A Structure for Fast Synchronizing Variable-Length Codes

Slim Chabbouh, *Student Member, IEEE,* and Catherine Lamy, *Member, IEEE*

***Abstract*—A new approach for transforming variable-length codes, optimal in the minimum average codeword length sense, into fast synchronizing codes with the same length distribution is proposed in this letter. Based on the adaptation of an intrinsically fast synchronizing structure whose properties are demonstrated, this approach provides good synchronization recovery for a very low computational complexity. Finally, numerical results are presented and compared with reference ones.**

***Index Terms*—Synchronization, tree construction, variable-length codes.**

I. Introduction

WIDELY used in modern communications due to their good compression capabilities, variable-length codes (VLCs) have the disadvantage of being extremely susceptible to errors. Various techniques have consequently been developed to limit the number of corrupted symbols and the propagation of errors in the decoded bitstream. Among these techniques we find the self-synchronizing codes introduced in [1] and generalized in [2] at the expense of a slight overhead, or the suffix-based algorithm proposed in [3].

We introduce in this letter, a generic variable-length code tree structure with good recovery properties, that is later modified to achieve the optimal length distribution of the Huffman code [4]. Section II presents the chosen error model. The original structure for fast synchronizing variable-length codes is introduced in Section III, where a bound on its performance in terms of error span is derived, following the path opened in [5]. The algorithm used to adapt the structure to the statistics of a given source is presented in Section IV. Numerical results are then given in Section V and finally some conclusions are drawn.

II. Synchronization Properties Analysis

The error model proposed in [5] is used to study the decoder behavior, that is to say the transmission fault process is a random single bit inversion. An *error state* is then defined as a nonsynchronized state. How quickly the decoder recovers synchronization from such a state is the *error span* S [5], i.e., the average number of symbols decoded until synchronization. This value is a good measure of the performance of the code in terms of error recovery and can be expressed as follows:

$$S = \sum_{k \in I} P_{C_k}^{\text{err}} \overline{\eta_k} \tag{1}$$

where I is the set of the codeword indexes, $P_{C_k}^{\text{err}}$ is the probability of the erroneous symbol to be C_k and $\overline{\eta_k}$ is the average number of symbols to be decoded until synchronization when the corrupted symbol is C_k.

For a code which is well matched to the source statistics, the probability of a codeword C_k can be approximated to $P_{C_k} = 2^{-l_k}$, where l_k is the length of C_k and the probability of the erroneous symbol to be C_k can be approximated to $P_{C_k}^{err} = 2^{-l_k} l_k/\overline{l}$, where $\overline{l}$ is the average length of the code, giving the following error span expression:

$$S = \sum_{k \in I} 2^{-l_k} \frac{l_k}{\overline{l}} \overline{\eta_k}. \tag{2}$$

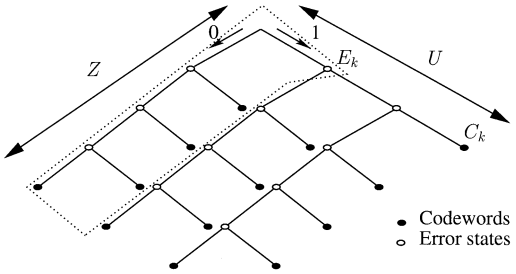


Fig. 1. Tree structure of a fast synchronizing code.

III. Structure for Fast Synchronizing Variable-Length Codes

Let us define the following family $\mathcal{C}$ of variable-length codes in order to minimize the contribution of the most probable codewords on S:

$$\mathcal{C} = \left\{ \begin{array}{c} \{1_i\, 0_j\, 1\}_{i \in [0,U-1]\ j \in [1,Z-1]}, \\ \{1_i\, 0_Z\}_{i \in [0,U-1]}, \\ 1_U \end{array} \right\} \tag{3}$$

where 1_i and 0_i represent length-i strings of ones and zeros and Z (resp. U) denotes the maximum length for strings of zeros (resp. ones) in a codeword, with $U \leq Z$. An example of such a code is given in Fig. 1 for $(Z, U) = (4, 3)$.

A. Effect of Errors

The impact of every possible error on the codewords of $\mathcal{C}$ is given in Table I, with, for each codeword type, the number of

Manuscript received May 15, 2002. The associate editor coordinating the review of this letter and approving it for publication was Prof. M. Fossorier.

S. Chabbouh was with Philips Recherche France, 92156 Suresnes Cedex, France. He is now with the Ecole Nationale Superieure des Telecommuncations, 75013 Paris, France.

C. Lamy was with Philips Recherche France, 92156 Suresnes Cedex, France. She is now with THALES Communications France, TRS/TSI, 92231 Gennevilliers Cedex, France (e-mail: catherine.lamy@fr.thalesgroup.com).

