

# New Iterative Decoding of Variable Length Codes with Turbo-Codes

Xavier Jaspas and Luc Vandendorpe

Commun. and Remote Sensing Lab. — Université catholique de Louvain (UCL)

Place du Levant, 2 — B-1348 Louvain-la-Neuve, Belgium

Email: jaspas@tele.ucl.ac.be, vandendorpe@tele.ucl.ac.be

**Abstract**—A new joint source-channel (JSC) turbo-decoding scheme, with three soft-in/soft-out (SISO) modules, is presented for variable length coded sources protected by a turbo-code. This new scheme offers the advantage to combine the powerfulness of the used turbo-code with the robustness of the reversible variable length codes (VLC), and moreover with a decoding complexity close (or less) to the one of the turbo-code alone. The performance due to the turbo-code alone leads to an increased freedom in the VLC design — depending on the application —, and suggests to revisit the VLC distance criteria investigated in [5].

The decoding algorithm for this three SISO modules scheme is briefly presented in the framework of Bayesian networks. Compared with previous JSC turbo-decoders and with a classical turbo-code, simulation results show the relevance and the superior performance of the proposed scheme.

## I. INTRODUCTION

In any efficient source coders, the variable length codes (VLC) are preferably used for their compression capabilities, as opposed to the fixed length codes. They are however sensitive to even a single binary error. Fortunately, the residual redundancy present in the VLCs can be exploited at the decoder so as to improve the robustness, especially if this redundancy is used iteratively with an error correcting code.

This last *joint source-channel* (JSC) turbo technique has been explored recently in different ways [3]–[6]. These ones can be classified in two groups, according to the source model into consideration and to the side information required to the decoding. Concerning the source model, it is usually either a memoryless source [4]–[6] or a first order Markov source [3]. Note that it is very easy to convert the decoder from one model to the other one by modifying (reducing or increasing) the state space. Concerning the side information, the decoding algorithm requires either the number of transmitted symbols or the number of bits [5], [6], or it requires both [3], [4].

All these JSC turbo techniques use only *two* soft-in/soft-out (SISO) modules in the iterative scheme: one module for the VLC decoding (combined with a bit-accumulator in [6]) and one for the decoding of a recursive systematic convolutional (RSC) code. Having such a scheme in mind, Bauer *et al.* in [5] studied the distance spectra of VLCs for a source of four symbols and showed that the free distance  $d_f$  strongly

Xavier Jaspas thanks the Belgian N.F.S. for its financial support. This work is partly funded by the Federal Office for Scientific, Technical and Cultural Affairs, Belgium, through IAP contract No P5/11.

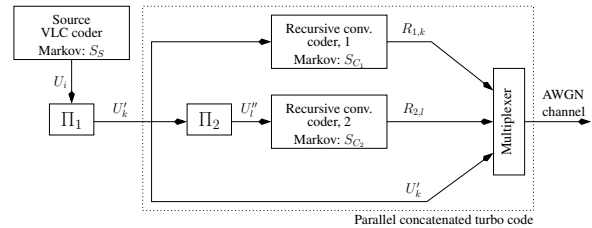


Fig. 1. Proposed VLC turbo coder.

determines the symbol error rate performance. In this context, [6], [7] provided a simple modification of the algorithm found in [11] to construct reversible VLCs (RVLCs) with  $d_f = 2$ .

In this paper, we propose a new JSC iterative decoder with *three* SISO modules for VLCs protected by a turbo-code (instead of one RSC code); the coder is illustrated in Fig. 1. The iterative behavior is intuitively different: in a sense, the turbo convergence can be ensured by only two of the three modules. This gives more freedom for the design of the constituent codes, and a fortiori for the design of the VLCs.

Section II describes the proposed transmission scheme. In section III, the appropriate iterative exchange between the three SISO modules is reported using Bayesian networks. The main characteristics of the source SISO module are given in section IV. Section V suggests to revisit the free distance criterion of VLCs, and introduces a new simple structure of redundancy for VLCs. In section VI, simulation results are reported for transmission over an additive white Gaussian noise (AWGN) channel. Compared to previous schemes, the three SISO modules decoder shows superior performance whatever VLC is used. Section VII concludes the paper.

