Error-resilient and progressive transmission and decoding of compressed images

Hervé Jégou†, Thomas Guionnet* and Christine Guillemot*
†IRISA/ENS Cachan, *IRISA/INRIA,
Campus universitaire de Beaulieu, 35042 Rennes, France
{*Herve.Jegou,Thomas.Guionnet,Christine.Guillemot*}*@irisa.fr*

ABSTRACT-

This paper describes an approach for structuring and transmitting a sequence of variable length codewords in order to allow for a robust and progressive transmission over noisy channels. The dependencies within the resulting bitstream can be easily modeled with a stochastic automaton leading naturally to efficient soft decoding techniques with a complexity comparable to the complexity obtained with classical bitstream structures. Simulation results reveal, with both theoretical sources and real images, high performances in terms of symbol error rates and mean square reconstruction error at no cost in compression efficiency with both hard and soft decoding.

1. Introduction

Entropy coding, producing variable length codewords, is a core component of any data compression scheme. The main drawback of VLCs is their high sensitivity to channel noise: when some bits are altered by the channel, synchronization losses can occur at the receiver, the position of symbol boundaries are not properly estimated, leading to dramatic symbol error rates. This phenomenon has motivated studies of the synchronization capability of VLCs as well as the design of codes with better synchronization properties [1]. Reversible VLCs [2][3], have also been designed to fight against de-synchronizations. Soft VLC decoding ideas, exploiting residual source redundancy have also been shown to reduce the "de-synchronization" effect hence decrease the residual symbol error rates [4][5][6][7].

For a given number of source symbols, the number of bits produced by a VLC coder is a random variable. The decoding problem is then to properly segment the noisy bitstream into measures on symbols and estimate the symbols from the noisy sequence of bits (or measurements) that is received. This segmentation problem can be addressed by introducing a-priori information in the bitstream, taking often the form of synchronization patterns. This a-priori information is then exploited as constraints by the decoding process. One can alternately, by a proper structuring of the bitstream, reveal and exploit constraints on some bit positions. This idea is applied in [8] to blocks within an image. A structure of fixed length size slots inherently creates hard synchronization points in the bitstream. The principle can be applied directly on a symbol (hence codeword) basis. The principle can however be pushed further in order to optimize criteria of resilience, computing complexity and progressivity.

In this paper, given a VLC, we focus on the design of a transmission scheme of the variable length codewords in order to achieve low SER (symbol error rate) and high SNR performances in presence of transmission errors. The process for constructing the bitstream is regarded as a dynamic bit mapping φ between an intermediate binary representation $\mathbf{B}$ of the sequence of symbols and the bitstream $\mathbf{E}$ that will be transmitted on the channel. The intermediate representation is obtained by the codeword assignment to the different symbols. The decoder proceeds similarly with a bit mapping ψ, which in presence of transmission noise, may not be the inverse of the mapping φ. Maximum error resilience is achieved when the highest number of bit mappings are deterministic or have the highest confidence measure. A so-called *constant* mapping defining some hard synchronization points in the bitstream, which are independent of the source realization, is described. The bitstream structure allows for significant improvements in terms of SER and SNR performances with respect to classical transmission scheme where the variable length codewords are concatenated even with hard decoding techniques. The performances can be further improved by using soft decoding algorithms, e.g. by using the BCJR [9] algorithm with MPM (Maximum of Posterior Marginals) or MMSE (Minimum Mean square Error) criteria when targeting respectively SER and SNR performances. The bitstream structure is also amenable to progressive decoding using either a decoding method computing the expectation of the symbols given the set of received bits or the BCJR decoding algorithm. When the BCJR algorithm is used, the probabilities of the transitions on the trellis that correspond to non-received bits are simply set to the posterior marginals of the corresponding bits which turn out to be close to $\frac{1}{2}$[1]. In other words, the bits are considered as non informative. The results with both theoretical sources and real images show that the bitstream structure allows for significant SER

[1]The posterior marginals of the bits will be equal to $\frac{1}{2}$ if the minimum description length (mdl) of the VLC is very close to the entropy of the source

and SNR improvement in presence of transmission errors with respect to classical bitstream structures. The soft-decision decoding results turn out also to be higher than those obtained in [4], with the same decoding complexity.

