# Error Restricted Fast MAP Decoding of VLC

Ming Jia, Jiangtao Wen, Shaozhi Ye, and Xing Li

*Abstract*— **Joint source channel techniques based on Variable-Length Coding (VLC) have been widely used. One of the most famous VLC decoders is optimal Maximum A Posteriori (MAP) decoder based on directed graph search and soft-input theory. Due to the high complexity of directed graph search, many reduced complexity methods have been proposed. In this paper, we propose two error restricted algorithms for fast MAP decoding of VLC and compare them with three existing methods. Simulation results show that our methods outperform existing methods in terms of decoding complexity with nearly the same performance on Symbol Error Rate (SER) of optimal decoding. When used in a larger codeword set, the superiority in decoding complexity of our methods is more remarkable.**

*Index Terms*— **Joint source channel decoding, MAP estimation, SISO decoding, variable length code.**

## I. Introduction

FOR both practical and theoretical reason, our research is based on the classical communication scheme which is consists of an encoder block, a channel block and a decoder block. To compare our algorithm with the methods proposed in [1] [2] [3] [4], the source VLC encoding block represents Huffman encoding using BPSK modulation, and the channel block represents a classical transmission channel with Additive White Gaussian Noise(AWGN).

Much work has been done in this area. [4] utilized trellis structures based on a modified form of Viterbi algorithm in VLC decoding when the transmitted symbols number is unknown. By selecting only one state at each bit step, the method in [4] greatly reduced the complexity. Assuming the knowledge of the transmitted symbols number, [2] proposed a computationally complex optimal MAP decoding method and an efficient approximation by directed graph search.

During the process of optimal directed graph search, the computational complexity is square to sequence length. In order to reduce complexity, [3] proposed two approximate methods by pruning some states in the process. These methods are denoted as Approximate Maximum A Posteriori decoder 1 (AMAP1) and AMAP2. At each bit step, different from the optimal search, AMAP1 keeps only one state for every possible symbol number, which is the best in the sense of partial a posteriori probability for all graph states at this specific bit step. Fig. 1 illustrates an example of AMAP1. AMAP2 keeps the same number of states at each bit step. These two methods greatly reduce computational complexity.

Manuscript received March 14, 2005. The associate editor coordinating the review of this letter and approving it for publication was Prof. Marc Fossorier.

M. Jia, S. Ye, and X. Li are with the Dept. of Electronic Engineering, Tsinghua University,Beijing, P. R. China (e-mail: jiam03@mails.tsinghua.edu.cn).

J. Wen is with Mobilygen Corp., Santa Clara, CA.

Digital Object Identifier 10.1109/LCOMM.2005.10005.

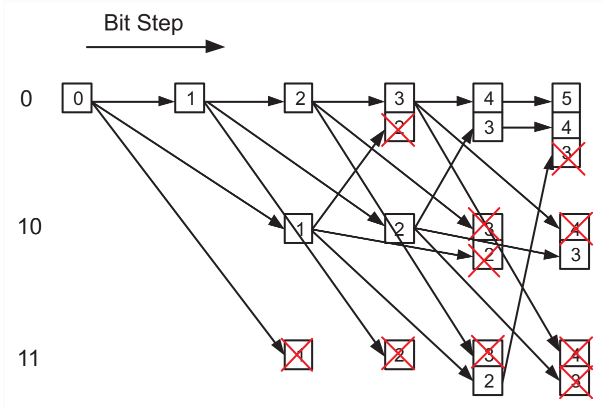


Fig. 1. Trellis structure and decoding example of AMAP1 using codeword set $C_0$.

In this paper, we first partition the decoding process into two operations (reduction and projection) and propose our first decoder: Error-restricted MAP decoder 1 (EMAP1). Then by introducing a threshold $D_{max}$ to select better branches, we proposed our second decoder EMAP2. Simulation results show that our proposed decoders can achieve nearly the same performance on SER as the optimal decoder, while greatly reducing the decoding complexity. Compared with other approximate decoders, our proposed decoders can achieve better tradeoffs between SER and decoding complexity at high Signal Noise Ratio (SNR) case.

The remainder of this paper is organized as follows. In Section 2, we propose two decoders. Simulation results and comparisons are presented and analyzed in Section 3. Finally, Section 4 concludes the whole paper.

