JOINT SOURCE-CHANNEL TURBO DECODING OF VLC-CODED MARKOV SOURCES

E. Fabre, A. Guyader, C. Guillemot

IRISA-INRIA, Campus universitaire de Beaulieu,
35042 Rennes Cedex, FRANCE. E-mail : name@irisa.fr

ABSTRACT

We analyse the dependencies between the variables involved in the source and channel coding chain. This chain is composed of 1/ a Markov source of symbols, followed by 2/ a variable length source coder, and 3/ a channel coder. The output process is analysed in the framework of Bayesian networks, which provide both an intuitive representation of the structure of dependencies, and a way of deriving joint (soft) decoding algorithms. Joint decoding relying on the hidden Markov model (HMM) of the global coding chain is intractable, except in trivial cases, due to the high dimensionality of the state space. We advocate instead an iterative procedure inspired from serial turbo codes, in which the three models of the coding chain are used in alternance. This idea of using separately each factor of a big product model inside an iterative procedure usually requires the presence of an interleaver between successive components. We show that only one interleaver is necessary here, placed between the source coder and the channel coder. As a subproduct, we also derive a soft VLC decoder with good (and adjustable) synchronization properties.

1. INTRODUCTION

The wide usage of variable length codes (VLCs) in data compression has motivated recent work on robust decoding of variable length coded streams [1], [2], [3], [4]. The authors in [2] derive a global stochastic automaton by computing the product of the separate automata of the Markov source, the source coder and the channel coder. The resulting automaton is used to perform a MAP decoding with the Viterbi algorithm. Although states which cannot be reached through any valid sequence of transitions can be eliminated, the state space can remain very large. In [4], the authors propose a serially concatenated iterative system consisting of an outer variable length encoder and of an inner binary convolutional coder separated by an interleaver.

In this paper, we also follow these lines in the case of a very general coding chain, encompassing as particular cases the models of the papers above. We focus on an analysis and modeling of the dependencies between the variables involved in the complete chain of source and channel coding, by means of the Bayesian-network formalism. Our starting point is a state space model of the three different elements in the chain : the source of symbols, the source coder and the channel coder. These models are cascaded to produce the bitstream sent over the channel, and the randomness of variables is introduced by assuming a white noise input of the cascade. The product of these three automata induce immediately a state variable model of the bitstream : the triple of states – one state for each model – appears to be a Markov chain, the transitions of which generate the sequence of output bits, that are sent over the channel. The observed output of a memoryless channel corresponds to noisy measurements of these bits. Therefore, we are exactly in the HMM framework for which fast estimation algorithms are well known.

This nice picture suffers from two difficulties. First, the presence of two time indexes : the symbol clock of the source model, and the bit clock of the channel coder model. The translation is performed by the VLC source coder. Since not all symbols have the same length, the number of bits of the coded sequence (as well as the position of symbol starts) is a random variable, which is quite unusual. We therefore have to solve a joint segmentation + estimation problem. The second difficulty is more classical : it comes from the fact that the state space dimension of the product model explodes in most practical cases, so that a direct application of usual techniques is unaffordable, except in trivial cases.

In this paper, we thus rely on properties evidenced by serial turbo-codes to design an estimation strategy : instead of using the big product model, inference can be done in an iterative way, making use of part of the global model at each time. This decreases complexity since smaller state spaces are involved. We use this idea in the following way, as it was already suggested in [4] : we introduce an interleaver between the source coder and the channel coder. This allows the construction of an iterative soft decoder alternating between the channel coder model and the joint model of the source + source coder [1], with the bit clock as time index. But the idea can be pushed further : why not splitting also the joint model source (MS) + source coder (SC) ? We

[1] By contrast, [4] is assuming an i.i.d. source, which makes the source model useless.

demonstrate that, due to the pointwise translation of symbols into bits, there is no need of an interleaver there. The joint MS+SC model can actually be processed optimally by a sequential use of the SC model, followed by the MS model.

2. PROBLEM STATEMENT

Let $S = S_1...S_K$ be the sequence of quantized source symbols taking their values in a finite alphabet composed of 2^q symbols. The sequence $S_1 \dots S_K$ is assumed to be a Markov chain. The symbols are then coded into a sequence of useful bits $U = U_1...U_N$, by means of a variable length code. The length N of the information bitstream is a random variable, function of S. U is then fed to the channel coder (a convolutional code), which yields the sequence $R = R_1...R_M$ of redundant bits. In the triple (S, U, R), all the randomness comes from S, since U and R are deterministic functions of S. The bitstream (U, R) is sent over a memoryless channel and received as measurements (Y, Z); so the problem we address consists in estimating S given the observed values $y = y_1...y_N$ and $z = z_1...z_M$, pointwise measurements on useful and redundant bits, respectively. Therefore we are exactly in an HMM framework for which fast estimation algorithms exist [5].