2. Problem statement and Notations

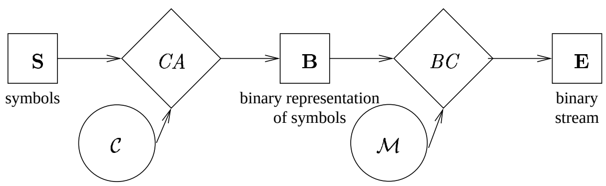


Fig. 1. Decomposition of VLC coding process.

Let $\mathbf{S} = (S_1, \ldots S_t, \ldots S_K)$ be a sequence of source symbols taking their values in a finite alphabet $\mathcal{A}$ composed of Ω symbols, $\mathcal{A} = \{a_1, \ldots a_i, \ldots a_\Omega\}$. Let $\mathcal{C}$=$\{c_1, \ldots, c_i, \ldots c_\Omega\}$ be a binary variable length code designed for this alphabet, according to its stationary probability $\boldsymbol{\mu}$. The length of the codeword c_i is denoted l_i and its l^{th} bit is denoted $c_i(l)$. The sequence of symbols $\mathbf{S}$ is first converted into a sequence of bits, referred to as the intermediate representation,

$$\mathbf{B} = \left(\begin{array}{ccccc} B_1^1 & & & & B_K^1 \\ \vdots & ; & \ldots & ; & \vdots \\ B_1^{L(S_1)} & & & & B_K^{L(S_K)} \end{array} \right). \qquad (1)$$

The term $L(S_K)$ represents the length of the codeword associated to the realization s_k of the symbol S_k. In classical compression systems, the sequence of codewords produced would be transmitted *sequentially*, forming a *concatenated* bitstream. In the sequel, the emitted bitstream is denoted $\mathbf{E} = E_1 \ldots E_{K_E}$ and the received sequence of noisy bits is denoted $\hat{\mathbf{E}} = \hat{E}_1 \ldots \hat{E}_{K_E}$. Similarly, the decoded noisy intermediate representation is referred to as $\hat{\mathbf{B}}$.

We consider the general framework depicted in Fig. 1, where the coding process is decomposed into two steps: codeword assignment (*CA*) and bitstream construction (*BC*). The algorithm which constructs the bitstream $\mathbf{E}$ from the intermediate binary representation $\mathbf{B}$ of the sequence of symbols is denoted $\mathcal{M}$. Note that the length K_E of the constructed bitstream $\mathbf{E}$, which does not depend on the algorithm used, is assumed to be known. Assuming that the first step has been performed ($\mathcal{C}$ is supposed to be known), we focus on the problem of designing algorithms $\mathcal{M}$ that will satisfy various properties of resiliency and progressivity.

3. Notion of mapping

The algorithm $\mathcal{M}$ can be regarded as a dynamic bit *mapping* between $\mathbf{B}$ and $\mathbf{E}$. This mapping, referred to as φ, depends on the realization $\mathbf{b}$ of $\mathbf{B}$, and is defined on the set $\mathcal{I}$ of tuples (t, l) that parses $\mathbf{B}$, denoted $\mathcal{I}(\mathbf{B}) = \{(t, l) \;/\; 1 \leq t \leq K,\; 1 \leq l \leq L(S_t)\}$, as

$$\begin{array}{ccc} \mathcal{I}(\mathbf{B}) & \rightarrow & [1..K_E] \\ (t, l) & \rightarrow & \varphi(t, l) = n \end{array} \qquad (2)$$