## II. Proposed Algorithms

Similar as [4], we define the codeword set as $C$ with dimension $K$. The bit length of codeword $c_i$ is represented by $l_i$. Without losing generality, assume that $l_1 \leq l_2 \leq \cdots \leq l_K$. Further, let $\alpha$ denote the number of different code lengths in the codeword $C$. We then use $S$ to represent the number of symbols in received packet, and $B$ to represent the bit length. The source symbol sequence is $x$, the source bit sequence is $e = e_1, e_2, \cdots, e_B$ and the received bits at VLC decoder are denoted as $o = o_1, o_2, \cdots, o_B$. We define the metric of possible decoded path as $d$. $d$ can be calculated under MAP criterion or simply under Maximum Likelihood (ML) criterion. Without lose of generality, we assume all the decoders will select the branch with smaller $d$ as the decoded path.

We propose here an adaptation of AMAP1, named EMAP1. The operations of each decoder can be partitioned into two

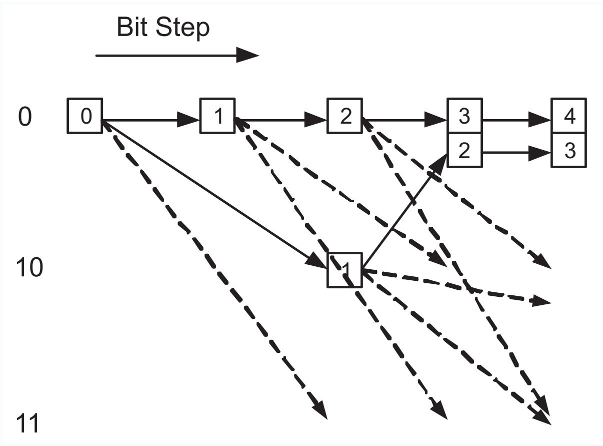


Fig. 2. Decoding process of EMAP2.

steps: 1) at each bit step $q$, there are many states. In order to reduce complexity, all the approximate decoders need to reduce the number of states, this operation is called reduction; 2) after reduction, the decoder projects several branches for each state to the later bit steps, and this operation is called projection. The reduction operation of EMAP1 is as follows: the decoder first reserves only one state for each number of symbols, and then if the number of remained states is larger than $N_{max}$, only the $N_{max}$ states with small $d_i$ value are kept. The projection operation of EMAP1 is different from existing decoders. While existing decoders project all $K$ branches for each state, EMAP1 classifies the branches into $\alpha$ groups by bit length, and projects only one branch with the minimum $d_i$ in each group. For example, when used in the codeword set $C_1$ (described in the beginning of Section 3), previous decoders project all branches ranging from $c_1$ to $c_9$ at every bit step. Instead, EMAP1 decoder calculates the increased metric of each branch as $d_i, i = 1, 2, 3, \cdots, 9$, and then groups these branches by bit length. The projected branches are defined as $l_i, i = 1, 2, 3, 4$, and satisfied $l_1 = min(d_1, d_2)$, $l_2 = min(d_3, d_4)$, $l_3 = min(d_5, d_6, d_7)$, $l_4 = min(d_8, d_9)$.

Because the number of incorrect paths may increase during the bit time, it is stiff to use a fixed $N_{max}$. As an adaptation of EMAP1, $N_{max}$ can be increased during bit time. For example, let $N_{max}(q) = 2 + \lceil q \rceil$ will be more efficient, where $\lceil q \rceil$ means the smallest integer larger than $q$.

Although EMAP1 already reduces a large number of states, there are still so many projected branches from each state which aggravate the burden of the following bit steps. When the channel condition is fairly mild, however, not all the branches need to be projected.

The reduction operation of EMAP2 is the same as EMAP1, while its projection operation is quite different. It employs a threshold $D_{max}$ to prune projected branches. Instead of projecting all $K$ branches for each state, EMAP2 projects branch $c_i$ which satisfies following requirement: $\forall j \leq l_i, |c_{ij} - o_{q+j}| < D_{max}$. For example, considering the case when using $C_0$, $x = \{2, 1, 1\}$, $e = \{1, -1, -1, -1\}$, $o = \{-0.2, -0.8, -0.7, -0.9\}$, $N = 3$, and $D_{max} = 1.3$. At bit step 0, $|c_1 - o_1| = |-1-(-0.2)| < D_{max}$, $|c_{21} - o_1| = |1-(-0.2)| < D_{max}$, $|c_{22} - o_2| = |-1-(-0.8)| < D_{max}$, $|c_{31} - o_1| = |-1-(-0.2)| < D_{max}$, $|c_{32} - o_2| = |1-(-0.8)| > D_{max}$. Then $c_3$ is not satisfied the above requirement, and can not be projected. At bit step 1, 2 and 3, only $c_1$ meet the above requirement and can be projected. Finally at bit step 4, EMAP2 choose the path with the appropriate symbol number, it is to say the path $\{c_2, c_1, c_1\}$, and correctly decoded this packet. Fig. 2 shows the decoding process of above example, where the dashed lines represent the pruned paths.

