Reduced complexity Maximum A Posteriori decoding of variable-length codes

Catherine LAMY - Olivier POTHIER
Philips Recherche France (PRF)
51 rue Carnot
B.P. 301
F-92156 Suresnes

Abstract— **Until very recently, joint source channel techniques mostly focused on systems using fixed-length coding, eventhough variable-length decoding (VLC) is widely used, particularly in video coding. Typically, VLC bit streams are made channel-robust through packetization and standard forward-error correction (FEC). However, when the channel conditions are fairly mild, FEC can reveal itself bandwidth-inefficient. A variable-rate extension of joint source channel decoding could thus potentially replace FEC under mild conditions or, for noisier channels, could be used together with FEC to ameliorate the coding rate, extending in both cases the range of situations under which the bit stream is adequately protected. We propose here two reduced-complexity VLC soft-input decoding techniques, as well as a comparison with existing algorithms. Experimental results of a new proposed VLC decoding algorithm show very good performance and low complexity.**

Keywords—**joint source channel decoding, MAP estimation, soft-input soft-output decoding, variable length code.**

I. INTRODUCTION

We consider in this article the classical communication scheme presented in Figure 1, where popular variable-length source coding schemes such as the Huffman [1] or Lempel-Ziv [2][3] ones are explicitely included at the source encoder part. The channel block on the other hand may represent either a classical transmission channel or the concatenation of a channel encoder, a transmission channel and a channel decoder.

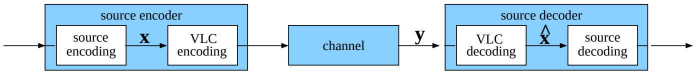

Fig. 1. Communication system model.

VLC schemes, very efficient in terms of source compression, are also very sensitive to channel errors. Generally, the higher the obtained compression factor is, the more sensitive the scheme is to channel errors: as a matter of fact, in “efficient” (compression-like speaking) schemes, a single error usually blows up the whole decoding. One classical way to overcome this problem is to use resynchronisation methods, articifial blocking, synchronization sequences, but also concealment techniques [4]... Unfortunately, in an error-prone environment, it soon becomes obvious that these techniques are often too costful in terms of bitrate and consequently that only improving the VLC decoding could provide substantial amelioration in the recovery of the transmitted data.

The best possible amelioration to the classical decoding algorithm, *i.e.* to the basic bit-by-bit hard-input hard-output decoding (*hard decoding technique*) consists in the determination of the best sequence at the output of the VLC decoder according to the Maximum *A Posteriori* (MAP) rule. Given our system model, this corresponds to finding the estimated sequence $\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} Prob(\mathbf{x}|\mathbf{y})$ where $\mathbf{x}$ and $\mathbf{y}$ are respectively the original and the received sequences.

In the fixed-length code case, MAP decoding is classically achieved by searching for the optimal path in a trellis. This long-known technique can be efficiently implemented via dynamic programming [5][6] and gives very good results, both in terms of performance and complexity. However, in the variable-length code case, the nature of the code itself greatly complicates the decoding operation as there is no more direct relation between the information symbols and the received bits, so the previously mentioned techniques are no more applicable. Recently, several different approaches have been proposed, whether exact but computationally complex MAP decoding methods, or unequally efficient approximations [7][8][9], and whether assuming the knowledge of the transmitted symbols number [7][9] or not [10][11].

Our paper is structured as follows. Section II introduces briefly the well-known optimal MAP decoding of variable-length codes and proposes two new reduced-complexity MAP versions whose construction is compared to the one described in [12]. Simulation results are presented and analysed in Section III. Finally, Section IV draws some conclusions.

II. EXACT AND APPROXIMATE MAP DECODING TECHNIQUES

A. Existing MAP VLC decoding techniques

As mentioned previsouly, whereas in the case of fixed-length codes sequence, MAP decoding can be viewed as the search for the optimal (or equivalently with the best metric) path within a trellis, the case of variable-length codes needs more complicated graphs to be solved. The first works in the domain of such graph decoding have been those of Sayood [7][13] and Park & Miller[9][12]. The first one, introduced by Demir & Sayood [7], relies on a path metric which incorporates both channel and source statistics. In the standard Viterbi decoder [5], at each step only one of the paths entering a state is kept as the *survivor path* and the others are pruned. It can be shown that the pruning does not affect the optimality of the sequence estimation when the applications use fixed length path labels. However, in the case of variable length path labels, different paths entering a state have used up a different number of bits from the received sequence and can therefore be extended differently. The pruning would then affect the optimality of the final selection, so a classical trellis decoding can not be used.