## II. PROPOSED TRANSMISSION SCHEME

We propose to protect the VLCs sequence by means of the efficient parallel concatenated turbo-codes [2], as in Fig. 1. Each coder is modeled as a Markov chain:  $S_S$  for the source coder,  $S_{C,1}$  and  $S_{C,2}$  for the recursive systematic convolutional coders (RSCs). The Markov chain  $S_S$  produces the information bits  $U_i$ . These bits are then interleaved and protected by a turbo-code as suggested in the figure.

The choice of a turbo-code as error correcting code for the VLC sequence is motivated by two reasons. Firstly, in many practical situations the turbo-code combined with a hard detection of the VLC sequence seems to perform better than

JSC turbo-decoding schemes proposed in the literature [3]–[6]. The second reason is a consequence of the first one, and can be formulated as a question: can we still improve the turbo-code performance by using the VLC residual redundancy in the turbo-decoder?

### III. THREE SISO MODULES BAYESIAN NETWORK AND TURBO-DECODING

Let us first mention that it is possible to iteratively decode the three Markov chains  $S_S$ ,  $S_{C,1}$  and  $S_{C,2}$  with only two SISO modules if we suppress the first interleaver  $\Pi_1$  in Fig. 1. But this would result in an increased state space for the first SISO module — because  $S_S$  and  $S_{C,1}$  then have to be considered as a single more complex Markov chain [6]. This has a major drawback: practical complexity constraints lead to use either a very simple recursive coder  $S_{C,1}$  (to limit the state space) or a suboptimal decoding for the first SISO module. This is the main reason for investigating an iterative decoding of each constituent codes separately:  $S_S$ ,  $S_{C,1}$  and  $S_{C,2}$ , which results in a three SISO modules iterative decoding.

Consider the Bayesian network [12] given in Fig. 2 which corresponds to the proposed turbo-(de)coder. To each hidden Markov chain in the figure corresponds a SISO module. The task of a SISO module corresponds to the two-way schedule (BCJR algorithm) [12] applied to its Markov chain. The task of the turbo-decoder is exchanging information messages between the SISO modules. This exchange can be determined by a sort of flooding schedule [12] around each node  $U_i$ , and is reported in Tab. I, using the following notations:

$$\hat{\mathcal{Y}}, \hat{\mathcal{Y}}_A^{\setminus B} \quad (1)$$

- Given a Bayesian network, the set of all available measures (black vertices) is denoted  $\hat{\mathcal{Y}}$ .
- If removing the edges  $(A, B)$  and  $(A, C)$  separates the network into two subgraphs, then the set of measures on the subgraph containing the vertex  $A$  (resp.  $B, C$ ) is denoted  $\hat{\mathcal{Y}}_A^{\setminus B, C}$  (resp.  $\hat{\mathcal{Y}}_{B, C}^{\setminus A}$ ). Hence  $\hat{\mathcal{Y}} = \hat{\mathcal{Y}}_A^{\setminus B, C} \cup \hat{\mathcal{Y}}_{B, C}^{\setminus A}$ . This can be generalized straightforwardly to any number of removed edges.

This exchange completely describes the turbo-decoder, see [8] for details.

Surprisingly, the decoding complexity can be made lower than that of the turbo-code alone. In a few words, this is essentially because generally less iterations are required and because the source module is not needed at each iteration<sup>1</sup>. Analysis using EXIT charts will be reported in future work.

### IV. SOURCE SISO MODULE ALGORITHM

As mentioned above, the source SISO module is based on the application of the BCJR algorithm on the Markov chain  $S_S$  (for details, see [8]) which is here determined like the trellis introduced by Balakirsky in [1] — and used later in [5]–[7]. At the last iteration, a sequence MAP detection is applied.

<sup>1</sup>During the iterations without this one, the two other modules use the last output produced by the source module.

Note that this trellis is still valid in the case of a first order Markov source: one only has to add the last decoded symbol to the states of  $S_S$ , which increases the state space and the decoding complexity.

Concerning the side information, the decoder requires the number  $N$  of transmitted bits  $U_i$ . If the number  $K$  of transmitted symbols is also known to the decoder, the decoding complexity is roughly multiplied by  $K$ .

### V. VARIABLE LENGTH CODES REVISITED

VLCs offer good compression ratios but are more sensitive than fixed length codes to channel impairments. This is due to the variable boundaries between symbols and the corresponding bits. One can distinguish two types of errors. In the type I, an erroneous bit does not alter these boundaries: the bit error results in a local symbol error. In the type II, an error on one bit causes a modification of the symbol boundaries: the bit error desynchronizes the decoder, and symbol errors occur in burst. Intuitively, errors of type II are more dangerous for the source distortion.