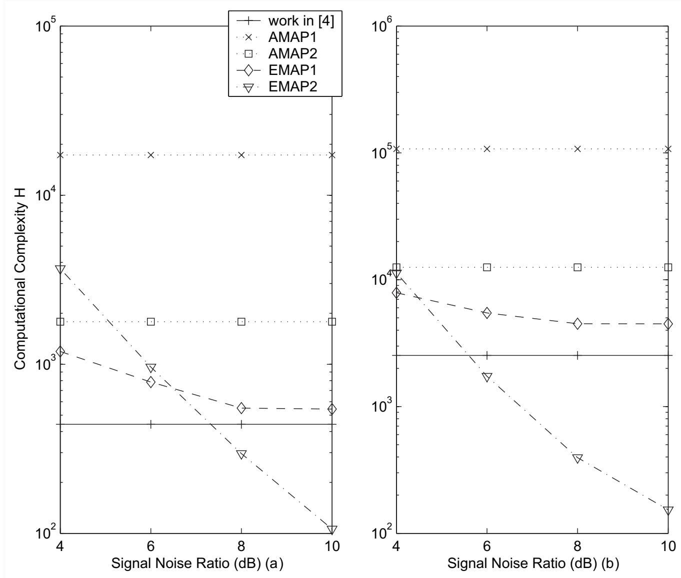


Fig. 3. Computational Complexity of each decoder under different channel conditions. (a) uses codeword set $C_0$, and (b) uses codeword set $C_1$.

## III. Simulation Results

Our experiments are based on BPSK modulation over an AWGN channel. The VLC code alphabets are: $C_0 =$

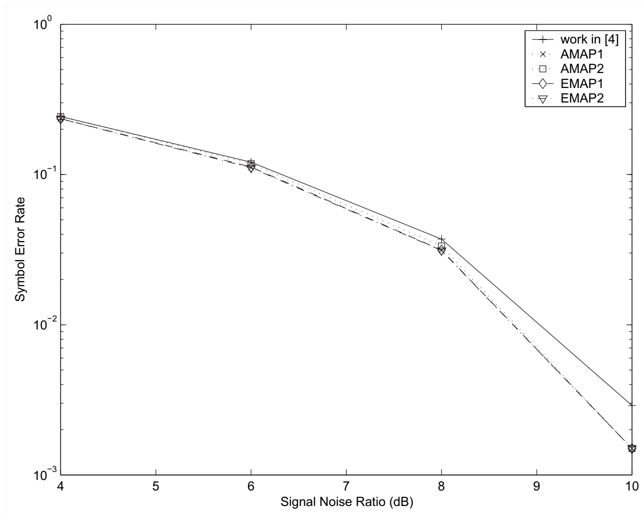


Fig. 4. Symbol Error Rate vs. Signal Noise Raito of each decoder when using codeword set $C_1$. The SER is calculated under Levenshtein distance.

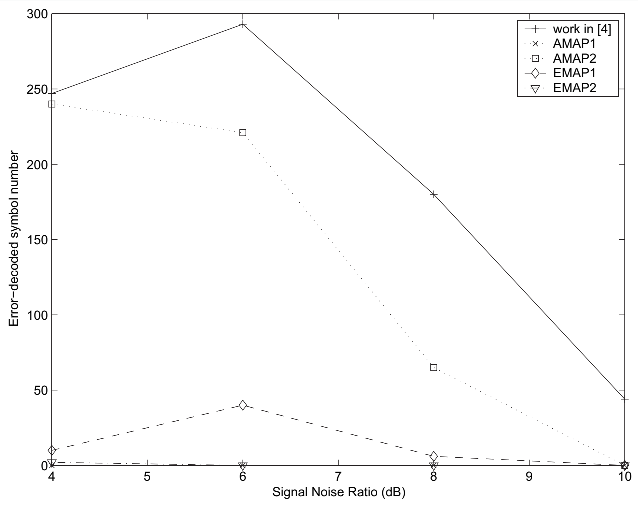


Fig. 5. The total number of error-decoded symbols of each method minus that of AMAP1 when using codeword set $C_1$.

