Lower bounds on the existence of binary error-correcting variable-length codes

Catherine Lamy[1]
Philips Recherche France, Suresnes, France
lamy@ieee.org

François-Xavier Bergot
Paris, France
francois-xavier.bergot@enst.fr

Abstract — **This paper deals with codes that combine source and channel encoding operations. After discussing the potential usefulness of these variable-length error correcting (VLEC) codes, necessary conditions on their length distribution are established. It is shown that, depending on the targetted application, several families of VLEC codes can be defined, whose conditions of existence differ.**

Index Terms — **variable-length error-correcting codes, variable-length codes, error correction coding, joint source-channel coding, bounding methods**

I. Introduction

Traditionally, the two antagonistic encoding operations of compression and error correction are separated from each other, following Shannon's well-known *separation* theorem [1] which states that source coding and channel coding can, asymptotically with the length of the source data, be designed separately without any loss of performance for the overall system. For instance, variable-length codes (VLC), which are classically used in source coding for their compression capabilities, are often associated with Forward Error Correction (FEC) techniques [2] which combat the effects of a real transmission channel (fading, noise, ...). It has however been shown that separation does not necessarily lead to the less complex solution [3], nor is always applicable [4]. In particular, joint source and channel coding (JSCC) can offer better solutions in wireless communications, offering a complexity reduction of the overall system while maintaining a satisfactory performance. Quite recently, JSCC techniques that include a co-ordination between source and channel encoders were investigated, and techniques were developed that improve both encoding and decoding processes while keeping the overall complexity at an acceptable level [5]. Although these methods rely on a collaboration between both coders, they still keep the encoders and decoders separated. Another approach consists in using codes that offers both good compression and error correction capabilities [6].

We explore in this paper the possibility of performing both compression and error correction in a single coding step. Such an error correcting and compressing process is done here using codewords of varying lengths that have particular distance properties with each other, which allows to enable error detection and correction. The codes will hence be called Variable-Length Error Correcting (VLEC) codes in the following. In this contribution, the goal is set to establish existence conditions for the different considered families of VLEC codes.

The paper is organised as follows. The potential usefulness of error detecting/correcting and compressing codes is discussed in Section II and illustrated with an example. Known families of VLEC codes are reviewed and a new one is introduced in Section III. Generalisations of the Kraft-MacMillan inequality are then proposed in Section IV for two families of variable-length error correcting codes. Finally, some conclusions are drawn.

II. Motivation for error detection/correction and compression coding

Although separated coding has long been the most considered one in the literature and research community, it is well-known that a joint approach provides potential benefits over a traditional two-step approach. Let us consider for instance the two reversible variable-length codes (RVLC) $\mathcal{C}_1$, which is a truncated Golomb-Rice code [7], and $\mathcal{C}_2$ given in Table 1. While from a compression point of view, these two codes are equivalent as they offer the same compression rate $R = 18/7 \simeq 2.57$ bits/symbol, it is easy to see that all codewords of identical length of $\mathcal{C}_2$ differ at least by two bits. As a consequence, $\mathcal{C}_2$ offers better performance than $\mathcal{C}_1$. For instance, simulation results show a gain greater than 2 dB at bit error rate $P_{eb} = 10^{-4}$ over an additive white Gaussian noise (AWGN) channel with binary phase shift keying (BPSK) modulation and soft-input VLC decoding [8].

		$\mathcal{C}_1$ [7]	$\mathcal{C}_2$
u	$P(u)$	$\mathbf{c}^{(1)}(u)$	$\mathbf{c}^{(2)}(u)$
A	4/14	00	00
B	4/14	01	11
C	2/14	110	101
D	2/14	111	010
E	1/14	1010	1001
F	1/14	1011	0110
R (bits/symb)		18/7	18/7

Table 1: Two different RVLC designs.