Huffman VLCs for stationary sources are usually not protected against errors. They have in fact a free distance  $d_f = 1$  [7]. In the following, we propose to revisit the more robust reversible VLCs and we introduce a way to improve error resilience in the presence of an imperfect source model.

#### A. Reversible VLCs

Reversible VLCs (RVLCs) are both prefix and suffix codes: a code word is neither a prefix nor a suffix of a longer code word. These codes were introduced in [10], and an efficient suboptimal algorithm for their construction is given in [11]. In this work, we consider only asymmetric RVLCs.

RVLCs are protected against errors of type II [7] with a Hamming distance of 2 (between the exact and erroneous sequences). To improve their correcting capabilities, [7] gives a simple modification to the algorithm found in [11]. The obtained RVLCs are protected against both types of errors with a Hamming distance of two: their free distance is  $d_f = 2$ .

However, this free distance  $d_f = 2$  is obtained at the cost of an increased average length. In order to keep the overall code rate constant, the bit stream is thus protected by a weaker (e.g. more punctured) channel code. With the proposed scheme, depending on the application, this is not always the optimal solution as simulation results will show.

#### B. Least probable fixed - variable length codes

In a VLC, if code words with occurrence probability below a given threshold  $P_{\text{lpf}}$  are constrained to be of the same fixed length, then we call the resulting code a *least probable fixed - variable length code* (LPF-VLC).

These codes have the advantage to improve the synchronization, especially with an imperfect source model. The least (a priori) probable words could be indeed systematically rejected<sup>2</sup> by the source decoder when the channel is very

<sup>2</sup>Because when the channel is noisy, the measures obtained from the channel are not sufficiently “convincing”. Hence the most (a priori) probable words are preferably chosen.

TABLE I

EXCHANGED MESSAGES AT ITERATION  $j$  BETWEEN THE THREE MARKOV CHAINS AROUND EACH VERTEX  $U_i = U'_k = U''_l$ .

	receives from vertex $U_i = U'_k = U''_l$	gives to vertex $U_i = U'_k = U''_l$
$S_{C_1}$	$P_{[j]}(U'_k, \hat{\mathcal{Y}}_{U'_k}^{S_{C_1,k}}) = P(\hat{Y}_{U'_k}   U'_k) P_{[j-1]}(\hat{\mathcal{Y}}_{S_{C_2,l}}^{U''_l}   U''_l) P_{[j-1]}(U_i, \hat{\mathcal{Y}}_{S_{S,i-1}, S_{S,i}}^{U_i})$	$P_{[j]}(\hat{\mathcal{Y}}_{S_{C_1,k}}^{U'_k}   U'_k)$
$S_{C_2}$	$P_{[j]}(U''_l, \hat{\mathcal{Y}}_{U''_l}^{S_{C_2,l}}) = P(\hat{Y}_{U''_l}   U''_l) P_{[j]}(\hat{\mathcal{Y}}_{S_{C_1,k}}^{U'_k}   U'_k) P_{[j-1]}(U_i, \hat{\mathcal{Y}}_{S_{S,i-1}, S_{S,i}}^{U_i})$	$P_{[j]}(\hat{\mathcal{Y}}_{S_{C_2,l}}^{U''_l}   U''_l)$
$S_S$	$P_{[j]}(\hat{\mathcal{Y}}_{U_i}^{S_{S,i-1}, S_{S,i}}   U_i) = P(\hat{Y}_{U'_k}   U'_k) P_{[j]}(\hat{\mathcal{Y}}_{S_{C_1,k}}^{U'_k}   U'_k) P_{[j]}(\hat{\mathcal{Y}}_{S_{C_2,l}}^{U''_l}   U''_l)$	$P_{[j]}(U_i, \hat{\mathcal{Y}}_{S_{S,i-1}, S_{S,i}}^{U_i})$