Similarly, the decoding algorithm proceeds with a bit mapping between the received bitstream $\hat{\mathbf{E}}$ and an intermediate representation $\hat{\mathbf{B}}$ of the received sequence of codewords. This mapping, referred to as ψ, depends on the noisy realization $\hat{\mathbf{b}}$ of $\hat{\mathbf{B}}$ and is defined as

$$\begin{array}{ccc} [1..K_E] & \rightarrow & \mathcal{I}(\hat{\mathbf{B}}) \\ n & \rightarrow & \psi(n) = (t, l) \end{array} \qquad (3)$$

where the set $\mathcal{I}(\hat{\mathbf{B}})$, in presence of bit errors, may not be equal to $\mathcal{I}(\mathbf{B})$. The composed function $\pi = \psi \, o \, \varphi$ is a dynamic mapping function from $\mathcal{I}(\mathbf{B})$ into $\mathcal{I}(\hat{\mathbf{B}})$. An element is decoded in the correct position if and only if $\pi(\,(t, l)\,) = (t, l)$.

3.1. Notion of constant submapping

The bitstream can be seen as constructed from the intermediate binary representation by using a set of mapping functions. The error resilience depends on the capability, in presence of channel errors, to map a bitstream element of $\hat{\mathbf{E}}$ to the correct position in the decoded intermediate representation.

Definition 1: an element index (t, l) is said to be *constant* by $\pi = \psi \, o \, \varphi$, iff $n = \varphi(t, l)$ does not depend on the realization $\mathbf{b}$. Similarly, the bitstream index n is also said to be *constant*.

Let $\mathcal{I}_C$ denote the set of *constant* indexes. The restriction φ_C of φ to this subset and its inverse $\psi_C = \varphi_C^{-1}$ are also said to be *constant*. Such constant submappings can not be altered by channel noise: $\forall\, \hat{\mathbf{b}},\ (t, l) \in \mathcal{I}_C \Rightarrow \pi(t, l) = (t, l)$. Let $h_{\mathcal{C}}^-$ denote the length of the shortest codeword of the codetree. Since for each realization $\mathbf{b}$, $\mathcal{I}_C \subseteq \mathcal{I}(\mathbf{b})$, we deduce that if π_C is a constant submapping, then $\mathcal{I}_C \subseteq [1..K] \times [1..h_{\mathcal{C}}^-]$.

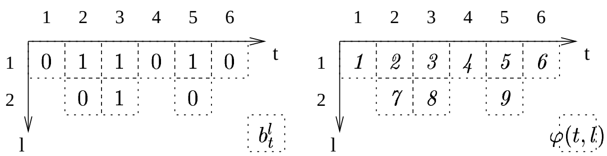
3.2. Construction of constant submappings

We focus here on the design of error resilient transmission schemes, i.e. on the identification of a submapping π_C of the mapping π that would be *constant*, i.e., such that $\mathcal{I}_C = [1..K] \times [1..h_{\mathcal{C}}^-]$. Given a code $\mathcal{C}$, the *constant* submapping defines a set of “hard” synchronization points. Notice first that a variable length codetree comprises a section of a fixed length equal to the minimum length of a codeword denoted $h_{\mathcal{C}}^-$, followed by a variable length section. A *constant* mapping can thus be defined as the composition of functions $\varphi_C : [1..K] \times [1..h_{\mathcal{C}}^-] \rightarrow [1..K h_{\mathcal{C}}^-]$ and $\psi_C = \varphi_C^{-1}$ defined such that

$$(t, l) \rightarrow \varphi_C(t, l) = (l - 1)K + t \qquad (4)$$

The bits which do not belong to the definition set of this constant submapping are then simply concatenated.

Example: let $\mathcal{A}$=$\{a_1, a_2, a_3\}$ be the alphabet of the source $\hat{S}$ with the stationary probabilities given by μ_1=0.50, μ_2=0.25 and μ_3=0.25. The corresponding Huffman code is $\mathcal{C}_1$=$\{0, 10, 11\}$ and to the source realization $\hat{s} = 0\,1\,2\,0\,1\,0$, one can associate the intermediate representation and mapping φ: which finally lead to the emitted bitstream



$\mathbf{e}$=011010010. Note that the set $\mathcal{I}_C$ of constant indexes is the set $[1; 6] \times [1; 1]$.