3. JOINT MODEL OF THE PAIR SOURCE + SOURCE CODER

Let us first assume that $S_1 \dots S_K$ is a white noise sequence, each S_k obeying a stationary distribution P_s. For a non dyadic P_s, and despite the optimality of the Huffman code, the average length of a codeword remains strictly above the lower bound (at most 1 bit above it). Therefore there remains some correlation between the bits at the output of the coder. This form of redundancy can be modeled and exploited to help the segmentation + estimation of the bitstream (and thus the estimation of the symbol sequence). However, this inner codeword redundancy is quite low. The MS model incorporates the major part of the redundancy.

A model of the distribution of U, capturing both inner codeword and intersymbol correlation, can be obtained by mapping the transition probability $P_t(S_{k+1}|S_k)$ on the Huffman tree, which requires to keep track of the last symbol S_k produced. The state variable of a bit clock model of U thus takes the form $X_n = (v, s)$, where s is the last symbol produced, and v is a vertex on the Huffman tree. It is also necessary to count symbols, in order to ensure that the last state X_N corresponds to a correct segmentation of the bitstream into K symbols. So the actual state variable is $X_n = (v, s, k)$, where k is augmented by 1 each time v reaches a leaf of the Huffman tree. Bits of U are produced by transitions of this Markov chain X, i.e. $U_n = \phi(X_{n-1}, X_n)$. This yields the dependency structure shown on the upper part of fig. 1, which is well adapted to fast MAP or MPM estimation (by a BCJR), provided the state space for X_n is not too large. One can further extend this model with a semi-Markov model of the channel coder (CC), taking U as input and producing R, with state variable X'_n (lower part of fig. 1). A global (product) HMM model of the pair (U, R) follows by gathering the state variables X_n and X'_n, but this model is intractable in practice.

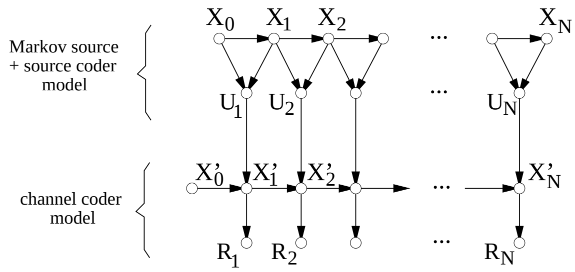

Fig. 1. A graphical model representing dependencies between the model of U and the channel coder model, producing R. X_n is the state of the source + Huffman coder model, and X'_n is the internal state of the convolutional coder. Pointwise measurements Y and Z on U and R are not represented for clarity.

4. ITERATIVE ESTIMATION

A direct estimation based on the global HMM model is only affordable for trivial cases, and should be approximated in most practical situations. We consider instead iterative inference, using in alternance parts of the model.

4.1. Two model case

We first consider a separate representation of the state variables of the two models: X_n for the source + Huffman coder model, and X'_n for the channel convolutional coder model, in order to make apparent dependencies between them. The Bayesian network incorporating the complete chain MS+SC+CC is depicted on fig. 1: the top part represents the bit clock product model for MS+SC, and the bottom part represents the serial concatenation of a convolutional encoder. Variables of R are depicted as functions of the coder state X', but could as well be functions of state transitions. State spaces of variables $X_n = (v, s, k)$ and $X'_n = (m)$ are smaller than in the global HMM, obtained by agregating X_n and X'_n into the same state variable.

The price to pay for this expansion is a quite complex Bayesian network (or Markov field) which is not a tree. Hence, we go out of the range of fast algorithms like BCJR or Viterbi, that only extend to trees. However, if cycles of the graph are long enough, efficient approximate MPM (max of posterior marginals) or MAP estimators can be obtained by running a belief propagation algorithm on the graph as if it were a tree, ignoring the presence of cycles [6, 7]. As discovered by [8], cycles can be made "long" at no cost by simply introducing an interleaver between the two models (fig. 2), which evidences a structure similar to serial turbo codes.

Belief propagation on this graph can arrange message circulations in such a way that it amounts to performing a soft decoding on each model separately. One ends up with an iterative estimation procedure, alternating use of the two models, with exchange of soft information, just like for serial turbo codes. The only difference is in the last step, chosen to be a standard MAP step instead of an MPM step as usual, in order to prevent finding a non decodable bitstream, or a bitstream associated to an incorrect number of symbols. However, the estimation of the joint MS+SC model still remains hardly tractable beyond 4 or 5 bits quantization of the source, for K around several hundreds.