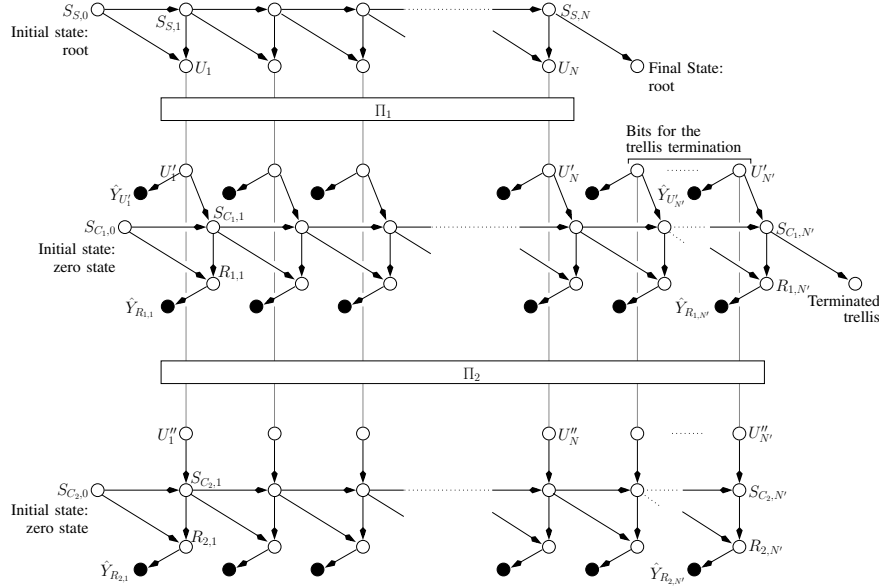


Fig. 2. Bayesian network of the proposed turbo-(de)coder. The vertices “final state: root” and “terminated trellis” are not random but deterministic.

noisy. This rejection is a source of desynchronization if words of different lengths are confused. Moreover, this problem becomes bigger if some a priori probabilities are wrongly too low — with an imperfect source model. With LPF codes, least probable words are generally more confused with words of the same length, which does not desynchronize the decoder. Thus, they help to fight type II errors.

### C. Combined LPF-RVLC

Of course, RVLCs and the LPF-approach can be combined. The threshold  $P_{\text{lpf}}$  is used to level the robustness. While adjusting this, we should not forget the trade-off in the coding rate allocation between the source coder and the channel coder to get the best overall performance.

### D. EXIT charts analysis and discussion

Despite the dependency of the performance on the source statistics, general trends can be observed. EXIT charts [13] are given in Fig. 3 for the different VLCs presented above: RVLCs with  $d_f = 2$ , LPF-RVLCs with  $d_f = 1$  and LPF-VLCs — all of them with roughly the same average length for a fair comparison. These EXIT charts give the mutual information between the  $U_i$  and the output messages  $P(U_i, \hat{\mathcal{Y}}_{S_{S,i-1}, S_{S,i}}^{U_i})$  for any given level of mutual information between the  $U_i$  and

the input messages  $P(\hat{\mathcal{Y}}_{U_i}^{S_{S,i-1}, S_{S,i}} | U_i)$ . Note that in this way, these charts do not have to be calculated for different signal to noise ratios on the channel.

The curve for the RVLCs with  $d_f = 2$  is the only one that reaches the point (1, 1), the error free point. Hence with a two SISO modules turbo-decoder or with a separate VLC decoder as in the previous works [5]–[7], this curve usually gives the best performance. This is why in these works,  $d_f = 2$  appeared to be a crucial criterion for soft VLC decoding.

With a three SISO modules turbo-decoder, things are different. The turbo convergence towards the point (1, 1) can be achieved by the other two SISO modules — the two RSCs in our case. The VLC design is thus less constrained: it can then be considered just as a helper in the turbo scheme. Depending on the application, the emphasis can be put e.g. on the better protection of a portion of the stream<sup>3</sup> — in the case of unequal error protection —, on the convergence at low or high levels of input mutual information, etc. . .

This is why the other curves can now be worthwhile with the proposed scheme. For example, Fig. 3 shows us the LPF-RVLCs with  $d_f = 1$  are less efficient than the RVLCs with

<sup>3</sup>without degrading the rest of the stream quality (because the efficient turbo-code still protects it).

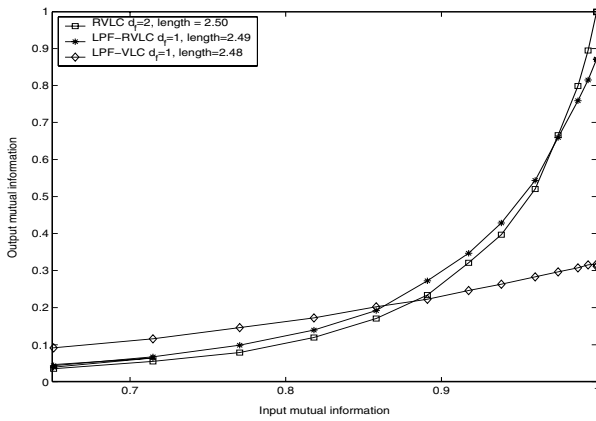


Fig. 3. EXIT charts of different VLCs for a source of 23 symbols with entropy 2.29. These VLCs have approximately the same average length.

