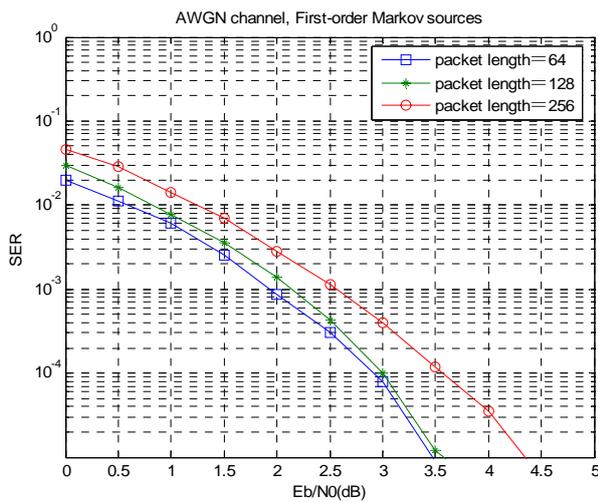


Fig. 5 SER with different packet size for two sources with CCs

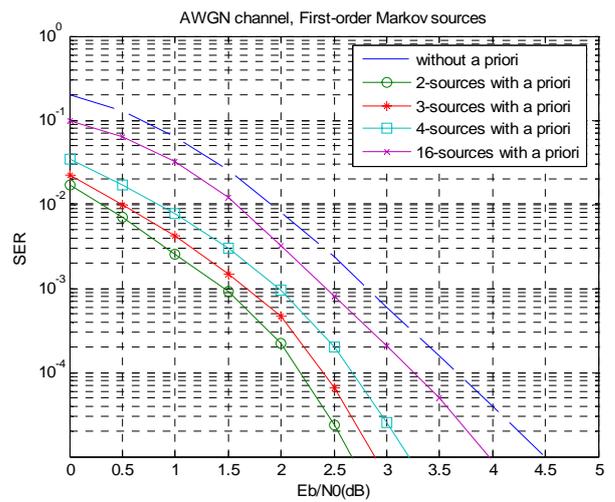


Fig. 7. SER with and without use of a *priori* source information for different size of source channels with a (2,1,6) CC

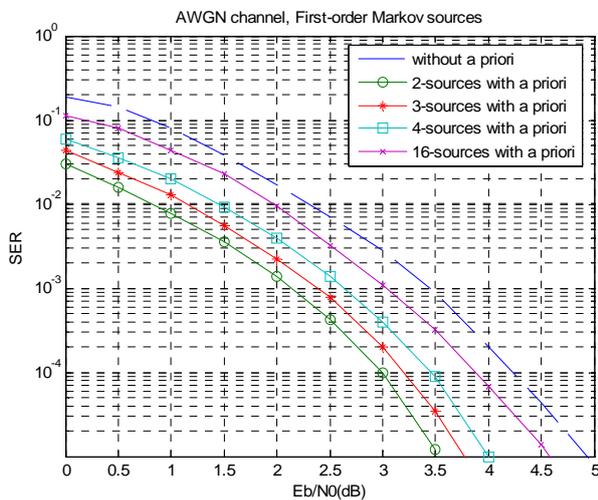


Fig. 6 SER with and without use of a *priori* source information for different number of sources with a (2,1,4) CC

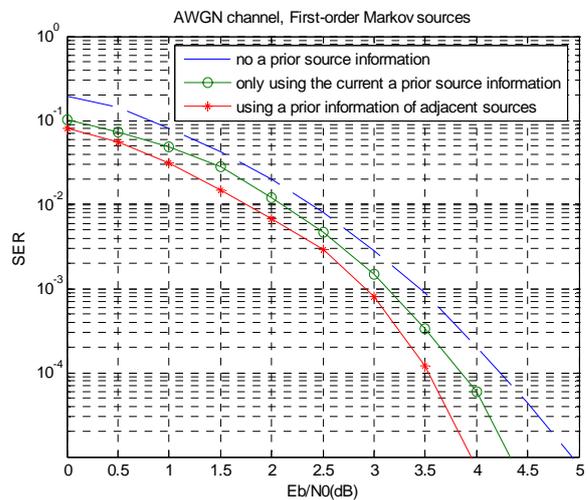


Fig. 8 SER for 8-sources in three cases: not using a prior information, only using the current a prior source information and using the a prior information of adjacent sources

which will increase the computational complexity. The experiment afterwards chooses the packet size as 128 for a tradeoff between complexity and performance.