Digital Object Identifier 10.1109/LCOMM.2002.805547

TABLE I
INFLUENCE OF AN ERROR IN A CODEWORD C_k

for an error in C_k	N_{vc}	N_{2vc}	N_{vces}	N_{es}
$C_k = 0\,1$	0	0	0	2
$C_k = 0_{Z-1>j>1}\,1$	1	$j-2$	1	1
$C_k = 0_{Z-1}\,1$	2	$Z-3$	1	0
$C_k = 1_{U-2>i>0}\,0\,1$	1	$i-1$	0	2
$C_k = 1_{U-2}\,0\,1$	2	$U-3$	0	1
$C_k = 1_{U-1}\,0\,1$	1	$U-2$	1	1
$C_k = 1_{U-1>i>0}\,0_{Z-1>j>1}\,1$	2	$i+j-3$	1	1
$C_k = 1_{U-1>i>0}\,0_{Z-1}\,1$	2	$i+Z-4$	2	0
$C_k = 1_{U-1}\,0_{Z-1>j>1}\,1$	1	$U+j-3$	1	1
.....	.....	.....	.....	.....
$C_k = 1_{U-1}\,0_{Z-1}\,1$	1	$U+Z-4$	2	0
$C_k = 1_U$	1	0	$U-2$	1
$C_k = 0_Z$	1	0	$Z-2$	1
$C_k = 1_{U-1>i>0}\,0_Z$	1	$i-1$	$Z-1$	1
$C_k = 1_{U-1}\,0_Z$	1	$U-2$	Z	0

error positions resulting in a given transformation of the corrupted symbol: N_{vc} for the transformation of the original codeword into a valid codeword, N_{2vc} into the concatenation of two valid codewords, N_{es} into an error state and N_{vces} into the concatenation of a valid codeword and an error state. It is to be noted that each error state obtained in Table I belongs to the set $\mathcal{E} = \{1_\alpha\,0_\beta\}_{\alpha\in[0,U-1]\ \beta\in[0,Z-1]}$.

B. Recovery From Error States and Bound on Error Span

Considering that the recovery from an error state $E_k \in \mathcal{E}$ resulting from an erroneous codeword C_k also depends on the codeword C_h following the error state, the expression of S in (2) becomes

$$S = \sum_{(k,h)\in I^2} P_{C_k}^{err} P_{C_h} \overline{\eta_{k,h}} \simeq \sum_{(k,h)\in I^2} 2^{-(l_k+l_h)} \frac{l_k}{\bar{l}} \overline{\eta_{k,h}} \quad (4)$$

where $\overline{\eta_{k,h}}$ is the average number of symbols decoded until synchronization when the corrupted symbol is C_k and the following is C_h and l_h is the length of C_h.

Proposition 1: For any error state E_k resulting from a corrupted codeword C_k and its following codeword C_h such that $C_h \in \tilde{\mathcal{C}} = \mathcal{C} \setminus \{1_U, 0_Z\}$, synchronization is always recovered after decoding C_h.

Proof: Let E_k and C_h be an error state and its following codeword. The decoded sequence, concatenation of E_k and C_h, has either the form $1_\alpha 0_\beta 1_i 0_j 1$, with $i \in [0, U-1], j \in [1, Z-1]$ or the form $1_\alpha 0_\beta 1_i 0_Z$, with $i \in [1, U-1]$. The synchronization is recovered in both cases, as the decoded sequence is a valid codeword or the concatenation of two valid codewords. ■

Proposition 2: The concatenation of an error state E_k resulting from a corrupted codeword C_k and a following codeword $C_h \in \{1_U, 0_Z\}$ results either in an immediate resynchronization or in the concatenation of a valid codeword and a new error state $E_h \in \mathcal{E}$.

Proof: Similar to the previous case, we find that the decoded sequence has either the form $1_\alpha 0_\beta 1_U$ or the form $1_\alpha 0_\beta 0_Z$. This results whether in the concatenation of a valid codeword and an error state 1_α, 1_{U-1} or $0_{Z-\beta}$ or in a valid codeword, hence the proposition. ■

Proposition 3: Assuming $1 \gg 2^{-Z}$ and $1 \gg 2^{-U}$, the error span S is upper-bounded by 2.

