Iterative Construction of Reversible Variable-Length Codes and Variable-Length Error-Correcting Codes

Jin Wang, *Student Member, IEEE*, Lie-Liang Yang, *Senior Member, IEEE*, and Lajos Hanzo, *Fellow, IEEE*

***Abstract*—We propose a generic algorithm for the construction of efficient reversible variable-length codes (RVLCs) and variable-length error-correcting (VLEC) codes, which optimizes the codeword length distribution. The algorithm may be applied to any existing codeword selection mechanism, and it is capable of generating codes of higher efficiency in comparison to the algorithms disseminated in the literature.**

***Index Terms*—Code design, free distance, Huffman codes, reversibile variable length codes (RVLCs), variable length error correcting (VLEC) codes.**

I. Introduction

REVERSIBLE variable-length codes (RVLCs) were proposed for facilitating the bidirectional decoding of a source-encoded bitstream, which mitigate for example the visual effects of transmission errors in case of losing synchronization between the encoder and decoder in wireless video telephony [1]. The application of RVLCs has been extensively studied [2]–[9], particularly during the development of the video standards H.264 and MPEG-4.

The construction of RVLCs was studied in [4]–[9]. Most RVLC constructions, commence from a Huffman code designed for the source concerned and then replace the Huffman codewords by identical length codewords that satisfy both the prefix and suffix conditions necessitated. If the number of valid codewords is insufficient, longer codewords have to be assigned, resulting in an increased redundancy. By contrast, if there are more candidate codewords than necessary, different codeword selection mechanisms may be applied. For example, the maximum symmetrical suffix length (MSSL) scheme [5] may be used for designing symmetric RVLCs, or the minimum repetition gap (MRG) [6] metric and the so-called affix index [8] metric may be invoked for designing asymmetric RVLCs.

All of the above codeword selection mechanisms attempt to match the codeword length distribution of the RVLC to that of the corresponding Huffman code. Since a RVLC has to satisfy the suffix condition in addition to the prefix condition, the desirable codeword length distribution of a Huffman code is usually not matched by that of the RVLC for the same source. Moreover, both the value of the minimal average codeword length and the optimal codeword length distribution of the RVLC are unknown for a given source. Therefore, we propose a heuristic method of finding an improved codeword length distribution in Section II of this letter. Experimental results show that the proposed algorithm generally produces more efficient RVLCs than all known algorithms.

Recently, variable-length error-correcting (VLEC) codes have rekindled the community's interest [10] owing to their good distance properties, which were studied in [11], [12], where a heuristic VLEC code construction algorithm was proposed and optimized. However, the optimization procedure may become prohibitively complex for large-alphabet sources. Hence, in Section III we apply a method similar to that used in Section II as an alternative to the optimization procedure of [11], resulting in a significantly lower complexity. Section IV concludes the letter.

Manuscript received March 9, 2004. The associate editor coordinating the review of this letter and approving it for publication was Prof. A. Haimovich.

The authors are with the School of Electronics and Computer Science, University of Southampton SO17 1BJ, U.K. (e-mail: lh@ecs.soton.ac.uk).

Digital Object Identifier 10.1109/LCOMM.2004.837645

II. Construction of RVLCs

Let $U = \{u_1, \ldots, u_M\}$ be an M-ary i.i.d. information source having the probability mass function of $P_u = \{p_1, \ldots, p_M\}, p_1 \geq \cdots \geq p_M$. The proposed algorithm bases its codeword selection procedure on that of the algorithm in [6] or [8] and attempts to optimize the length distribution of the resultant RVLC level by level[1]. The construction procedure is as follows.

Step 1) Employ Tsai's [6] or Lakovic' [8] algorithm for constructing an initial RVLC, and calculate the codeword length vector $\mathbf{n}$, where $n(i)$ is the number of codewords of length i in bits.

Step 2) Increase the number of required codewords at length i by one, and decrease the number of required codewords at the maximum codeword length $L_{\max}$ by one, yielding

$$n(i) := n(i) + 1, \quad n(L_{\max}) := n(L_{\max}) - 1.$$

Use Tsai's or Lakovic's algorithm for constructing an RVLC based on the modified length distribution instead of that of the Huffman code.