Simulation result in Fig.6 compares the performance of the joint systems using and not using a *priori* information for different number of sources with a (2,1,4) CC. If the *a priori* information is not used, the decoding performance has no correlation with the number of sources. As shown in Fig.6, the multiple sources JSCD using the *a priori* information improve the performance evidently. For two sources JSCD, the gain in SER is around 1.5dB when $SER = 10^{-2}$ and around 1.3dB when $SER = 10^{-3}$. However, with the increase of the source branches, the performances of our decoding method will decrease. But there is still a gain around 0.5 dB when the source branches number is increased to 16.

Fig.7 provides the decoding results for (2,1,6) CC with different branches of sources, the generator polynomials of the CC are $[1 + D^3 + D^4 + D^5 + D^6, 1 + D + D^3 + D^4 + D^6]$. As can be seen from Fig.6 and Fig.7, with the larger storage length of CCs, the improvement of

our decoding algorithm is more prominent. For two sources JSCD, there is a gain around 1.7dB when $SER = 10^{-2}$ and around 1.5dB when $SER = 10^{-3}$.

The *a priori* information of source $P(s_t^{(i)} | s_{t-M}^{(i)})$ and the *a priori* information of source in the neighboring branch $P(s_{t+M-1}^{(i)} | s_{t-1}^{(i)})$ are considered in our decoding algorithm. To evaluate the effect of *a priori* information of adjacent sources, experiments have been performed in the following three cases: not using a *priori* information, only using the current a prior source information and using the a prior information of adjacent sources.

Fig.8 presents the results of the three cases mentioned above, where the source branch number is equal to 8 and we use the (2,1,4) convolutional code for experiment. When $SER = 10^{-3}$, JSCD using a *priori* information of adjacent sources brings a gain around 0.7dB over the one that not using the *a priori* information, and around 0.3dB over the one that only using the current *a priori* source information. It testified the theory we proved above that there is an obvious improvement when using a *priori*

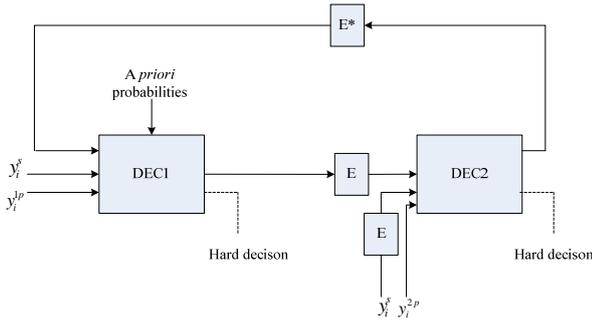


Fig. 9 Turbo decoding with a *priori* probability from the source

information of adjacent sources in JSCD, at the same time, confirmed the theory of Shannon that the redundancy in the source can be used by the receiver to resist channel noise.

IV. JSCD FOR MULTIPLE VLC-ENCODED SOURCES WITH TCS

A. JSCD Based on Multiple VLC-encoded Sources with TCS

Since using channel and source information we can obtain soft outputs, the method proposed in the previous section can also be applied to a Turbo-Code like iterative decoding. The transmission scheme is the same as Fig. 2, but the convolutional encoder and decoder is replaced by a turbo encoder and decoder. The TC used is a parallel concatenation of two recursive systematic convolutional coders separated by a line-column interleaver, followed by a 1/2 puncturing. The input sequence of the Turbo code is denoted by $u = (u_1, u_2, \dots)$ and the output is $u = (x_1, x_2, \dots)$. Denote the received sequence by $Y = (y_1, y_2, \dots)$. We choose a TC of rate 1/2 in this paper, and then y_i represents the pair (y_i^s, y_i^p) . A parallel decoding scheme is shown in Fig. 9, and the notation used is as follow: y_i^s is the Logarithm of Likelihood Ratio (LLR) sequence associated with the information bits; y_i^{1p} is the LLR sequence associated with the parity bits of DEC1; y_i^{2p} is the LLR sequence associated with the parity bits of DEC2; E and E* are the interleaver and the desinterleaver, respectively.