It is also to be noted that efficient VLEC codes, once found, can easily be used in practical applications. As a matter of fact, the VLEC encoder can be trivially implemented using a lookup table and the VLEC decoding can be carried out either by an adaptation of the classical VLC hard decoding with searching for a match between the input bit strings and the codeword list, this relying on distance computations, or by an efficient and low-complexity soft-input decoder such as the one presented in [8].

Thus far however, no algorithm or construction has been discovered that finds the optimal VLEC code for a particular input data set without relying on some sort of exhaustive computer screening, whatever clever this one may be. In the absence of construction algorithm such as the Huffman one [9]

[1]C. Lamy is now with THALES Communications France, Gennevilliers, France (catherine.lamy@fr.thalesgroup.com).

to build VLEC codes, it is desirable to have an analytical approach that will allow to test any integer list representing codeword lengths to check if they could be the codeword lengths of a valid VLEC code. Such an analytical approach could for instance be used to direct and speed up a code generation algorithm such as the “Heuristic” method proposed by Buttigieg in [6] for a given family of VLEC codes (see Section III-B).

III. Codes combining error correction and compression capabilities

Several approaches have been introduced recently in [6, 10], to perform combined error correction and compression. These schemes consider variable-length codes for source coding and respect a minimum distance between each codeword for protection against channel errors. Let us introduce some notations and definitions to classify the families of VLEC codes.

A. Definitions and notations

Let $\mathcal{C}$ be a uniquely decodable [11] variable-length code of cardinality $\mathcal{N}_\mathcal{C}$. Let $|\mathbf{x}^i|$ denotes the length of each word $\mathbf{x}^i$ of $\mathcal{C}$. Let σ be the number of different word lengths in $\mathcal{C}$. We denote these different lengths by $L_1, L_2, \ldots, L_\sigma$, where $L_1 < L_2 < \ldots < L_\sigma$, and the corresponding number of codewords by $n_1, n_2, \ldots, n_\sigma$, where $\sum_{i=1}^{\sigma} n_i = \mathcal{N}_\mathcal{C}$.

Definition 1 *Let* $\mathbf{x} = (x_1, \ldots, x_{|\mathbf{x}|})$ *be a word of* $\mathcal{C}$. *A sequence* $(x_k, \ldots x_\ell)$, $1 \leq k \leq \ell \leq |\mathbf{x}|$ *is a* sub-word of $\mathbf{x}$.

Definition 2 *Let* $f_i = \mathbf{x}^{i_1}\mathbf{x}^{i_2}\cdots\mathbf{x}^{i_n}$ *be a concatenation of* n *words of* $\mathcal{C}$. *The set* $F_N = \{f_i : |f_i| = N\}$ *is called the* extended code *of* $\mathcal{C}$ *of order N.*

Definition 3 *The* Hamming weight $w(\mathbf{x})$ *of a word* $\mathbf{x}$ *is the number of non-zero symbols in* $\mathbf{x}$. *The* Hamming distance $h(\mathbf{x}^i, \mathbf{x}^j)$ *between two words* $\mathbf{x}^i$ *and* $\mathbf{x}^j$ *of equal length is the number of positions in which* $\mathbf{x}^i$ *and* $\mathbf{x}^j$ *differ.*

Definition 4 *The* sliding distance $s(\mathbf{x}^i, \mathbf{x}^j)$ *between two codewords* $\mathbf{x}^i$ *and* $\mathbf{x}^j$ *of lengths* $|\mathbf{x}^i|$ *and* $|\mathbf{x}^j|$ *of a code* $\mathcal{C}$ *is defined as the minimum of the Hamming distances between the* ℓ*-length codeword and every sub-word of length* ℓ *of the other codeword, with* $\ell = \min\{|\mathbf{x}^i|, |\mathbf{x}^j|\}$.

$$s(\mathbf{x}^i, \mathbf{x}^j) = \min_{\substack{0 \leq u \leq |\mathbf{x}^i| - \ell \\ 0 \leq v \leq |\mathbf{x}^j| - \ell}} \left\{ h(x^i_{u+1}\cdots x^i_{u+\ell}, x^j_{v+1}\cdots x^j_{v+\ell}) \right\}.$$

The minimum sliding distance $s_{\min}$ *of a code* $\mathcal{C}$ *is the minimum value for all sliding distances between every possible pair of codewords in* $\mathcal{C}$: $s_{\min} = \min_{(\mathbf{x}^i,\mathbf{x}^j)\in\mathcal{C}, i\neq j} \left\{ s(\mathbf{x}^i, \mathbf{x}^j) \right\}$.