Step 3) Recalculate the average code length $\bar{L}$, and if $\bar{L}$ was increased, restore the previous length distribution, and proceed to the next level; otherwise update the codeword length vector $\mathbf{n}$ and go to Step 2.

Step 4) Repeat Steps 2 and 3 until the last level is reached.

Table I compares the RVLCs designed for the English alphabet, constructed using the proposed algorithm (based on Tsai's algorithm [6]), and the algorithms published in [4], [6], [8]. As a benefit of its improved length distribution, the proposed algorithm yields a RVLC of significantly higher

[1]The proposed algorithm is also applicable to optimize the construction of symmetric RVLCs of [5].

TABLE I
VLCs FOR THE ENGLISH ALPHABET

	Huffman code	Takishima's [4] RVLC	Tsai's [6] RVLC	Lakovic's [8] RVLC	Proposed RVLC
Codeword length vector	(0,0,2,7,7,5,1,1,1,2)	(0,0,2,5,7,5,1,2,1,1,1,1)	(0,0,2,7,4,3,5,2,1,2)	(0,0,2,7,6,2,3,3,1,2)	(0,0,3,6,6,2,2,2,2,2,1)
Avg. length	4.15572	4.36068	4.30678	4.25145	4.18732
Max length	10	12	10	10	11

TABLE II
VLCs FOR CANTERBURY CORPUS, RESULTS OF TSAI'S RVLCs AND LIN'S RVLCs ARE FROM [9]

File	Number of Codewords	Huffman Code		Tsai's [6] RVLC		Lin's [9] RVLC-1		Lin's [9] RVLC-2		Proposed RVLC	
		Avg.	Max	Avg.	Max	Avg.	Max	Avg.	Max	Avg.	Max
asyoulik.txt	68	4.84465	15	5.01142	15	5.00954	15	5.13624	11	4.92273	11
alice29.txt	74	4.61244	16	4.80326	17	4.68871	18	4.86762	11	4.70569	11
xargs.1	74	4.92382	12	5.07334	13	5.16087	15	5.27537	11	5.00166	11
grammar.lsp	76	4.66434	12	4.85461	12	4.78581	17	4.96130	11	4.80247	12
plrabn12.txt	81	4.57534	19	4.80659	19	4.64910	17	4.84043	11	4.71036	12
lcet10.txt	84	4.69712	16	4.87868	16	4.74177	17	4.93372	11	4.80024	12
cp.html	86	5.26716	14	5.37113	14	5.77080	16	5.74580	11	5.29342	14
fi elds.c	90	5.04090	13	5.26987	13	5.20278	13	5.36233	11	5.18341	11
ptt5	159	1.66091	17	1.71814	17	1.70401	15	1.72843	13	1.69580	13
sum	255	5.36504	14	5.49767	13	6.01870	15	5.78572	13	5.47330	13
kennedy.xls	256	3.59337	12	3.89401	13	3.85384	14	3.86296	14	3.79759	14

efficiency. In [9] Lin *et al.* proposed two backtracking based construction algorithms for asymmetrical RVLCs, which are capable of achieving either lower average codeword lengths or shorter maximum codeword lengths in comparison to the algorithm of [6]. Table II compares the achievable performance of the algorithm proposed in this paper and that of various benchmarkers based on the Canterbury Corpus file set (available in http://corpus.canterbury.ac.nz), which was designed specifically for testing new compression algorithms. It is shown that the proposed algorithm is capable of reducing both the average codeword length and the maximum codeword length at the same time.

In [7] Lakovic *et al.* proposed a construction algorithm based on the MRG metric for designing RVLCs having a free distance of $d_f = 2$, which significantly outperformed the regular RVLCs having a free distance of $d_f = 1$, when using a joint source/channel decoding scheme. Since the RVLC construction of [7] was also based on the codeword length distribution of Huffman codes, the proposed optimization procedure may be readily invoked in this context. Owing to space limitation, the corresponding results are omitted here.