The turbo decoding requires the computation of extrinsic information for each transmitted bit. Turbo decoding is an iterative process, and the extrinsic information is processed through the two constituent decoders and through a number of iterations. Several algorithms can be used to compute the extrinsic information, such as the Maximum A Posteriori algorithm (MAP) proposed by Bahl Cocke Jelinek and Raviv [8], the SUBMAP algorithm [16], and the Soft Output Viterbi Algorithm [7]. Our JSCD method can be applied to all these algorithm. In this letter, the Max-Log-MAP presented in [16] is chosen, because it offers a good tradeoff between complexity and efficiency.

We first examine how to use the a *priori* source information in a turbo decoder based on the MAP

algorithm. The SUBMAP will be presented afterwards as an approximation of this optimal algorithm. Our goal is thus to compute the a *posteriori* probability (APP) of the transmitted bits

$$APP(u_t) = P(u_t | Y) = \sum_i P(u_t, s_t^{(i)} | Y). \quad (10)$$

The joint probability $P(u_t, s_t^{(i)} | Y)$ is proportional to $P(u_t, s_t^{(i)} | Y)$, then (10) can be replaced by

$$\begin{aligned} APP(u_t) &\sim \sum_i P(u_t, s_t^{(i)} | Y) \\ &= \sum_i \sum_j \beta(s_t^{(i)}) P(u_t, s_t^{(i)}, y_t | s_{t-1}^{(j)}) \alpha(s_{t-1}^{(j)}) \\ &= \sum_i \sum_j \beta(s_t^{(i)}) \alpha(s_{t-1}^{(j)}) \\ &\times P(y_t | u_t, s_t^{(i)}, s_{t-1}^{(j)}) P(u_t | s_t^{(i)}, s_{t-1}^{(j)}) P(s_t^{(i)} | s_{t-1}^{(j)}) \end{aligned} \quad (11)$$

Where, the forward and backward probabilities at s_t are

$$\begin{aligned} \alpha(s_t) &= P(s_t, y_t) \\ &= \sum_{s_{t-1}} \sum_{u_{t-1}=0,1} \alpha(s_{t-1}) P(u_{t-1}, s_{t-1}, y_{t-1} | s_{t-1}). \end{aligned} \quad (12)$$

$$\begin{aligned} \beta(s_t) &= P(y_{t+1} | s_t) \\ &= \sum_{s_{t+1}} \sum_{u_{t+1}=0,1} \beta(s_{t+1}) P(u_{t+1}, s_{t+1}, y_{t+1} | s_t). \end{aligned} \quad (13)$$

Furthermore, $P(u_t | s_t^{(i)}, s_{t-1}^{(j)})$ in (11) is equal to 0 or 1, depending on whether the branch exists or not, with $P(y_t | u_t, s_t^{(i)}, s_{t-1}^{(j)})$ being the error probability of the AWGN channel and $P(s_t^{(i)} | s_{t-1}^{(j)})$ the a *priori* source information. For multiple sources with M channels, similar to the decoding of CCs in the previous section, $s_t^{(i)}$ is in the same code tree with the previous state $s_{t-M}^{(i)}$ in time $t-M$, while the state $s_{t-1}^{(i)}$ is in the same code tree with the future state $s_{t+M-1}^{(i)}$ in time $t+M-1$. So the APPs can be computed as

$$\begin{aligned} APP(u_t) &= \sum_i \sum_j \beta(s_t^{(i)}) \alpha(s_{t-1}^{(j)}) P(y_t | u_t, s_t^{(i)}, s_{t-1}^{(j)}) P(u_t | s_t^{(i)}, s_{t-1}^{(j)}) \\ &\times P(s_t^{(i)} | s_{t-M}^{(i)}) P(s_{t+M-1}^{(j)} | s_{t-1}^{(j)}) \end{aligned} \quad (14)$$

The last equation (14) is the basis of the Turbo decoding. As shown in Fig. 9, decoder DEC1 computes an extrinsic probability using channel and source information for each transmitted bit, then decoder DEC2 performs the decoding using these a posteriori channel information. However, the second decoder can not use the a *priori* information the same way because the interleaver breaks the links between the bits of the Huffman coded source.