Definition 5 *The* minimum block distance b_k *associated to the codeword length* L_k *of a code* $\mathcal{C}$ *is defined as the minimum Hamming distance between all distinct codewords of* $\mathcal{C}$ *with length* L_k.

$$b_k = \min_{\substack{(\mathbf{x}^i,\mathbf{x}^j) \in \mathcal{C},\, i \neq j \\ |\mathbf{x}^i| = |\mathbf{x}^j| = L_k}} \left\{ h(\mathbf{x}^i, \mathbf{x}^j) \right\}.$$

The overall minimum block distance $b_{\min}$ *of a code* $\mathcal{C}$ *is the minimum block distance value for every possible length* L_k: $b_{\min} = \min_{1\leq k\leq\sigma} b_k$.

Definition 6 *The* diverging distance $d(\mathbf{x}^i, \mathbf{x}^j)$ *(resp. the* converging distance $c(\mathbf{x}^i, \mathbf{x}^j)$*) between two codewords of different lengths* $|\mathbf{x}^i|$ *and* $|\mathbf{x}^j|$ *of a code* $\mathcal{C}$ *is defined as the Hamming distance between the* ℓ*-length prefixes (resp. suffixes) of codewords* $\mathbf{x}^i$ *and* $\mathbf{x}^j$*, with* $\ell = \min\{|\mathbf{x}^i|, |\mathbf{x}^j|\}$.

$$d(\mathbf{x}^i, \mathbf{x}^j) = h(x^i_1\cdots x^i_\ell, x^j_1\cdots x^j_\ell),$$

$$c(\mathbf{x}^i, \mathbf{x}^j) = h(x^i_{|\mathbf{x}^i|-\ell+1}\cdots x^i_{|\mathbf{x}^i|}, x^j_{|\mathbf{x}^j|-\ell+1}\cdots x^j_{|\mathbf{x}^j|}).$$

The minimum diverging distance $d_{\min}$ *(resp.* minimum converging distance $c_{\min}$*) of a code* $\mathcal{C}$ *is the minimum value for all diverging (resp. converging) distances between every possible couple of codewords in* $\mathcal{C}$: $d_{\min} = \min_{(\mathbf{x}^i,\mathbf{x}^j)\in\mathcal{C}, |\mathbf{x}^i|\neq|\mathbf{x}^j|} \left\{ d(\mathbf{x}^i, \mathbf{x}^j) \right\}$ *and* $c_{\min} = \min_{(\mathbf{x}^i,\mathbf{x}^j)\in\mathcal{C}, |\mathbf{x}^i|\neq|\mathbf{x}^j|} \left\{ c(\mathbf{x}^i, \mathbf{x}^j) \right\}$.

Definition 7 *The* free distance d_{free} *is the minimum Hamming distance in the set of all arbitrary extended codes:* $d_{free} = \min\left\{ h(\mathbf{f}^i, \mathbf{f}^j) : \mathbf{f}^i, \mathbf{f}^j \in F_N, N = 1, \ldots, \infty \right\}$.
The free distance d_{free} *of a VLEC code is bounded by ([6]):*

$$d_{free} \geq \min\left(b_{\min}, d_{\min} + c_{\min}\right) \qquad (1)$$

B. VLEC types

We can classify the approaches to VLEC codes into three categories. One category of codes that we denote by family I, corresponds to VLEC codes such that a minimum Hamming distance is maintained between codewords, in order to enable error detection and/or correction. As such, these codes differ from traditional Huffman codes in their treatment of the prefix property: instead of having simply different codeword prefixes, the prefixes must now be a minimum diverging distance $d_{\min}$ apart from each other, offering an error correction capacity $t = \lfloor (d_{\min} - 1)/2 \rfloor$. $\mathcal{C}_3$ given in Table 2 is an example of such a code with $d_{\min} = 3$.