III. CONSTRUCTION OF VLEC CODES

The family of VLEC codes may be characterized by three different distances, namely the minimum block distance $b_{\min}$, the minimum diverging distance $d_{\min}$, and the minimum converging distance $c_{\min}$ [11]. Their relationship is described by the following formula [11]:

$$d_f \geq \min(b_{\min}, d_{\min} + c_{\min}).$$

It may be shown that Huffman codes and RVLCs may also be viewed as VLEC codes. For Huffman codes, we have $b_{\min} = d_{\min} = 1, c_{\min} = 0$. By contrast, for RVLCs, we have $d_{\min} = c_{\min} = 1$ and $b_{\min} = 1$ for RVLCs having a free distance of $d_f = 1$, while $b_{\min} = 2$ for RVLCs having a free distance of $d_f = 2$.

The construction of an optimal VLEC code requires finding a variable length code, which satisfies the specific distance requirements and additionally has a minimum average codeword length. Generally, one takes $b_{\min} = d_{\min} + c_{\min} = d_f$ and $d_{\min} = \lceil d_f/2 \rceil$. In [11] Buttigieg proposed the following heuristic construction method.

— Initialize the target VLEC codeword set C to a fixed-length code of length L_1 having a minimal distance $b_{\min}$. This step, as well as all following steps resulting in sets of identical-length codewords having a given distance, are carried out with the aid of either the greedy algorithm (GA) or the majority voting algorithm (MVA) [11].

— Create a set W that contains all L_1-tuples having at least a distance $d_{\min}$ from each codeword in C. If the set W is not empty, an extra bit is affixed at the end of all its words. This new set having twice the number of words replaces the previous set W.

— Delete all words in the set W that do not have at least a distance $c_{\min}$ from all codewords of C. At this point, the set W satisfies both the $d_{\min}$ and $c_{\min}$ minimum distance requirements with respect to the set C.

— Select the codewords from the set W that are at a distance of $b_{\min}$ using the GA or the MVA. The selected codewords are then added to set C.

TABLE III
PROPOSED VLEC CODES FOR THE ENGLISH ALPHABET

Free Distance	Avg. Length	Max Length
$d_f = 1$	4.19959	10
$d_f = 2$	4.23656	9
$d_f = 3$	6.20530	11

The procedure is repeated until it finds the required number of codewords or has no more options to explore. If no more codewords of the required properties can be found, or if the maximal codeword length is reached, shorter codewords are deleted until one finds an adequate VLEC code structure.

This basic construction does not exploit the source statistics, hence may result in inefficient codes. Buttigieg [11] also proposed an exhaustive search procedure for optimizing the codeword length distribution, which starts by deleting codewords from the last added group and tests all possible codeword length combinations, then selects the best one as the final design. However, its complexity increases exponentially with the number of source symbols.

Hence here we propose a codeword length distribution optimization procedure similar to that of Section II as an alternative to the exhaustive search.

Step 1) Use the above procedure for constructing an initial VLEC code, and calculate the codeword length vector $\mathbf{n}$.

Step 2) Decrease the number of codewords required at length $i, n(i)$, by one. The values of $n(j), j \leq i$ are retained and no restrictions are imposed on the number of codewords at higher lengths. Then use the above procedure again for constructing a VLEC code based on the modified length distribution.

Step 3) Recalculate the average codeword length $\bar{L}$, and if $\bar{L}$ was increased, restore the previous length distribution, and proceed to the next codeword length; otherwise update the codeword length vector $\mathbf{n}$ and go to Step 2.

Step 4) Repeat Steps 2 and 3 until the last codeword length is reached.

In [11] Buttigieg only considered the construction of VLEC codes having $d_f > 2$. The experimental results of Table IV demonstrate that the proposed algorithm may also be used for constructing VLEC codes having $d_f = 1, 2$, which are the equivalents of the RVLCs having $d_f = 1, 2$ in Section II. It is worth noting that the VLEC code having $d_f = 2$ results in an even shorter average codeword length, than the corresponding RVLC having $d_f = 2$ proposed in [7], where the average codeword length is 4.34534 and the maximum codeword length is 10, which indicates that this is a meritorious way of constructing such codes.