$d_f = 2$  at correcting residual errors<sup>4</sup>. This is due to their incapacity to fight type I errors. *But*, because of the use of the LPF approach, they perform better when errors are more numerous. At last we see that, comparatively, the LPF-VLCs behave well in the presence of many errors but not at all with residual errors. This is explained on the one hand by a higher threshold  $P_{\text{lpf}}$  — which promotes synchronization —, and on the other hand by the lack of any bit-protection mechanism.

Let us remind that this discussion concerns just general trends: they are not always observed as precisely as noted above. Indeed everything depends on the statistical distribution of the source at stake.

## VI. SIMULATION RESULTS

The proposed three SISO modules turbo-decoder has been tested through simulations with sequences of independent symbols. For easy comparison with previous results, these symbols are those from the English alphabet as in [4], [6], [7], [10], [11]. To get significant results, all simulations were run until 60 erroneous decoded sequences were obtained for each  $E_b/N_0$ . The two RSCs are defined by the octal generators  $(G_r, G) = (037, 021)_8$ . These are punctured so as to obtain an overall code rate  $r_s r_c = 1/2$ , where  $r_s$  and  $r_c$  are respectively the source and the channel code rates.

Simulation results show in Fig. 4 and 5 the symbol error rate (SER) evaluated with the Levenshtein distance ( $\text{SER}_L$ ). The Levenshtein distance is more appropriate [5] than measures computed using a simple symbol comparison. In the figure, “hard” stands for hard VLC decoding — of course in this case, the VLC decoder is not used during the iterations<sup>5</sup>.

*Previous schemes, two SISO modules.* Fig. 4 shows us the performance of a JSC scheme with only one RSC — as found in the literature [3]–[5] —, and compares this one with the combination of the turbo-code and a hard VLC decoder. Clearly, as noted above, the turbo-code performs better whatever the VLCs are used in the previous JSC scheme.

<sup>4</sup>i.e. high level of input mutual information

<sup>5</sup>And if only one RSC is used, then there is no need to iterate.

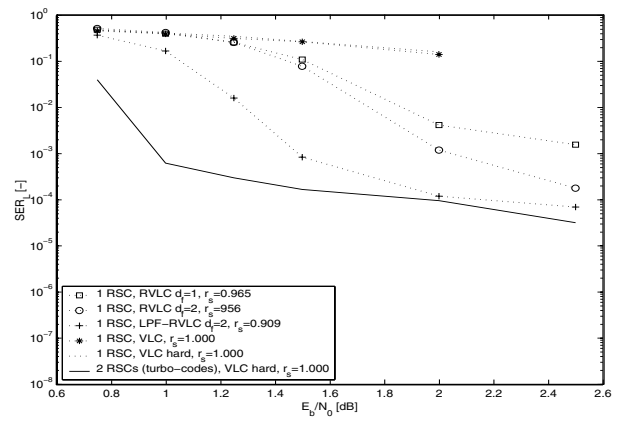


Fig. 4. Comparison of different VLCs with the scheme with only one RSC. Iteration 20. Sequences length  $N \leq 4000$  bits. Source symbols: English alphabet [10]. A relative code rate  $r_s$  of the source coder is given, which is fixed equal to one for the “VLC” curves.

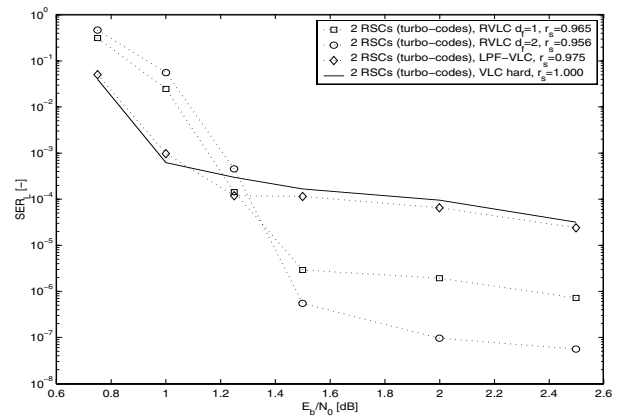


Fig. 5. Proposed scheme (JSC with a turbo-code). Parameters as Fig. 4.