	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$
u	$\mathbf{c}^{(3)}(u)$	$\mathbf{c}^{(4)}(u)$	$\mathbf{c}^{(5)}(u)$
A	0000	0000	0000
B	11100	110111	1110
C	101110	101110	10011
D	011111	011101	01010

Table 2: Examples of VLEC codes.

In a second class, named family II, we place codes that maintain a minimum sliding distance $s_{\min}$ between codewords, offering an error correction capacity $t = \lfloor (s_{\min} - 1)/2 \rfloor$. Considering indeed that variable-length codes are not only susceptible to bit errors but also to loss of synchronisation, protecting only their prefixes may indeed not be enough. As a matter of fact, when synchronisation is lost, prefixes may be shifted. In such cases, the minimum diverging distance is no longer guaranteed and catastrophic error propagation may occur. $\mathcal{C}_4$ given in Table 2 is an example of such a code with $s_{\min} = 3$.

A third class, called family III, is the class of codes ensuring free distance over block-like sequences. Indeed, several approaches [5, 12, 8] have recently been introduced to improve decoding of variable-length codes, that have in common the fact that they consider the overall sequence of variable-length codewords to perform the decoding rather than decoding each codeword instantaneously. Whether working on trellises or code trees, these approches are all able to perform optimal sequence based Maximum A Posteriori (MAP) decoding. As such, these techniques do not rely on the Hamming distance or sliding distance between the codewords, but on the code free distance, as pointed out by Buttigieg [6]. $\mathcal{C}_5$ given in Table 2 is an example of such a code with $b_{\min} = 3$, $d_{\min} = 2$ and $c_{\min} = 1$, resulting in $d_{free} = 3$.

IV. Existence of variable length error correcting codes

The Kraft-MacMillan inequality [11] establishes a lower bound on the existence of a uniquely decodable prefix code given a set of codeword lengths. Following the studies done by Wenisch *et al.* [10] for family I of VLEC codes, which resulted in the following existence condition

$$B_I = \sum_{i=1}^{\mathcal{N}_\mathcal{C}} 2^{-|\mathbf{x}^i|} \sum_{j=0}^{t} \binom{|\mathbf{x}^i|}{j} \leq 1, \quad (2)$$

we propose to establish similar bounds for families II and III. Let $\mathcal{S}$ be an L_σ dimensional binary space in the following.

A. Bound for family II codes

Let $s_{\min}$ be the targetted minimal sliding distance, resulting in a desired error correction $t = \lfloor (s_{\min} - 1)/2 \rfloor$. To achieve error correction t, to each codeword in $\mathcal{C}$ must correspond one of a set of non-overlapping regions in $\mathcal{S}$. Specifically, a codeword $\mathbf{x}^i$ occupies all points in $\mathcal{S}$ whose coordinate representation contains $\mathbf{x}^i$ as a sub-word or any version of $\mathbf{x}^i$ with up to w changes to it. The total number of points in $\mathcal{S}$ occupied by such a codeword is $A^{|\mathbf{x}^i|}_{L_\sigma,t}$, as detailed in Appendix.

For the code to exist, the total number of points occupied by all the codewords can not exceed the space cardinality 2^{L_σ}, which results in the following MacMillan-like inequality:

$$B_{II} = \sum_{i=1}^{\mathcal{N}_\mathcal{C}} 2^{-L_\sigma} A^{|\mathbf{x}^i|}_{L_\sigma,t} = \sum_{i=1}^{\sigma} \frac{n_i}{2^{L_\sigma}} A^{L_i}_{L_\sigma,t} \leq 1. \quad (3)$$

where $A^{L_i}_{L_\sigma,t}$ is derived by recurrence from equations (7)-(9).