$\{0, 10, 11\}$ with dimension $K = 3$ and symbol probabilities $\{P_1 = 0.5, P_2 = 0.25, P_3 = 0.25\}$; $C_1 = \{00, 01, 100, 101, 1100, 1101, 1110, 11110, 11111\}$ with $K = 9$ and $\{P_1 = \frac{1}{4}, P_2 = \frac{1}{4}, P_3 = \frac{1}{8}, P_4 = \frac{1}{8}, P_5 = \frac{1}{16}, P_6 = \frac{1}{16}, P_7 = \frac{1}{16}, P_8 = \frac{1}{32}, P_9 = \frac{1}{32}\}$ . Each simulation used 300 different packets, and each packet has 100 symbols.

We now define the computational complexity of a decoder as $H = \sum_{q=1}^{B} H_q = \sum_{q=1}^{B} G_q J_q$, where $G_q$ and $J_q$ represent the number of reserved states and the number of projected branches for each state at bit step $q$ respectively. $G_q J_q$ means the overall projected branches at bit step $q$. By this definition, the complexity criterion in work [3] can be represented as $\bar{H} = \sum_{q=1}^{B} G_q$. As decoders in [2] [3] [4] projected all the branches in the codeword set, we have $J_q = K$ and $H = K \sum_{q=1}^{B} G_q$. At this point of view, the definition of $\bar{H}$ can be considered as a special case of our definition of $H$.

Fig. 3 shows the computational complexity of each decoder under different channel conditions. Fig. 3a shows the value of $H$ when using codeword set $C_0$ while Fig. 3b uses codeword set $C_1$. We can see that $H_{emap2}$ is far less than other decoders at high SNR, and this predominance is more notable in larger codeword set (such as in $C_1$).

For every approximate decoder, tradeoff has to be made between the performance of SER and decoding complexity. We adjust the parameters $N_{max}$ and $D_{max}$ of each approximate decoder for they can have low decoding complexity without losing much on SER performance. Fig. 4 shows the SER of each decoder in Levenshtein distance [5] when using codeword set $C_1$. Because the difference in performance of each decoder is very small, to show our results more clearly, we compute the differences in the overall number of error-decoded symbols between other decoders and AMAP1 decoder (shown in Fig. 5). We can see that the performance on SER of EMAP1 and EMAP2 are very close to AMAP1, and are better than that of other approximate decoders. By noticing that the chosen parameters of each decoder remain the same in Fig. 3, Fig. 4 and Fig. 5, we can conclude that EMAP1 and EMAP2 is better than AMAP2 at high SNR, both in terms of SER performance and decoding complexity.

## IV. Conclusions

We propose here two novel reduced complexity methods in VLC decoding which are based on MAP rule and SISO techniques. Compared with previous methods, such as decoder in [4], optimal MAP decoder in [2] and AMAP1, AMAP2 decoder in [3], our algorithms show good performance and small complexity. The remarkable reduction of decoding complexity is achieved by two-step decoding process (reduction and projection), and then by pruning the number of branches in projection operation. Simulation results show that our proposed decoders can achieve nearly the same performance on SER as the optimal decoder, while greatly reducing the decoding complexity. Compared with other approximate decoders, our proposed decoders can achieve better tradeoffs between SER and decoding complexity at high SNR case. The advantages of our decoder is more remarkable when the dimension of codeword set becomes larger.

## References

[1] J. Wen and J. D. Villasenor, “Soft-input soft-output decoding of variable length codes,” *IEEE Trans. Commun.*, vol. 50, pp. 689-692, May 2002.

[2] 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, pp. 1-6, Jan. 2000.

[3] C. Lamy and O. Pothier, “Reduced complexity maximum a posteriori decoding of variable-length codes,” in *Proc. GLOBECOM 2001*, vol. 2, pp. 1410-1413.

[4] V. Buttigieg and P. G. Farrell, “Variable-length error-correcting codes,” *IEE Proc. Commun.*, vol. 147, pp. 211-215, Aug. 2000.

[5] V. I. Levenshtein, “Binary codes with correction of deletions, insertions and substitution of symbols,” Dokl. Akad Nauk. SSSR, 163, pp. 845-848, 1965