In contrast with classical transmission schemes where the codewords are concatenated, error propagation will only take place on the tuples (t, l) which do not belong to $\mathcal{I}_C$. The above mapping amounts to transmit the fixed length section of the codewords bit plane per bit plane. Hence, for a Huffman tree, the most frequent symbols will not suffer from de-synchronization. This layered bitstream structure is well suited for unequal error protection. The code $\mathcal{C}$ should hence be designed so that most of the signal energy is concentrated on the fixed length section of the codetree. In contrast with EREC, encoding and hard decoding can be processed with linear complexity in $O(K)$ where K is the length of the sequence of symbols. Moreover, the bitstream structure is amenable to soft-decision decoding using Bayesian estimation principles: the state model proposed in [10] can be applied with few modifications, as we will see in section 4.

4. SOFT-DECISION DECODING

Trellis-based soft-decision decoding techniques making use of Bayesian estimators can be used to further improve the decoding SER and SNR performances. Assuming that the sequence $\hat{S}$ can be modeled as a Markov process, MAP, MPM or MMSE estimators, using e.g. the BCJR algorithm [9], can be run on the trellis representation of this source model [6]. Every path of this trellis corresponds to a sequence that the source could have emitted. This trellis is indexed by symbol clock t and a clock n' that denotes the number of bits *used to encode the* t *first symbols*. Thus, $n' = \sum_{t'=1}^{t} l(s_{t'})$. Notice that, if the *BC* is the concatenation of codewords, then n'=n, where n is the current bit position in the bitstream $\mathbf{e}$. Using a *BC* that relies on a constant mapping, as proposed in 3-2, we have to distinguish those values and their respective random variables N'_t and N_t.

In order to take into account the Markovian property of the source, we have several states for a given couple (t, n'). Thus, for a given symbol clock t, a state of the trellis is fully indexed by the couple (a_i, n'). The BCJR algorithm proceeds with the estimation of the probabilities $\mathbb{P}(S_t = a_i | \hat{E}_1; \ldots \hat{E}_{K_E})$, knowing

- the Markov source transitions probabilities, i.e., $\nu_{i,i'} = \mathbb{P}(S_t = a_i | S_{t-1} = a_{i'})$,
- The channel transition probabilities $\mathbb{P}(\hat{E}_n = \hat{e}_n | E_n = e_n)$, assumed to follow a discrete memoryless channel (DMC) model.

Using similar notations as in [9], the estimation proceeds with forward and backward recursive computations of the quantities

$$\begin{aligned} \alpha_t(a_i, n') = \; & \mathbb{P}(S_t = a_i; N'_t = n'; (\hat{e}_{\varphi(t',l)})), \\ & \text{with } 1 \leq t' \leq t, 1 \leq l \leq L(s_{t'}) \end{aligned}$$

$$\begin{aligned} \beta_t(a_i, n') = \; & \mathbb{P}((\hat{e}_{\varphi(t',l)}) | S_t = a_i; N'_t = n'), \\ & \text{with } t+1 \leq t' \leq K, 1 \leq l \leq L(s_{t'}) \end{aligned}$$

$$\begin{aligned} \gamma_t(a_{i_2}, n'_2, a_{i_1}, n'_1) = \; & \mathbb{P}(S_t = a_{i_1}; N'_t = n'_1; (\hat{e}_{\varphi(t,l)})_{1 \leq l \leq l_{i_1}} | \\ & S_{t-1} = a_{i_2}; N'_{t-1} = n'_2) \\ = \; & \delta_{n_1 - n_2 - l_{i_1}} \nu_{i_1, i_2} \prod_{l=1}^{l_{i_1}} p(\hat{e}_{\varphi(t,l)} | c_{i_1}(l)) \end{aligned} \quad (5)$$