B. Bound for family III codes

From equation (1), d_{free} relies on the three distances $d_{\min}$, $c_{\min}$ and $b_{\min}$ and not on a unique one as family I and family II codes do. As a consequence, we obtain the three following conditions for family III codes.

The first condition comes from the minimal diverging distance $d_{\min}$. To ensure such a distance between words, to each codeword of $\mathcal{C}$ must correspond one non-overlapping region of $\mathcal{S}$. Each prefix of length $|\mathbf{x}^i|$ whose coordinate representation contains $\mathbf{x}^i$ as a sub-word or any version of $\mathbf{x}^i$ with up to $\lfloor (d_{\min} - 1)/2 \rfloor$ changes in it, is forbidden. It results in the elimination of all points in $\mathcal{S}$ whose coordinate representations begin with such a forbidden prefix, *i.e.* in the discarding of $\sum_{j=0}^{\lfloor (d_{\min}-1)/2 \rfloor} \left(2^{L_\sigma - |\mathbf{x}^i|} \binom{|\mathbf{x}^i|}{j} \right)$ points. Hence, for the code to exist, we have our first condition

$$B^{d_{\min}}_{III} = \sum_{i=1}^{\sigma} \frac{n_i}{2^{L_i}} \sum_{j=0}^{\lfloor \frac{d_{\min}-1}{2} \rfloor} \binom{L_i}{j} \leq 1. \quad (4)$$

It is to be noted that this first condition is the same as the one obtained for family I in equation (2), which is due to the fact that in both cases we consider the distance on the prefixes.

The second condition is imposed by the minimal block distance. This distance condition concerns only words of equal length, but does not impose anything when word lengths differ. As a consequence, we must work at fixed depth in the code tree, just as would be done for fixed-length codes. For every possible length L_i in the code tree, we find 2^{L_i} possible nodes. Among those, when there are at least two candidate words at same length L_i, *i.e.* $n_i \geq 2$, we need to exclude the nodes whose coordinate representations are any version of candidate words with up to $\lfloor (b_{\min} - 1)/2 \rfloor$ changes in it, that is, $n_i \sum_{j=0}^{\lfloor (b_{\min}-1)/2 \rfloor} \binom{L_i}{j}$ words. Verifying for each length L_i that the total number of occupied and eliminated nodes is inferior to 2^{L_i}, we obtain our second set of conditions on the code existence

$$B^{b_{\min}}_{III}(i) = \frac{(1 - \delta^1_{n_i}) n_i}{2^{L_i}} \sum_{j=0}^{\lfloor \frac{b_{\min}-1}{2} \rfloor} \binom{L_i}{j} \leq 1, \quad 1 \leq i \leq \sigma \quad (5)$$

where δ is Kronecker's symbol. This condition is in fact the classical sphere-packing bound on the size of an error-correcting code of given length and minimal distance. This bound has been improved [13] and is still a research problem [14]. Such refinements remain however too complex to be given here but may be similarly applied.

The third condition results from the minimal converging distance $c_{\min}$. This imposes a distance only on the suffixes of the codewords, but allows a word to contain as internal subword, the suffix of another. As a consequence, we can not proceed similarly as in the $B^{d_{\min}}_{III}$ bound case, but we will consider the leaves at each depth L_i of the code tree. For each depth L_k in the code tree with $L_k < L_i$, we have n_k possible suffixes and we must exclude the points that would correspond to those suffixes among the candidate words of length L_i. For each length L_i, we consequently need to rule out $\binom{L_k}{j}$ possible versions of the suffix with j up to $\lfloor (c_{\min} - 1)/2 \rfloor$ and yet have at least n_i possible candidate words. Hence, a third set of conditions for the code to exist, is given by

$$B^{c_{\min}}_{III}(i,k) = \frac{n_i}{2^{L_i}} + \frac{n_k}{2^{L_k}} \sum_{j=0}^{\lfloor \frac{c_{\min}-1}{2} \rfloor} \binom{L_k}{j} \leq 1, \; 1 \leq k < i \leq \sigma \quad (6)$$