New graph representations, neatly unified and summarised in [14], have consequently been introduced, that keep as many survivors as there are paths with a different number of symbols (resp. number of bits) coming at the considered state at a given bit time (resp. symbol time). Examples of theses graphs, respectively denoted *symbol-constrained directed graph* and *bit-constrained directed graph*, are recalled in Figure 2 and Figure 3 for the variable-length code of dimension $K = 3$ and maximal length $L_{max} = 2$ with codewords set $\{0, 10, 11\}$.

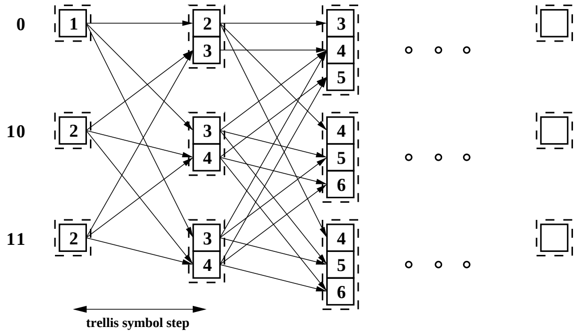

Fig. 2. Symbol-constrained directed graph representation for VLC decoding.

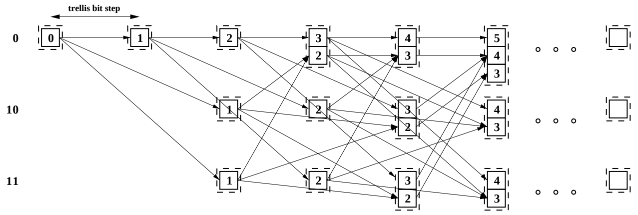

Fig. 3. Bit-constrained directed graph representation for VLC decoding.

It has been shown in [12] that these two representations are not as similar as they first seem. In fact, whereas they are equivalent in the case of the optimal MAP decoding, those two graphs will let differences appear when state reduction operations are proposed. In fact, the reductions will consist in the comparison of symbol sequences of different lengths (each dashed set represented in Figures 2 and 3) and pruning some of them has not the same effect in both cases. While the bit-constrained method performs a state reduction that is consistent with a MAP criterion and finds the symbol sequence that is MAP-optimal on the resulting reduced graph via dynamic programming, the symbol-constrained one however only implements an approximation of the symbol-constrained MAP rule. In the first case indeed, the comparison is done between paths containing the same number of bits, hence with coherent *a priori* probabilities contributions while in the second case the number of bits differs. Taking this important result into consideration, we choose from now on to only consider comparison and complexity reduction techniques with the best method, that is to say the bit-constrained one.

B. New reduced complexity MAP VLC decoding techniques

The main problem of the decoding is the huge number of states in the graph when the sequence length grows. In fact, it is obvious from the construction of the decoding graph, illustrated in Figure 3 for a code of dimension 3, that the number of states per bit 'time' or trellis bit step is linear with the value of the trellis bit step parameter. Such a complexity being obviously prohibitive, Park & Miller proposed to prune all states but one in each dashed set of their bit-constrained representation. While this state reduction leads to a drastic complexity reduction, it however loses in terms of performance, as will be shown in Section III. We propose in this paper a first method, denoted *Approximate Maximum A Posteriori decoding 1* (AMAP-1) which leads to a different state reduction operation. Choosing to apply a method very similar to dynamic programming, we still rely first on a forward propagation with metric derivation process, where the pointers to previous states are saved and in the second step a traceback process to establish the optimal sequence. Whereas in standard dynamic programming, this first forward operation would save at each time and for each number t of symbols the best partial sequence terminating in each graph state, we propose here to keep only one sequence for each number t of symbols at each time. The saved sequence will be the best in the sense of partial *a posteriori* probability for all graph states at the considered trellis bit step. An example of this state reduction is given in Figure 4 where the crosses show the states that are removed. Note that when a state is removed, it implies that the branches that would have come from it are no more considered as likely candidates: the reduction of the trellis size is clearly noticeable, as the number of surviving states in the trellis at step i (*i.e.* for a partial sequence of i bits) is at most equal to i.