Notice that only the processing of γ is modified by the choice of the constant mapping instead of the concatenation of codewords. Thus, for the concatenated mapping, we have $\varphi(t, l) = n'_2 + i$. For the constant mapping, we have

$$\begin{aligned} \varphi(t, l) &= (l-1) * K + t & \text{if } l <= h_{\mathcal{C}}^- \\ &= (K - t) h_{\mathcal{C}}^- + n'_2 + l & \text{otherwise} \end{aligned} \quad (6)$$

The quantities $\alpha_t(a_i, n')$ and $\beta_t(a_i, n')$ being computed, the BCJR algorithm can then proceed with the estimation of $\mathbb{P}(N'_t = n'; S_t = a_i) \propto \lambda_t(a_i, n') = \alpha_t(a_i, n')\, \beta_t(a_i, n')$ and the subsequent marginal probabilities $\mathbb{P}(S_t = a_i)$. The MPM and MMSE estimators $\tilde{S}_t$ and $\overline{S_t}$ are then derived as

$$\tilde{S}_t = arg \max_{a_i \in \mathcal{A}} \mathbb{P}(S_t = a_i) \quad (7)$$

$$\overline{S_t} = \sum_{a_i \in \mathcal{A}} a_i \mathbb{P}(S_t = a_i) \quad (8)$$

5. PROGRESSIVE DECODING

VLC codewords can be decoded progressively by regarding the bit generated by the transitions at a given level of the codetree as a bit-plane or a layer. Let us assume that the first l bits of a codeword have been received. They correspond to an internal node n_j on the codetree. Let $\mathcal{L}_j$ and $\tilde{\mu_j} = \sum_{n_i \in \mathcal{L}_j} \mu_i$ respectively denote the leaves deduced from n_j and the probability associated to the node n_j. Then the reconstruction value $\tilde{s}_j$ is given by

$$\tilde{s}_j = \frac{1}{\tilde{\mu_j}} \sum_{n_i \in \mathcal{L}_j} \mu_i \, s_i \quad (9)$$

The mean square error associated to this reconstruction is given by the variance of the source knowing the first bits.

The BCJR can also be applied on the truncated bitstream to compute an MMSE estimation of the received sequence, by setting the transitions on the trellis that correspond to the non-received bits to their posterior marginals or to an approximated value of $\frac{1}{2}$.

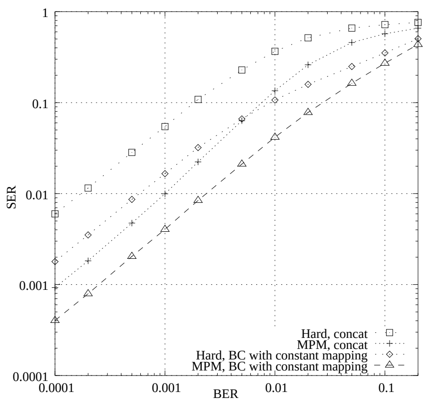


Fig. 2. Respective SER performance of bitstream structure with hard and MPM decoding.

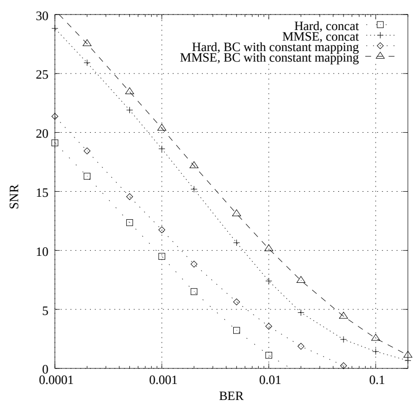


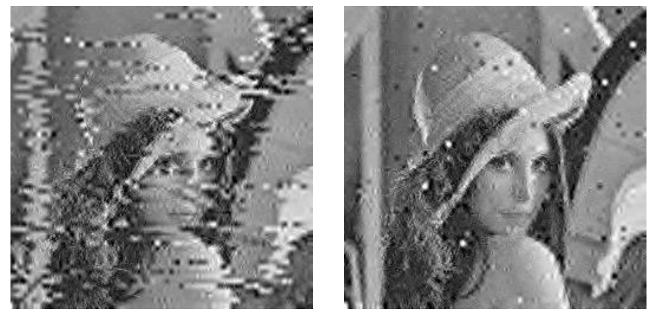
Fig. 3. Respective SNR performance of bitstream structure with hard and MMSE decoding.