Proof: As, by Proposition 1, two codewords ensure the resynchronization after an error state, we can derive bounds for $\overline{\eta_{k,h}}$ when $(C_k, C_h) \in \mathcal{C} \times \tilde{\mathcal{C}}$ from Table I with the formula: $\overline{\eta_{k,h}}_{\{C_k\}} \leq (N_{vc_k} + 2N_{2vc_k} + 3N_{vces_k} + 2N_{es_k})/l_k$. This results in: $\overline{\eta_k}_{\{C_k = 1_{U-1}0_{Z-1}1\}} \leq 2 + 1/(U+Z-1)$, $\overline{\eta_k}_{\{C_k\in\{1_i0_j1\}\setminus 1_{U-1}0_{Z-1}1\}} \leq 2$, $\overline{\eta_k}_{\{C_k\in\{1_U\}\cup\{1_i0_Z\}\}} \leq 3$. From (4) and as, by Proposition 2, $\overline{\eta_{k,1_U}} \leq 1 + \overline{\eta_k}$ and $\overline{\eta_{k,0_Z}} \leq 1 + \overline{\eta_k}$, we obtain:

$$S \leq \frac{\sum_{C_k\in\tilde{\mathcal{C}}} \left(\overline{\eta_k} 2^{-l_k} \frac{l_k}{\bar{l}}\right) + \frac{1}{2^U} + \frac{1}{2^Z}}{\left(1 - \frac{1}{2^U} - \frac{1}{2^Z}\right)}$$

where the sum over C_k, from the upper-bounds on $\overline{\eta_k}$ determined above, is found equal to

$$\frac{2 + 2^{-U-2}(U-8) + 2^{-Z-1}(Z-3) + 2^{-U-Z-1}(4-Z-U)}{1 - 2^{-U} - 2^{-Z} + 2^{-Z-U}}.$$

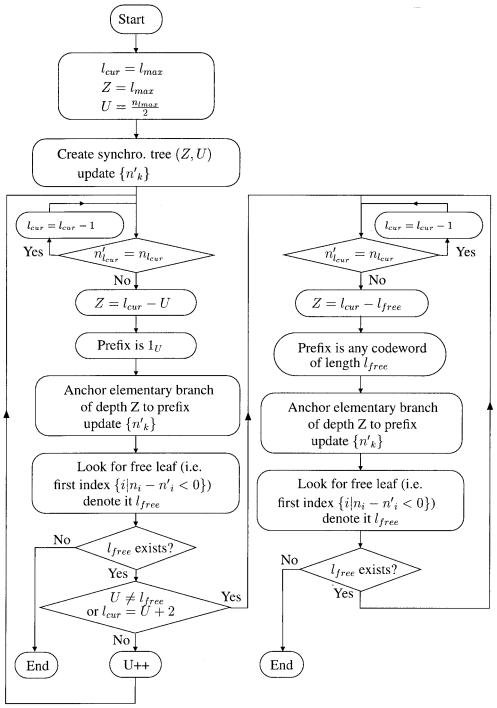
Replacing this expression in the upper-bound on S, when $1 \gg 2^{-Z}$ and $1 \gg 2^{-U}$, we obtain $S \leq 2$.

IV. OPTIMIZING THE SYNCHRONIZATION OF A HUFFMAN CODE

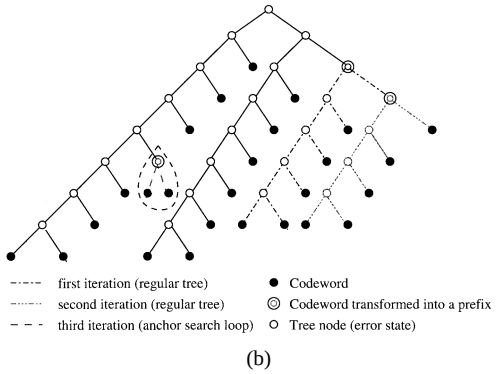
The main disadvantage of the structure proposed in Section III is that, although it offers very good error recovery performance, it cannot reach every possible compression efficiency and is far from optimal average length. The structure must consequently be modified to be applied to any given source. Since the restrictions come from the fact that our codes are the repetition of U *elementary branches* of the same depth Z (dashed set in Fig. 1), we propose to build codes where the branch size may vary. Provided the length of the smallest branch is long enough, similar statistical synchronization properties will be obtained. Finally, modifying the existing tree, by using some of its codewords as prefixes, will hopefully preserve the statistical good behavior.