It is to be noted that the three conditions given by inequalities (4), (5) and (6) can not be merged into a unique one. Indeed, they do not correspond to similar exclusion rules, and work on different parts of the code tree. The first one corresponds to the initial branches of the tree, the second to words of equal lengths, *i.e.* to tree nodes of given depth, and the last, to leaves.

C. Numerical example

To illustrate the interest of the bounds determined in this Section, we have aplied them to the length distributions of the codes $\mathcal{C}_4$ and $\mathcal{C}_5$ introduced in Table 2 for different values of $s_{\min}$, $d_{\min}$, $b_{\min}$ and $c_{\min}$. The results obtained are given in Table 3. It is easy to see that they allow to rule out some cases, while confirming the status of other combinations as "possible". For instance, we see that $\mathcal{C}_5$ length distribution is incompatible with $d_{\min} = 3$, whereas it allows the combination $d_{\min} = 2$, $b_{\min} = 3$ and $c_{\min} = 1$, yielding $d_{free} = 3$. In the same way, $\mathcal{C}_4$ length distribution can never verify $b_{\min} = 5$, but is compatible with the bound on sliding distance $s_{\min} = 3$.

bound expression	$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$
B_{II} ($s_{\min}$=3)	0.953125	0.8125	1.1875
B_{II} ($s_{\min}$=5)	2.125	1.875	2.5625
$B_{III}^{d_{\min}}$ ($d_{\min}$=2)	0.125	0.109375	0.1875
$B_{III}^{d_{\min}}$ ($d_{\min}$=3)	0.71875	0.640625	1
$B_{III}^{d_{\min}}$ ($d_{\min}$=5)	1.875	1.71875	2.375
$\max_{\substack{1 \leq i \leq \sigma \\ b_{\min}=3}} B_{III}^{b_{\min}}(i)$	0.21875	0.328125	0.625
$\max_{\substack{1 \leq i \leq \sigma \\ b_{\min}=5}} B_{III}^{b_{\min}}(i)$	0.6875	1.03125	1.375
$\max_{\substack{1 \leq k < i \leq \sigma \\ c_{\min}=1}} B_{III}^{c_{\min}}(i,k)$	0.09375	0.109375	0.1875
$\max_{\substack{1 \leq k < i \leq \sigma \\ c_{\min}=3}} B_{III}^{c_{\min}}(i,k)$	0.34375	0.359375	0.6875
$\max_{\substack{1 \leq k < i \leq \sigma \\ c_{\min}=5}} B_{III}^{c_{\min}}(i,k)$	0.71875	0.734375	1.4375

Table 3: Example of an application of the bounds for VLEC codes.

V. Conclusions

We have presented analytical bounds on the existence of different families of binary VLEC codes. In that way, we have extended research provided in [10], where only one family of variable-length error correcting codes was studied. It has been highlighted that such bounds could be used to help generate VLEC codes more efficiently, as they allow to avoid impossible structures. The adaptation of the methods exposed in this paper in the case of non-binary variable-length codes is left for further studies.

Appendix

Let us denote by $E(m, A)$ the cardinality of the set of strings of length m, that end by padding $A = a_1 a_2 \cdots a_\ell$ of length ℓ and that do not contain a sub-word of Hamming weight inferior or equal to t. Let $A_{L,t}^{\ell}$ be the cardinality of the set of strings of length L that contain a given sub-word of length ℓ or a modified version of it with up to t changes. It is easy to see that $A_{L,t}^{\ell}$ is also the number of words of $\mathcal{S}$ having a sub-word $\mathbf{x}$ of length ℓ with Hamming weight $w(\mathbf{x}) \leq t$, *i.e.* that

$$A_{L,t}^{\ell} = 2^L - \sum_A E(L, A). \tag{7}$$

Now let iteratively derive $E(L, A)$.