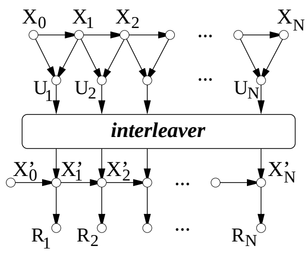

Fig. 2. Introducing an interleaver between the two hidden Markov models augments the minimal length of dependency cycles.

4.2. Three model case.

Therefore, one can further decide to process separately the MS and the SC models, applying the same method. Surprisingly, in this case, there is no need of an extra interleaver, and the successive use of the SC model and the MS model is *optimal*, which is a new result for soft VLC decoding.

The SC model alone assumes an input of independent symbols, and thus only makes use of the intra-codeword redundancy, and of the constraint on the number of symbols. Its state variable X_n reduces to a pair (v, k), since memory of the last symbol is useless. On the other side, the state variable X''_k of the Markov source model cannot be reduced to the last symbol produced (s). Actually, the difficulty of VLC decoding comes from the lack of synchronization between the symbol clock and the bit clock. In other words, the estimation of the transmitted symbols (or bits) must be performed jointly with the *segmentation* of the received bit stream. Hence the MS model state variable must include a counter of the number of bits produced by the first k symbols, which yields $X''_k = (s, n)$. Joint MS+SC soft decoding then amounts to estimating X assuming an input of independent symbols, then translating soft information on X_n into soft information on S_k, which requires some "clock conversion," and finally estimating X''. In other words, the joint MS+SC decoding first assumes a white source for soft VLC decoding, and then takes inter-symbol correlation into account.

For decoding the complete chain MS+SC+CC, we thus end up with a turbo procedure alternating between the two sources of redundancy, the Marko source and the channel code, where the intermediary VLC source coder model is used as a translator of soft information from the bit clock to the symbol clock. Full details can be found in [9].

5. EXPERIMENTAL RESULTS

To evaluate the performance of the joint decoding procedure, experiments have been performed on a first-order Gauss-Markov source, with zero-mean, unit-variance and correlation factor $\rho = 0.9$. The source is quantized with a 16 levels uniform quantizer (4 bits) on the interval $[-3, 3]$, and we consider sequences of $K = 200$ symbols. The VLC source coder is based on a Huffman code, designed for the stationary distribution of the source. The channel code is a recursive systematic convolutional code of rate $1/2$ defined by the polynomials $F(z) = 1 + z + z^2 + z^4$ and $G(z) = 1 + z^3 + z^4$. Since very few errors have been observed with rate $1/2$, we have augmented it to $3/4$ by puncturing the redundant bit stream. A variable size interleaver is introduced between the source coder and the channel coder. All the simulations have been performed assuming an AWGN channel with a BPSK modulation.

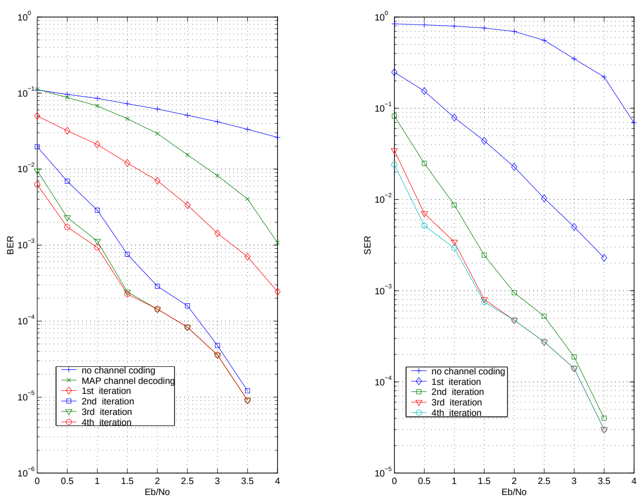

Fig. 3. *Residual BER (left) and SER (right) vs E_b/N_0, for successive iterations, with a Gauss-Markov source.*

Figure 3 provides the residual bit error rates (BER) and symbol error rates (SER) for different channel E_b/N_0. On each plot, the top curve corresponds to an ML estimation of the bitstream assuming independent bits (and no channel coding), followed by a hard Huffman decoding. On the BER plot, the second curve corresponds to a MAP channel decoding, assuming an input of independent bits. The third one is the result of the first iteration, where knowledge on symbol correlation and codeword structure has been introduced. Successive curves show the extra gain of iterations in the procedure, which depends on the degree of redundancy present on both sides of the source coder.