*Proposed scheme, three SISO modules.* Now if we look at Fig. 5, the same turbo-code is highly outperformed by the proposed JSC scheme for  $E_b/N_0 \geq 1.25\text{dB}$ .

*VLCs with the previous schemes.* As “predicted” by the EXIT charts, the RVLCs with  $d_f = 2$  are those which bring the best performance when only one RSC is used (Fig. 4). Furthermore, combining these RVLCs with the new LPF approach provides a considerable gain in distortion, especially at low  $E_b/N_0$  where the LPF code words help fighting desynchronization (see section V-B).

*VLCs with the proposed scheme.* With the proposed scheme, the RVLCs with  $d_f = 2$  still perform better than other VLCs for high  $E_b/N_0$  ( $\geq 1.5\text{dB}$ , here). Now however, the RVLCs with  $d_f = 1$  behave well too. Actually, the system runs around  $\text{SER}_L = 10^{-6}$  with  $d_f = 1$  and around  $\text{SER}_L = 10^{-7}$  with  $d_f = 2$ . At those  $\text{SER}_L$ , we could presume the source distortion degradation is almost zero for most source coders. After all we can say that the only difference in performance between RVLCs with  $d_f = 1$  and  $d_f = 2$  occurs for  $E_b/N_0 \leq 1.25\text{dB}$  where the RVLCs with  $d_f = 1$  perform better — which is a novel result, consequence of the proposed scheme.

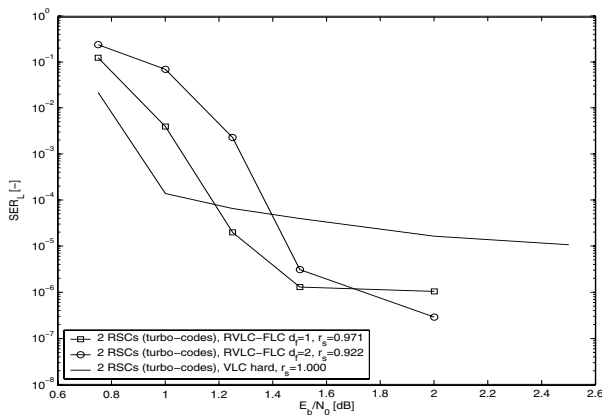


Fig. 6. Comparison of different VLCs with the proposed scheme. Iteration 20. Sequences length  $N \leq 4000$  bits. Source: 17 symbols, entropy 2.185.

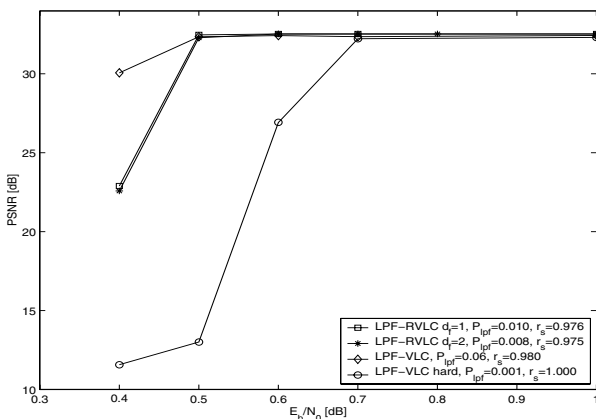


Fig. 7. PSNR for different  $E_b/N_0$ , iteration 20. Interleaver size: 65536. A relative code rate  $r_s$  of the source coder is given, which is fixed equal to one for the "LPF-VLC hard" curve.

This result is better brought into prominence in Fig. 6 where a different source of symbols is used.

In order to confirm these results with a real application, we also simulated (with the proposed scheme) the transfer of variable length coded DCT (discrete cosine transform)-compressed images [8], [9]. Tests were performed for several images, and are reported in this paper for the Lena image. The LPF-approach is always used — however sometimes with a very low  $P_{pr}$  threshold — because the DCT statistics model is imperfect [14]. Also markers are inserted in the bit stream to improve synchronization. The bits of the encoded image are split into several sequences if the interleavers are too small.