Table V compares the VLEC codes having $d_f = 3, 5, 7$ constructed by the proposed algorithm to those of [11]. By using the proposed optimization procedure instead of the exhaustive search of [11], only a small fraction of all possible codeword length combinations are tested, hence the construction complexity is significantly reduced without compromising the achievable performance. For sources of larger alphabets, the "accelerated" search techniques of [13] may be invoked for further complexity reduction.

TABLE IV
COMPARISON OF VLEC CODES FOR THE ENGLISH ALPHABET†

	Buttigieg's VLEC code [11]		Proposed VLEC code	
Free Distance	Avg. Length	Max Length	Avg. Length	Max Length
$d_f = 3$	6.370	13	6.3038	11
$d_f = 5$	8.467	12	8.4752	12
$d_f = 7$	10.70	15	10.7594	15

† The source distribution of [11] was used, which is different from that used in Table I and Table III.

IV. CONCLUSION

A generic algorithm was proposed for the construction of efficient RVLCs and VLEC codes. By means of iterative construction, it optimizes the codeword length distribution level by level, and results in codes of higher efficiency and/or shorter maximum codeword length than the algorithms previously disseminated in the literature. The optimization procedure commences its operation from the outer layer of the code construction, thus it can be applied to any existing algorithm aiming for efficient codeword design.

ACKNOWLEDGMENT

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

REFERENCES

[1] L. Hanzo, P. J. Cherriman, and J. Streit, *Wireless Video Communications: Second to Third Generation Systems and Beyond*. Piscataway, NJ: IEEE Press, 2001.

[2] J. Wen and J. D. Villasenor, "A class of reversible variable length codes for robust image and video coding," in *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, Lausanne, Switzerland, Oct. 1997, pp. 65–68.

[3] ——, "Reversible variable length codes for efficient and robust image and video coding," in *Data Compression Conf.*, UT, 1998, pp. 471–480.

[4] Y. Takishima, M. Wada, and H. Murakami, "Reversible variable length codes," *IEEE Trans. Commun.*, vol. 43, pp. 158–162, Feb./Mar./Apr. 1995.

[5] C. Tsai and J. Wu, "Modified symmetrical reversible variable-length code and its theoretical bounds," *IEEE Trans. Inform. Theory*, vol. 47, pp. 2543–2548, Sept. 2001.

[6] ——, "On constructing the huffman-code-based reversible variable-length codes," *IEEE Trans. Commun.*, vol. 49, pp. 1506–1509, Sept. 2001.

[7] K. Lakovic and J. Villasenor, "On design of error-correcting reversible variable length codes," *IEEE Commun. Lett.*, vol. 6, pp. 337–339, Aug. 2002.

[8] ——, "An algorithm for construction of efficient fix-free codes," *IEEE Commun. Lett.*, vol. 7, pp. 391–393, Aug. 2003.

[9] C. Lin, Y. Chuang, and J. Wu, "Generic construction algorithms for symmetric and asymmetric RVLCs," in *Proc. IEEE Int. Conf. on Computational Science(ICCS)*, Amsterdam, The Netherlands, Apr. 2002, pp. 968–972.

[10] A. Hedayat and A. Nosratinia, "Concatenated error-correcting entropy codes and channel codes," in *IEEE Int. Conf. on Commun. (ICC'03)*, vol. 5, Anchorage, AK, 2003, pp. 3090–3094.

[11] V. Buttigieg, "Variable-Length Error-Correcting Codes," Ph.D. dissertation, Dep. Elect. Eng., Univ. of Manchester, Manchester, U.K., 1995.

[12] ——, "Variable-length error-correcting codes," *Proc. IEE—Commun.*, vol. 147, no. 4, pp. 211–215, Aug. 2000.

[13] C. Lamy and J. Paccaut, "Optimized constructions for variable-length error correcting codes," in *Proc. IEEE Inform. Theory Workshop (ITW2003)*, Paris, France, Mar. 31–Apr. 4 2003, pp. 183–186.