Initialization step: $E(\ell, A) = \begin{cases} 0 & \text{if } w(A) \leq t \\ 1 & \text{otherwise} \end{cases}$

Recurrence relation: noticing that a string of length $m+1$ ending with padding $A = a_1 a_2 \cdots a_\ell$ is the concatenation of one bit and a string of length m ending with padding $A' = 0a_1 a_2 \cdots a_{\ell-1}$ or $A'' = 1a_1 a_2 \cdots a_{\ell-1}$, we have:

$$E(m+1, A) = \begin{cases} 0 & \text{if } w(A) \leq t \\ E(x, A') + E(x, A'') & \text{otherwise} \end{cases}$$

This leads to the following matrix expression:

$$\begin{pmatrix} E(m+1, 0...00) \\ E(m+1, 0...01) \\ \vdots \\ E(m+1, 1...11) \end{pmatrix} = M \begin{pmatrix} E(m, 0...00) \\ E(m, 0...01) \\ \vdots \\ E(m, 1...11) \end{pmatrix} \tag{8}$$

where the recurrence matrix $M = (m_{i,j})$ is defined by

$$m_{i,j} = \begin{cases} 1 & \text{if } w(i) > t \text{ and } j \in \{\lfloor i/2 \rfloor; \lfloor (2^\ell + i)/2 \rfloor\} \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

References

[1] C.E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pt. I, pp. 379–423, 1948; pt. II, pp. 623–656, 1948.

[2] G. Battail, *Théorie de l'information*. Paris: Masson, 1997.

[3] J.L. Massey, "Joint source and channel coding", *in Communication Systems and Random Process Theory, NATO Advanced Studies Institutes Series E25*, J. K. Skwirzynski editor, pp. 279-293. Sijthoff & Noordhoff, Alphen aan den Rijn, The Netherlands, 1978.

[4] S.B. Zahir Azami, P. Duhamel and O. Rioul, "Joint source-channel coding: panorama of methods," in *Proceedings of CNES workshop on Data Compression*, Toulouse, France, November 1996.

[5] N. Demir and K. Sayood, "Joint source/channel coding for variable length codes," in *Proceedings of the Data Compression Conference*, pp. 139–148, Snowbird, USA, March-April 1998.

[6] V. Buttigieg, *Variable-length error-correcting codes*. Ph.D. dissertation, University of Manchester, Manchester, United Kingdom, 1995.

[7] S.W. Golomb, "Run-Length encodings," *IEEE Transactions on Information Theory*, vol. 12, pp. 399–401, July 1966.

[8] L. Perros-Meilhac and C. Lamy, "Huffman tree based metric derivation for a low-complexity sequential soft VLC decoding," in *Proceedings of ICC'02*, pp. 783–787, New York, USA, April-May 2002.

[9] D.A. Huffman, "A Method for the Construction of Minimum Redundancy Codes," in *Proceedings of the Institute of Radio Engineers*, vol. 40, pp. 1098–1101, September 1952.

[10] T. Wenish, P.F. Swaszek and A.K. Uht, "Combined Error Correcting and Compressing Codes," in *Proceedings of ISIT'01*, p. 238, Washington, DC, USA, June 2001.

[11] T.M. Cover and J.A. Thomas, *Elements of Information Theory*. New York: John Wiley & Sons, 1991.

[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*, vol. 48(1), pp. 1–6, January 2000.

[13] F.J. MacWilliams and N.J.A. Sloane, *The Theory of Error-Correcting Codes*, chap. 17, Amsterdam: North-Holland, 1977.

[14] B. Mounits, T. Etzion and S. Litsyn, "Improvement on the Johnson upper Bound for Error-Correcting Codes", in *Proceedings of ISIT'02*, p. 345, Lausanne, Switzerland, July 2002.