Fig. 7 shows the PSNR for different  $E_b/N_0$  values. The proposed JSC scheme provides a coding gain of 0.2dB in  $E_b/N_0$  over the classic turbo-code (the "hard" curve, a hard source decoder is used after the iterations). In this case — imperfect source DCT model used —, the LPF-VLCs offer the best performance at low  $E_b/N_0$ , even though such VLCs would have shown bad performance with the two SISO modules turbo-decoder found in the literature [3]–[5].

## VII. CONCLUSIONS

A new joint source-channel (JSC) turbo decoder, with three SISO modules, has been proposed for VLC and turbo encoded bit streams. The decoding algorithm is given briefly in the framework of Bayesian networks. Simulation results have shown the superior performance of the proposed JSC scheme, compared with the classic turbo-code alone and the previous JSC schemes found in the literature.

Concerning the source codec, this work revisits the RVLCs with free distance 1 and 2. It also provides a new kind of structure (structured redundancy) for VLCs, i.e. the least probable fixed (LPF) code words, which considerably improves the quality in the case of an imperfect source model. Moreover, this LPF approach can be combined with RVLCs and this combination provides a considerable gain in some cases.

Performance bounds and three dimensional EXIT charts analysis will be reported in future work. Further research will focus mainly on spectral and synchronizing properties of VLCs, and on the influence of an imperfect model.

## REFERENCES

- [1] Balakirsky V.B., "Joint source-channel coding with variable length codes", in *Proc. IEEE Int. Symposium on Information Theory*, Ulm, Germany, pp. 491, Jul. 1997.
- [2] Berrou C. and Glavieux A., "Near Optimum Error Correcting Coding And Decoding: Turbo-Codes", *IEEE Trans. on Communications*, vol. 44, no. 10, pp. 1261-1271, Oct. 1996.
- [3] Guyader A., Fabre E., Guillemot C., and Robert M., "Joint Source-Channel Turbo Decoding of Entropy-Coded Sources", *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1680-1696, Sept. 2001.
- [4] Bauer R. and Hagenauer J., "Iterative Source/Channel-Decoding Using Reversible Variable Length Codes", *IEEE Proc. Data Compression Conference (DCC)*, pp. 93-102, Snowbird, USA, Mars 2000.
- [5] Bauer R. and Hagenauer J., "On variable length codes for iterative source/channel decoding", *IEEE Proc. Data Compression Conference (DCC)*, pp. 273-282, Snowbird, USA, March 2001.
- [6] Laković and Villasenor J., "Combining Variable Length Codes and Turbo Codes", *Proceedings IEEE 55th Vehicular Technology Conference (VTC)*, vol. 4, pp. 1719-1723, Birmingham, AL, USA, Mai 2002.
- [7] Laković and Villasenor J., "On Design of Error-Correcting Reversible Variable Length Codes", *IEEE Communications Letters*, vol. 6, no. 8, pp. 338-339, Aug. 2002.
- [8] Jaspas X. and Vandendorpe L., "Three SISO Modules Joint Source-Channel Turbo-Decoding of Variable Length Coded Images", *SCC04 - 5th international ITG conference on Source and Channel Coding*, Erlangen, Germany, pp. 279-286, Jan. 2004.
- [9] Jaspas X., "Turbo-décodage source-canal conjoints", *Master thesis*, Commun. and Remote Sensing Lab., Université Catholique de Louvain (UCL), Belgium, June 2003.
- [10] Takishima Y., Wada M. and Murakami H., "Reversible Variable Length Codes", *IEEE Trans. on Communications*, vol. 43, no. 2/3/4, pp. 158-162, Feb.-April 1995.
- [11] Tsai C.-W. and Wu J.-L., "On Constructing the Huffman-Code-Based Reversible Variable-Length Codes", *IEEE Trans. on Communications*, vol. 49, no. 9, pp. 1506-1509, Sept. 2001.
- [12] Kschischang Frank R. and Frey Brendan J., "Iterative Decoding of Compound Codes by Propability Propagation in Graphical Models", *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 1, pp. 219-230, Jan. 1998
- [13] ten Brink S., "Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes", *IEEE Trans. on Communications*, vol. 49, no. 10, pp. 1727-1737, Oct. 2001.
- [14] Lam E.Y., and Goodman J.W., "A Mathematical Analysis of the DCT Coefficients Distributions for Images", *IEEE Trans. on Image Proc.*, vol. 9, no. 10, pp. 1661-1666, Oct. 2000.