We propose also a second method, denoted AMAP-2 which corresponds to keeping at each trellis bit step the Nb_{max} best states in the sense of partial *a posteriori* probability. An example of this state reduction is given in Figure 5 where the crosses show the states that are removed. Here again the removal of a state implies that the branches that would have come from it are no more considered as likely candidates. The trellis size reduction is even more noticeable and easily adjustable, as the number of surviving states in the trellis at step i is at most equal

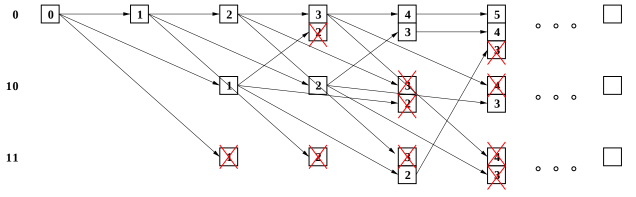

Fig. 4. Bit-constrained directed graph representation for VLC decoding after AMAP-1 state reduction.

to Nb_{max}.

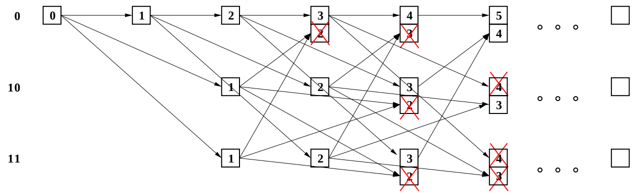

Fig. 5. Bit-constrained directed graph representation for VLC decoding after AMAP-2 state reduction.

In both methods, as well as in Park & Miller suboptimal decoding (denoted PM-AMAP), the final decision is taken between the paths ending with the correct and supposed known number of transmitted symbols.

III. NUMERICAL RESULTS ON DECODING PERFORMANCE AND COMPLEXITY

We tested the two new reduction-state methods with a BPSK modulation over an additive white Gaussian noise channel. In each case we used the variable-length code $\mathcal{C}_0 = \{0, 10, 11\}$ with dimension $K = 3$, maximal length $L_{max} = 2$ and symbol probabilities $\{p(0) = 0.5,\ p(10) = 0.25,\ p(11) = 0.25\}$ and the variable-length code $\mathcal{C}_1 = \{0, 100, 101, 110, 1110, 1111\}$ with dimension $K = 6$, maximal length $L_{max} = 4$ and symbol probabilities $\{p(0) = 0.5,\ p(100) = 0.15,\ p(101) = 0.17,\ p(110) = 0.08,\ p(1110) = 0.06,\ p(1111) = 0.04\}$ [15].

Figures 6 and 7 present performance results for several reference curves: classical hard VLC decoding performance (in circles), optimal soft-input VLC decoding (in squares) and PM-AMAP (in triangles up). We propose to compare those curves with the two solutions we elaborated: AMAP-1 (in crosses) and AMAP-2 (in triangles right) for which the number $Nb_{max} = 3$ of states to keep at each step was chosen equal to the one obtained for the PM-AMAP to ensure a fair comparison.

As expected, all performance are bounded by those of the MAP (optimal) decoding (in squares) and those of the hard decoding. From Figure 6, it appears that AMAP-1 and AMAP-2 methods perform better than the PM-AMAP one[1]. A study of the trellis complexity for each of these soft-input methods is given in Figure 8, that constitutes a good evaluation of the overall algorithm complexity since the number of transitions per state is the same for each algorithm, and consequently