In our letter, a Max-Log-MAP is used to limit the computational complexity. Denoting a set of n positive real numbers by $\lambda_i, i \in \{1, \dots, n\}$, the central approximation of this algorithm is given by

$$\ln\left(\sum_{i=1}^n e^{\lambda_i}\right) \approx \max_{i \in \{1, \dots, n\}} \{\lambda_i\}. \quad (15)$$

using this latter relation we get approximations of the different probabilities that are used in our simulations

$$\tilde{\alpha}(s_t) = \max_{s_{t-1}, u_t} (\tilde{P}(u_t, s_t, y_t | s_{t-1}) + \tilde{\alpha}(s_{t-1})) \quad (16)$$

$$\tilde{\beta}(s_t) = \max_{s_{t+1}, u_{t+1}} (\tilde{P}(u_{t+1}, s_{t+1}, y_{t+1} | s_t) + \tilde{\beta}(s_{t+1})) \quad (17)$$

$$\begin{aligned} & \ln P(u_t, s_t, Y) \\ &= \max_{s_{t-1}} (\tilde{\beta}(s_t) + \tilde{P}(u_t, s_t, y_t | s_{t-1}) + \tilde{\alpha}(s_{t-1})) \end{aligned} \quad (18)$$

B. Simulation Results

The generators used in the Turbo code are equal to $(1 + D + D^2 + D^4)/(1 + D^3 + D^4)$. The rate is equal to 1/2 and the size of the interleaver is 64×64 . Then the packet size is equal to 4096, contrary to the previous cases, the

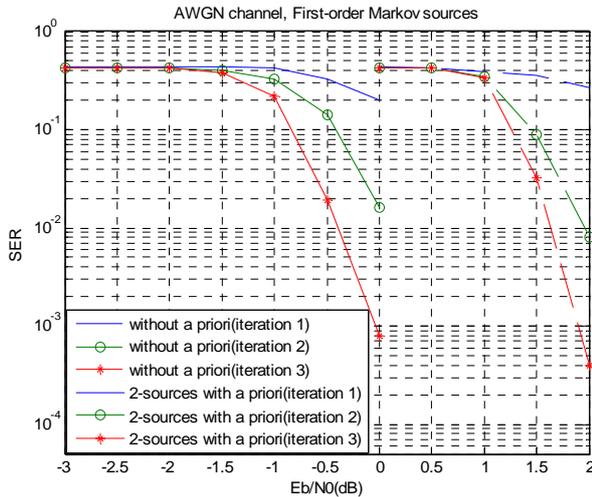


Fig.10 SER for 2-sources with and without use of a *priori* source information for different times of iteration with a TC.

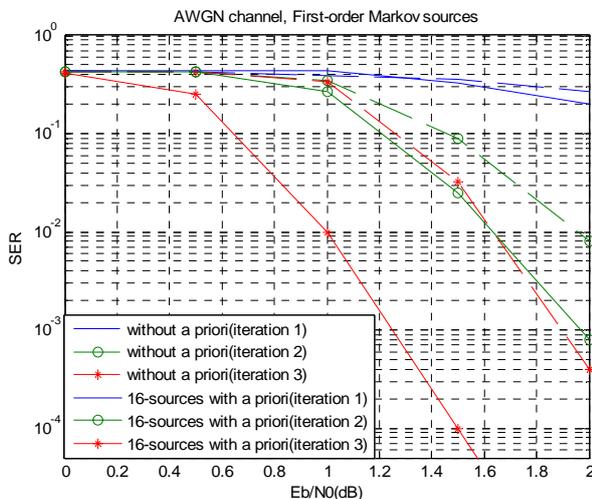


Fig.11 SER for 16-sources with and without use of a *priori* source information for different times of iteration with a TC.

packets are given in bits and not in symbols. As we work with VLCs, it means that a packet does not obviously correspond to an integer number of symbols, so some extra bits are added.

In our simulations, we compute the SER as a function of the signal-to-noise ratio for the first three iterations of turbo decoding. Result in fig.10 compares the performance of the 2-source joint systems using a *priori* information and the system not using the source information. The gain in SNR is around 2 dB when $SER = 10^{-2}$ and take three iterations. Similar to JSCD with CCs, with the increase of the source branches, the performance of the decoding method will decrease, but still has a gain around 0.7 dB when the source branches number is increased to 16.