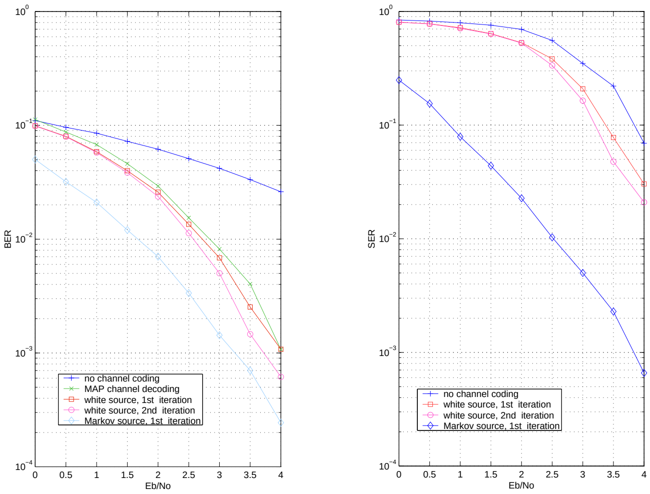

Fig. 4. *Same conditions as the previous figure, except that inter-symbol correlation is not taken into account.*

The same experiments have been performed assuming the symbol source is white (fig. 4), in order to evidence the gain introduced by the intersymbol correlation. On the BER plot, the top curve still represents the error rate without channel coding. The second one is obtained using the CC model only (1st step of the 1st iteration). Then comes the BER after the first iteration for a white noise model, which can be viewed as the BER at the output of the SC model for the Gauss-Markov source. The lowest curve is the BER at the end of the first iteration for the Gauss-Markov source. Hence these four curves help understanding the effect of each component in the model. As expected, the SC model has little influence since it uses little bit correlation and mainly relies on constraints on the number of bits and on codeword structure. Nevertheless, this effect is sufficient enough to evidence some gain in the successive iterations, when symbols are assumed to be independent. A comparison with the Markov source case shows that taking the inter-symbol correlation into account brings a gain of more than 2 dB for the SER.

6. CONCLUSION

The turbo principle, revealed by turbo codes, can be generalized into the iterative use of factors of a big product model. It is a promising strategy that has improved existing estimation algorithms in many problems, at almost no cost. We have shown that this strategy performs well in this specific problem of joint source-channel decoding. However, its use is not always relevant : in the particular case of the product model of the source + source coder, one doesn't need to separate factors by an interleaver. The iterative use of the factors can be optimal. This advocates a careful understanding of dependencies before choosing a turbo strategy.

Finally, let us mention that using the three models separately allows many variations. For example, a variable source coder can be used, in particular to introduce dummy bits at some known symbol positions, in order to facilitate the resynchronization of the symbol stream and the bit stream, which is the major weakness of VLC schemes. This "soft synchronization" possibility could be an alternative to the use of reversible variable length codes (RVLC), and offers the advantage of being easily tunable in terms of rate loss vs resynchronization power.

7. REFERENCES

[1] M. Park and D.J. Miller, "Decoding entropy-coded symbols over noisy channels by MAP sequence estimation for asynchronous HMMs," in *Proc. Conf. on Info. Sciences and Systems*, May 1998.

[2] A.H. Murad and T.E. Fuja, "Joint source-channel decoding of variable length encoded sources," in *Proc. Information Theory Workshop, ITW*, June 1998, pp. 94–95.

[3] N. Demir and K. Sayood, "Joint source-channel coding for variable length codes," in *Proc. IEEE Data Compression Conference, DCC*, March 1998, pp. 139–148.

[4] R. Bauer and J. Hagenauer, "Iterative source/channel decoding based on a trellis representation for variable length codes," in *Proc. IEEE Intl. Symposium on Information Theory, ISIT*, June 2000, p. 117.

[5] . R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rates," *IEEE Trans. on Information Theory*, pp. 284–287, March 1974.

[6] B.J. Frey and D.J.C. MacKay, "A revolution: belief propagation in graphs with cycles," in *Proceedings of the Neural Inform. Processing Systems Conf.*, December 1997.

[7] R.J. McEliece, D.J.C. MacKay, and J.-F. Cheng, "Turbo decoding as an instance of pearl's belief propagation algorithm," *IEEE J. on Sel. Areas in Com.*, vol. 16, no. 2, pp. 140–152, February 1998.

[8] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. on Communications*, vol. 44, no. 10, October 1996.

[9] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, "Joint source and channel turbo decoding of entropy-coded sources," *submitted to IEEE J. on Sel. Areas in Com.*, 2000.