Journal of Zhejiang University SCIENCE A
ISSN 1009-3095 (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

On constructing symmetrical reversible variable-length codes independent of the Huffman code*

HUO Jun-yan†, CHANG Yi-lin, MA Lin-hua, LUO Zhong
(*The National Key Laboratory on Integrated Service Networks, Xidian University, Xi'an 710071, China*)
†E-mail: jyhuo@mail.xidian.edu.cn

Received Dec. 12, 2005; revision accepted Feb. 22, 2006

Abstract: Reversible variable length codes (RVLCs) have received much attention due to their excellent error resilient capabilities. In this paper, a novel construction algorithm for symmetrical RVLC is proposed which is independent of the Huffman code. The proposed algorithm's codeword assignment is only based on symbol occurrence probability. It has many advantages over symmetrical construction algorithms available for easy realization and better code performance. In addition, the proposed algorithm simplifies the codeword selection mechanism dramatically.

Key words: Error resilience, Average code length, Huffman codes, Symmetrical reversible variable length codes (RVLCs)
doi:10.1631/jzus.2006.AS0059 **Document code:** A **CLC number:** TN919.8

INTRODUCTION

Reversible variable length codes (RVLCs) that can be instantaneously decoded both forward and backward have attracted many researchers' attention. Because of its excellent error resilient capabilities, RVLC is adopted by the new video coding standards H.263+ and MPEG-4. Meanwhile, associated with the insertion of synchronization markers, RVLC can improve its error resilience capabilities.

Research on RVLC was first published by Takishima *et al.*(1995). RVLC can be divided into symmetrical RVLC and asymmetrical RVLC. Tsai and Wu (2001a; 2001b) improved the RVLC codeword selection mechanism by arranging the candidate code according to a certain order. Jeong and Ho (2003a; 2003b) developed a symmetrical RVLC construction algorithm derived from the arbitrary assignment of '0' and '1' and constructed a symmetrical RVLC on a half binary tree. All of these algorithms existing are based on the Huffman code. Each of them is designed by starting from the Huffman code and converting the code into a symmetrical RVLC by respective codeword selection mechanism.

In this paper, a novel construction algorithm for symmetrical RVLC which is independent of the Huffman code is proposed. Codeword assignment in the proposed algorithm is only based on symbol occurrence probability. That is to say, there is no need to construct the Huffman code beforehand. In addition, the proposed method provides a shorter average code length, lower complexity and easier realization.

PROPERTY OF SYMMETRICAL RVLC CODE TREE

Fig.1 shows a Huffman code tree and a symmetrical RVLC code tree of 4 levels. The maximum codeword candidates number $m_0(L)$ of the Huffman code tree at level L is

$$m_0(L)=2^L, \tag{1}$$

while the maximum codeword candidates number

---

* Project (No. 60172030) partially supported by the National Natural Science Foundation of China

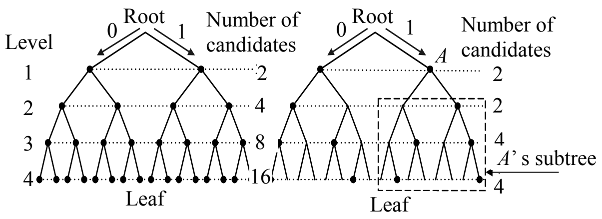

Fig.1 Distributions of the Huffman codeword candidates and symmetrical RVLC codeword candidates

$m_1(L)$ of symmetrical RVLC code tree at level L is

$$m_1(L) = 2^{\lfloor (L+1)/2 \rfloor}, \tag{2}$$

where $\lfloor x \rfloor$ denotes the largest integer less than or equal to x. Eq.(2) indicates that the odd level provides the same number of symmetrical RVLC codeword candidates as that of the next even level.

A leaf-to-root mechanism is necessary for constructing the Huffman code because if all candidates at the lower level are selected as codewords, the candidates in the following levels cannot be further selected due to the violation of the prefix condition. Nevertheless the above problem will not arise in the process of constructing symmetrical RVLC. For example, if all of the symmetrical candidates at level 2, '00' and '11', are selected, at level 3, '010' and '101' can also be used. That is to say, a root-to-leaf codeword selection mechanism may be used for constructing symmetrical RVLC.