V. CONCLUSION

We have proposed a methodology for multiple VLC encoded sources decoding, which is the extension of the JSCD method proposed by Marion Jeanne. We first transfer the symbol probabilities into the bit probabilities and then use the *a priori* bit information at the channel decoder. In the decoding process, not only the influence of the previous state to the current state, but also the current to the future have been considered, thus, as shown in the simulation results, by using the *a priori* information among the neighbors' sources reasonably, the decoding performance is remarkably improved. The simulation has indicated that the method in this paper obtain the gain evidently only when the number of source branches is not too big, issues involved with great source branch numbers still need to be studied in future.

REFERENCES

- [1] C E Shannon, "A mathematical theory of communication". Bell System Technical Journal, vol. 27, no. 27, pp. 379-423, 1948.
- [2] K. Sayood and J. C. Borckenhagen, "Use of residual redundancy in the design of joint source/channel coders," IEEE Trans. Commun., vol. 39, no. 6, pp. 838-846, 1991.
- [3] F. Alajaji, N. Phamdo, and T. Fuja, "Channel codes that exploit the residual redundancy in CELP-encoded speech," IEEE Trans. Speech Audio Processing, vol. 4, no. 5, pp. 325-336, 1996.
- [4] N. Phamdo and N. Farvardin, "Optimal detection of discrete Markov sources over discrete memoryless channels applications to combined source-channel coding," IEEE Trans. Inform. Theory, vol. 40, no. 1, pp. 186-193, 1994.
- [5] K. Sayood, F. Liu, and J. D. Gibchild, "A constrained joint source/channel coder design," IEEE J. Select. Areas Commun., vol. 12, no. 9, pp. 1584-1593, 1994.
- [6] M. Park and D. J. Miller, "Decoding entropy-coded symbols over noisy channels by MAP sequence estimation for asynchronous HMMs," in Proc. 32nd Annual Conference on Information Sciences and Systems (CISS '98), pp. 477-482, Princeton, NJ, USA, March 1998.
- [7] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with softdecision outputs and its applications," in Proc. IEEE Global Telecommunications Conference (GLOBECOM '89), vol. 3, pp. 1680-1686, Dallas, Tex, USA, November 1989.
- [8] 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. 20, no. 2, pp. 284–287, 1974.
- [9] A. H. Murad and T. E. Fuja, "Joint source-channel decoding of variable-length encoded sources," in *Proc. Information Theory Workshop (ITW'98)*, pp. 94–95, Killarney, Ireland, June 1998.
- [10] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, no. 9, pp. 2449–2457, 1995.
- [11] J. Garcia-Frias and J. D. Villasenor, "Joint turbo decoding and estimation of hidden Markov sources," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1671–1679, 2001.
- [12] G.-C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Non-systematic turbo codes for non-uniform i.i.d. sources over AWGN channels," in *Proc. Conference on Information Sciences and Systems (CISS '02)*, Princeton, NJ, USA, March 2002.
- [13] L. Guivarch, J.-C. Carlach, and P. Siohan, "Joint sourcechannel soft decoding of Huffman codes with turbo-codes," in *Proc. IEEE Data Compression Conference (DCC '00)*, pp. 83–92, Snowbird, Utah, USA, March 2000.
- [14] M. Jeanne, J.-C. Carlach, and P. Siohan, "Joint source-channel decoding of variable-length codes for convolutional codes and turbo codes," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 10–15, 2005.
- [15] K. P. Subbalakshmi and J. Vaisey, "On the joint source-channel decoding of variable length encoded sources: The BSC case," *IEEE Trans. Commun.*, vol. 49, pp. 2052–2055, Dec. 2001.
- [16] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log-domain," in *Proc. Int. Conf. Commun.*, Seattle, WA, Jun. 1995, pp. 1009–1013.

Weiwei Xiang was born in Jiangsu, China, in April 1979. She received the B.Sc. degree and the M.Sc. degree both in communication engineering, from Zhengzhou Information Science and Technology Institute, China, in 1999 and 2005, respectively. She is currently pursuing the Ph.D. degree. Her research interests include joint source-channel coding, signal processing in communications.

Peizhong Lu was born in Shanghai, China, in 1961, Post doctorate, professor, Ph.D. supervisor in Fudan University. He received the Ph.D degree in mathematics, from CAS (Chinese Academy of Sciences), China, in 1998. His research interests include joint source-channel coding, signal processing in communications, mathematics, parallel computing and information security.

He is currently a Professor at the Department of Computer Science, Fudan University, Shanghai, China.