6. SIMULATION RESULTS

The performance of the bitstream construction algorithm has been evaluated both in terms of SER and SNR with a normal distributed source of correlation $\rho = 0.5$, uniformly quantized on 8 values between -3 and +3. We have considered sequences of 100 symbols. The results are averaged over 10000 *significant* channel and source realizations. Fig. 2 and Fig. 3 respectively show the SER and the SNR obtained with the bitstream construction in comparison with the concatenated structure, for channel error rates going from 10^{-4} to 2.10^{-1}. Note that the Huffman code used in the simulations is not optimal for MMSE reconstruction: SNR Performances can be further improved if the code is designed such that the bit transitions mapped in deterministic positions bear most of the signal energy.

7. APPLICATION TO IMAGE TRANSMISSION OVER NOISY CHANNELS

The approach has been experimented with real sources. We have considered a simple image coding system where the image is first decomposed into 10 subbands using a three-stage wavelet transform. The low and high frequency subbands have been quantized with a uniform quantizer. Hard synchronization points are provided every 4096 samples. We have used a Huffman code, optimized so that



concatenation (17.23 dB) constant mapping (22.28 dB)

Fig. 4. Image transmission scheme on BSC with BER=0.005, hard decoding.

most of the signal energy is located in the fixed part of the codetree. Fig. 4 illustrates the visual improvement with respect to classical bitstream structures (hard decoding has been used in this test). The impact of potential desynchronizations is limited to bits bearing a low amount of signal energy.

8. CONCLUSION

In this paper, we have described a bitstream construction method allowing for error-resilient and progressive transmission of VLC-encoded sources over noisy channels. Note that the bitstream construction directly applies to reversible VLC while preserving the forward/backward decoding capability. Dependencies within the bitstream can be easily modelled by a stochastic automaton on which Bayesian estimation techniques apply directly, leading to high SER and SNR performances.

REFERENCES

[1] T. Ferguson and J. H. Rabinowitz, "Self-synchronizing huffman codes," *IEEE Trans. Inform. Theory*, vol. IT-30, no. 4, pp. 687–693, July 1984.

[2] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 158–162, Feb. 1995.

[3] J. Wen and J. D. Villasenor, "Reversible variable length codes for efficient and robust image and video coding," in *Proc. IEEE DCC'98*, Apr. 1998, pp. 471–480. [Online]. Available: citeseer.nj.nec.com/wen98reversible.html

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

[5] N. Demir and K. Sayood, "Joint source-channel coding for variable length codes," in *Proc. IEEE DCC'98*, Mar. 1998, pp. 139–148.

[6] R. Bauer and J. Hagenauer, "Turbo fec/vlc decoding and its application to text compression," in *Proc. Conf. Inform. Theory and Systems*, Mar. 2000, pp. WA6.6–WA6.11.

[7] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, "Joint source-channel turbo decoding of entropy coded sources," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1680–1696, Sept. 2001.

[8] D. Redmill and N. Kingsbury, "The erec: An error resilient technique for coding variable-length blocks of data," *IEEE Trans. Image Processing*, vol. 5, pp. 565–574, Apr. 1996.

[9] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, pp. 284–287, Mar. 1974.

[10] R. Bauer and J. Hagenauer, "Symbol by symbol map decoding of variable length codes," in *Proc. 3rd ITG Conf. Source and Channel Coding, Munich, Germany, Jan. 17.-19. 2000*, 2000, pp. 111–116. [Online]. Available: citeseer.nj.nec.com/bauer00symbolbysymbol.html