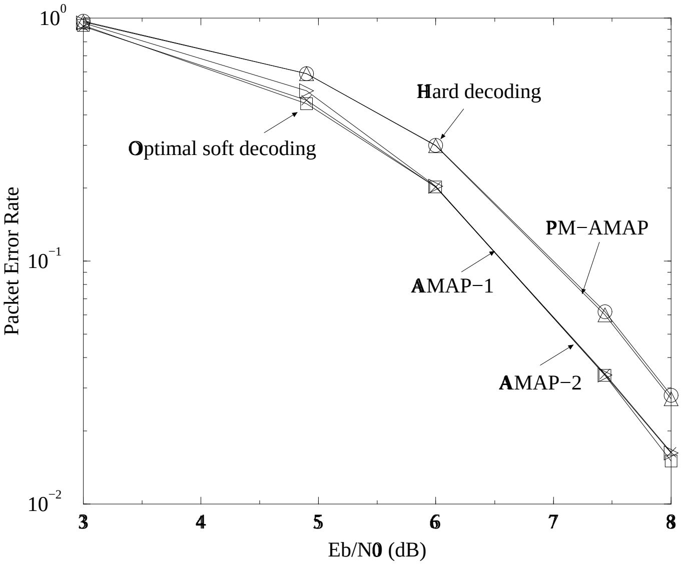

Fig. 6. Packet Error Rate vs. signal-to-noise ratio E_b/N_0 performance for various soft VLC decoding algorithms when applied to code $\mathcal{C}_0$ for frames of 100 transmitted symbols.

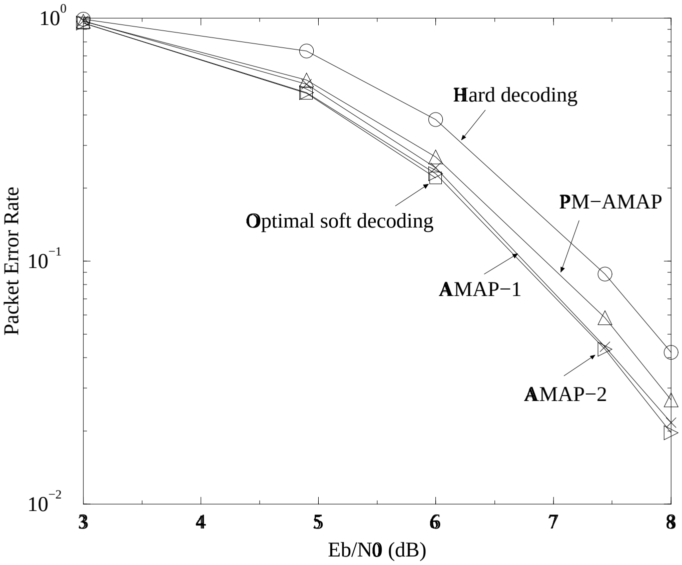

Fig. 7. Packet Error Rate vs. signal-to-noise ratio E_b/N_0 performance for various soft VLC decoding algorithms when applied to code $\mathcal{C}_1$ for frames of 100 transmitted symbols.

---

[1] Although PM-AMAP performs significantly better than hard decoding in terms of bit error rate, it does not always seem to achieve susbtantial benefits in terms of packet error rate.

represents a first approach in terms of design and cost feasibility for the several approximate algorithms. It appears that while the optimal soft VLC decoding and our first suboptimal method (AMAP-1) show a complexity linear with the trellis bit steps, PM-AMAP suboptimal decoding and our second method (AMAP-2) are both independent of the trellis bit steps value. Similar results can be obtained when considering the case of code $\mathcal{C}_1$. From the results presented in those three figures, it appears that the overall best solution is the AMAP-2 method, since it gives a better packet error rate (PER) with lower trellis complexity when compared to existing sub-optimal methods.

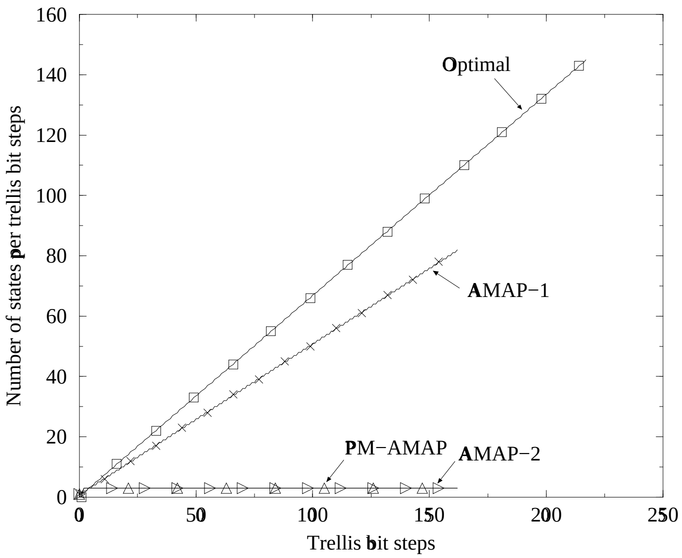

Fig. 8. Trellis complexity study for various soft VLC decoding algorithms when applied to code $\mathcal{C}_0$ for frames of 100 transmitted symbols.