The RVLC has to satisfy the prefix condition and the suffix condition to be decoded instantaneously in forward and backward directions. Especially for symmetrical RVLC codeword, the prefix must be identical with the suffix. Consequently, the codeword selection mechanism of symmetrical RVLC selects codewords satisfying symmetry and prefix condition at the same time.

NEW CONSTRUCTION ALGORITHM OF SYMMETRICAL RVLC

In this section, related works and details of the new construction algorithm for symmetrical RVLC are addressed. First, a basic type of container class in data structure is introduced, queue, which is a linear data structure with some restrictions on its operation (Fig.2). One can "enqueue" a code such as a_n by adding it to the rear of the queue, and "dequeue" a code by removing it from the front of the queue.

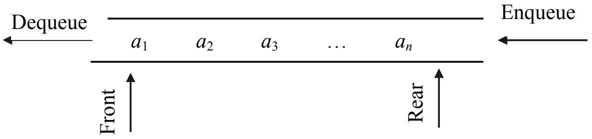

Fig.2 The operation defined in the queue

Queue is used to store the candidate codes. Every candidate code a_i in the queue is a string of bits such as '01001...'. The initialization of the queue is a crucial issue. The minimum codeword length of symmetrical RVLC codeword denoted as $L_{\min}$ has great effect on the performance of the algorithm and its choice is discussed later in this section. The queue is initialized as all $2^{L_{\min}}$ candidates of $L_{\min}$ bits. For instance, suppose $L_{\min}$ is equal to 2, then four possible combinations of '0' and '1', '00', '01', '10' and '11' are pushed into the queue at first.

The operation on the queue is as follows: the front candidate code is dequeued and checked whether it is symmetrical or not. If that is true, this candidate code is selected as a symmetrical RVLC codeword, otherwise its derived codes (which are constructed by adding '0' and '1' at the end of the original code) are enqueued. In the example above, the first candidate code '00' will be selected as a symmetrical RVLC codeword because it is symmetrical. And the following candidate code '01' is asymmetrical, so its derived codes '010' and '011' will be pushed into the queue. By this way, the candidate codes which satisfy the prefix condition are pushed into the queue successively.

The procedure for constructing symmetrical RVLC is summarized as follows:

Step 1: Arrange symbol probabilities in decreasing order, and initialize the queue with all $2^{L_{\min}}$ candidates of $L_{\min}$ bits.

Step 2: Dequeue the front candidate code, and check whether it is symmetrical. If so, select it as the target symmetrical RVLC codeword and assign it to the successive symbol. If not, push its derived codes into the queue.

Step 3: Repeat Step 2 until every symbol has been assigned a symmetrical RVLC codeword.

Step 4: Calculate the average code length.

The selection of $L_{\min}$ is discussed now. As known, the entropy, noted as $H(x)$, is the theoretical limit of the average code length. It is obvious that $L_{\min}$ should be less than $H(x)$ for a short average code length. In this paper, $L_{\min}$ is calculated as follows:

$$L_{\min,i} = \max(l_{\min,i}, 1), \tag{3}$$

where $l_{\min,i} = 1, 2, ..., \lfloor H(x)+1 \rfloor$.

The minimum codeword length $L_{\min}$ is selected as the $L_{\min,i}$ compared with which the average code length is shorter.