Following this, we propose to build fast synchronizing codes with the same *length distribution* as the binary Huffman codes. Let $L = (n_i)_{i=1,\ldots,l_{\max}}$ be the Huffman code length distribution, with n_i the number of codewords of length i, $l_{\max}$ the greatest codeword length and, by construction [4], $n_{l_{\max}}$ even. The algorithm given in the flowchart in Fig. 2(a) produces a code with a length distribution $L' = (n'_i)_{i=1,\ldots,l_{\max}}$ identical to L after the following main steps:

- create the skeleton tree with decreasing depths for each elementary branch with parameters $Z = l_{\max}$ and $U = n_{l_{\max}}/2$ to ensure that $n'_{l_{\max}} = n_{l_{\max}}$;
- for each length l_{cur} beginning from $l_{\max}$, if $n'_{l_{cur}} \neq n_{l_{cur}}$, use codeword 1_U as a prefix and anchor to it the maximal size elementary branch of depth $Z' = l_{cur} - U$ (left loop);
- if 1_U can not be used as a prefix, either because l_{cur} is too small or because using 1_U would irreparably deplete the current length distribution, find a suitable prefix (which exists since the Huffman distribution is not yet reached) by choosing the smallest free codeword (right loop). See for instance the dashed set in Fig. 2(b).



(a)



(b)

Fig. 2. Flowchart and example of an application of the synchronization optimization algorithm for Huffman codes. (a) Flowchart. (b) Example of optimized tree construction for motion vectors of a video codec.

V. Numerical Results

The results obtained, estimating S by simulation as in [5], with our optimization algorithm are presented in Table II for the two reference codes proposed in [3], i.e., a code for motion vectors and the English alphabet. In both cases, our codes perform as well as those obtained by Takishima, which is not surprising considering that the short words of these codes are similar, except for bit-inversions. Our algorithm is however much less complex. In fact, it is so simple that it can even be applied by hand for these relatively short codes, as illustrated in Fig. 2(b) for the motion vectors code, where the fast synchronizing structure is obtained in only three iterations of the algorithm. The simplicity of the algorithm allows us to apply it also to longer codes, such as the 206-symbols variable-length code used in the H.263 video codec [6] to encode the DCT coefficients. We obtain a much smaller error span $S_{\text{optim}} = 2.1634$ than the original one $S_{\text{orig}} = 3.3658$ for the same average length, which means the decoder would statistically resynchronize one symbol before the current case with our optimized code.

TABLE II
Comparing Takishima Codes With Our Optimised Ones

	Ref. [3]	Optimised		Ref. [3]	Optimised
S	1.6955	1.6851	S	2.1782	2.1899
MV1	10	01	E	110	001
MV2	110	001	T	010	101
MV3	010	101	O	1110	0001
MV4	0110	0001	A	0110	1001
MV5	1110	1001	N	1010	1101
MV6	0010	1101	I	0010	1111
MV7	0000	1111	R	1000	0101
MV8	00110	10001	S	0000	0111
MV9	11110	11001	H	1001	0110
MV10	01110	11101	D	01110	00001
MV11	001110	000001	L	00110	10001
MV12	111110	100001	U	11110	11001
MV13	011110	110001	C	10110	11101
MV14	000110	111001	M	00010	11100
MV15	000100	000011	P	11111	01001
MV16	000111	000010	F	00011	01000
MV17	0011110	0000001	Y	011110	000001
MV18	1111110	1000001	W	001110	100001
MV19	0111110	1100001	G	101110	100000
MV20	0001010	1100000	B	001111	110001
MV21	0111111	1110001	V	101111	110000
MV22	0011111	1110000	K	0111110	0000001
MV23	11111110	00000001	J	01111110	00000001
MV24	00010110	00000000	X	011111110	000000001
MV25	11111111	10000001	Z	0111111110	0000000001
MV26	00010111	10000000	Q	0111111111	0000000000

VI. Conclusion

We presented a new structure of variable-length codes with very good error recovery properties and explained their good behavior by deriving an asymptotical bound on their error span. A low-complexity algorithm was then proposed, which modifies this structure to obtain codes with good error recovery properties that are tuned to any chosen source statistics. This algorithm was shown to offer at least similar synchronization properties for several small and large variable-length codes well-known in the literature for a noticeable complexity reduction.

Acknowledgment

The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] T. J. Ferguson and J. H. Rabinowitz, "Self-synchronizing Huffman codes," *IEEE Trans. Inform. Theory*, vol. 30, pp. 687–693, July 1984.
[2] B. Montgomery and J. Abrahams, "Synchronization of binary source codes," *IEEE Trans. Inform. Theory*, vol. 32, pp. 849–854, Nov. 1986.
[3] Y. Takishima, M. Wada, and H. Murakami, "Error states and synchronization recovery for variable length codes," *IEEE Trans. Commun.*, vol. 42, pp. 783–791, Feb./Mar./Apr. 1994.
[4] D. A. Huffman, "A method for the construction of minimum redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, Sept. 1952.
[5] J. C. Maxted and J. P. Robinson, "Error recovery for variable length codes," *IEEE Trans. Inform. Theory*, vol. 31, pp. 794–801, Nov. 1985.
[6] "Video coding for low bitrate communications," Int. Telecommunication Union, ITU-T Draft H263, 1996.