IV. CONCLUSIONS

We proposed here to use a joint source channel (de)coding method that uses the residual source redundancy as a form of implicit channel protection. The decoder acts then as a statistical estimator of the transmitted bitstream. Two new reduced-complexity MAP decoding techniques for variable-length codes are described, valid for both a hard input and a soft input at the entrance of the considered VLC decoder. Obviously, the method shows its strength mainly for soft input.

As a matter of fact, the performance obtained by simulation with the A-MAP algorithm reach the optimal soft-input performance for a trellis complexity independent of the trellis bit steps value.

REFERENCES

[1] D. A. Huffman, “A method for the construction of minimum redundancy codes,” *Proceedings of the IRE*, pp. 1098–1101, vol. 40, 1951.

[2] J. Ziv and A. Lempel, “A universal algorithm for data compression,” *IEEE Transactions on Information Theory*, pp. 337–343, vol. 23, no. 3, May 1977.

[3] J. Ziv and A. Lempel, “Compression of individual sequences via variable-rate coding,” *IEEE Transactions on Information Theory*, pp. 530–536, vol. 24, no. 5, September 1978.

[4] Y. Wang and Q. Zhu, “Error control and concealment for video communication: a review,” *Proceedings of the IEEE*, pp. 974–997, vol. 86, May 1998.

[5] A. J. Viterbi and J. K. Omura, *Principles of Digital Communication and Coding*, New York: McGraw-Hill, 1979.

[6] N. Phamdo and N. Farvardin, “Optimal detection of discrete Markov sources over discrete memoryless channels – Application to combined source channel coding,” *IEEE Transactions on Information Theory*, pp. 186–193, vol. 40, no. 1, January 1994.

[7] N. Demir and K. Sayood, “Joint source/channel coding for variable length codes,” *Proceedings of the Data Compression Conference (DCC’98)*, Snowbird, Utah, USA, pp. 139–148, March 30-April 1, 1998.

[8] M. Park and D. J. Miller, “Decoding entropy-coded symbols over noisy channels by MAP sequence estimation for asynchronous HMMs,” *Proceedings of the Conference on Information Sciences and Systems (CISS’98)*, Princeton, NJ, USA, pp. 477-482, March, 1998.

[9] M. Park and D. J. Miller, “Joint source-channel decoding for variable length encoded data by exact and approcimate map sequence estimation,” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP’99)*, Phoenix, Arizona, USA, pp. 2451–2454, March 15-19, 1999.

[10] A. H. Murad and T. E. Fuja, “Joint source-channel decoding of variable-length encoded sources,” *Proceedings of the IEEE Information Theory Workshop (ITW’98)*, Killarney, Ireland, pp. 94-95, June, 1998.

[11] J. Wen and J. D. Villasenor, “Utilizing soft information in decoding of variable length codes,” *Proceedings of the Data Compression Conference (DCC’99)*, Snowbird, Utah, USA, pp. 131–139, March 29-31, 1999.

[12] M. Park and D. J. Miller, “Joint source-channel decoding for variable-length encoded data by exact and approximate MAP sequence estimation,” *IEEE Transactions on Communications*, pp. 1–6, vol. 48, no. 1, January 2000.

[13] H. H. Otu and K. Sayood, “A joint source/channel coder with block constraints,” *IEEE Transactions on communications*, pp. 1615–1618, vol. 47, no. 11, November 1999.

[14] R. Bauer and J. Hagenauer, “Turbo-FEC/VLC-decoding and its application to text compression,” *Proceedings of the Conference on Information Sciences and Systems (CISS’00)*, Princeton University, USA, pp. WA6-6–WA-11, March 15-17, 2000.

[15] N. Moreau, *Techniques de compression des signaux*. Paris: Masson, 1995.