ANALYSIS AND EXPERIMENTAL RESULTS

Improvements of the proposed algorithm over the existing ones come from the following facts: (1) This algorithm is independent of the Huffman code, and can directly construct the symmetrical codeword by occurrence probabilities. (2) Codewords at lower levels are sufficiently used for the most common symbols. The proposed algorithm can provide better performance over the conventional methods. (3) The most significant advantage of the proposed algorithm over other methods is that the proposed algorithm simplifies the codeword selection mechanism dramatically. In the conventional RVLC schemes, much time is used to pre-compute the number of available symmetrical codewords at each level and to judge the prefix condition, which can be saved in our algorithm. The codeword selection mechanism of the proposed algorithm is as follows. It introduces the queue as its store container. In the process of selection, if a candidate code has been selected as a symmetrical RVLC codeword, all its subtree candidate codes will not be pushed into the queue anymore. Only the derived codes of those candidate codes which are asymmetrical will be pushed into the queue as new candidate codes. That is to say, all of the candidate codes in the queue always satisfy the prefix condition. So what we have to do is to check whether the candidate code in the queue is symmetrical or not. (4) The minimum codeword length is a key factor because it affects the average codeword length and the coding performance significantly. In the Huffman-code-based algorithms, $L_{\min}$ is equal to the length of the shortest codeword of the given Huffman code ($L_{\text{Huffman\_min}}$). $L_{\min}$ in the proposed algorithm is calculated by Eq.(3) based on the symbols' probabilities.

Table 1 presents the proposed symmetrical coding result for several typical probability distributions such as English alphabet distribution, uniform distribution, exponential distribution and typical probability distribution in video. The typical source distribution of video is from the statistical value of the video test sequences of Miss America, Suzie, and Foreman. The conventional result is the optimal one of Takishima's, Tsai's and Jeong's. Results of $L_{\min}$ and bit length vector indicate that the selection mechanism of $L_{\min}$ proposed is more flexible and can obtain a shorter average codeword length.

CONCLUSION

A new efficient construction algorithm for symmetrical RVLC independent of the Huffman code is presented. The symmetrical RVLC codeword can directly be constructed from the given symbols' dis-

Table 1 Comparison of the average codeword length and minimum code length of the conventional algorithms with the proposed algorithm

Typical probability distributions	Symmetrical RVLCs algorithm	$L_{\min}$	Bit length vector	Average codeword length (bits/symbol)
English alphabet distribution	Conventional	3	(0,0,4,2,4,4,6,4,2)	4.4646
	Proposed	3	(0,0,4,2,4,4,6,4,2)	4.4646
Uniform distribution of 32 symbols	Conventional	5	(0,0,0,0,8,6,12,6)	6.5000
	Proposed	4	(0,0,0,4,6,6,10,6)	6.2500
Typical probability distribution in video (40 symbols)	Conventional	2	(0,1,2,2,4,4,5,5,7,7,3)	4.2744
	Proposed	3	(0,0,4,2,4,4,6,4,8,6,2)	4.2659
Exponential distribution of 16 symbols	Conventional	1	(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1)	2.0000

tribution. Using the queue as the container of codes simplifies the mechanism for the codeword selection. The candidate codes which are pushed into the queue all satisfy the prefix condition, so the codeword selected mechanism has only to check the code symmetry of the codes in the queue. In addition, the way to calculate the minimum codeword length is also given.

Experimental results confirm that the proposed algorithm can provide better performance than the existing methods and is characterized by low computation complexity and easy realization.

References

Jeong, W.H., Ho, Y.S., 2003a. A New Construction Algorithm for Symmetrical Reversible Variable-length Codes from the Huffman Code. ISCIS 2003, LNCS2869, p.675-682.

Jeong, W.H., Ho, Y.S., 2003b. Design of Symmetrical Reversible Variable-length Codes from the Huffman Code. Picture Coding Symposium, p.135-138.

Takishima, Y., Wada, M., Murakami, H., 1995. Reversible variable-length codes. *IEEE Trans. Commun.*, **43**(2):158-162. [doi:10.1109/26.380026]

Tsai, C.W., Wu, J.L., 2001a. A modified symmetrical reversible variable length code and its theoretical bounds. *IEEE Trans. Inform. Theory*, **47**(6):2543-2548. [doi:10.1109/18.945265]

Tsai, C.W., Wu, J.L., 2001b. On constructing the Huffman code-based reversible variable-length codes. *IEEE Trans. Commun.*, **49**(9):1506-1509. [doi:10.1109/26.950335]