International Journal of Information & Computation Technology.
ISSN 0974-2239 Volume 4, Number 11 (2014), pp. 1063-1068

http://www. irphouse.com

Redesigning of the Construction of Symmetrical RVLCs Based On Graph Model

Radhika Goel
Student, M.Tech(ECE)
Electronics and Communication Department,
JIIT, Noida, India

Richa Gupta
Electronics and Communication Department,
JIIT, Noida, India

Abstract

This paper redesigns Takishima's symmetrical Reversible Variable Length Code (RVLC) construction algorithm based on Lin's graph model. Graph model is used to represent the prefix and suffix relationship between the codewords of subsequent levels. This algorithm minimizes the average codeword length and provides an improvement over Takishima's symmetrical RVLC algorithm.
Keywords- Reversible variable length code (RVLC), prefix-free, suffix-free, average codeword length, multimedia communication.

1. Introduction

Before the invent of Takishima's Reversible variable length code algorithm [1], Variable Length Codes (VLC) such as Huffman codes were largely used in image and video coding standards [2][3][4][5]. VLCs are very sensitive to errors occurred due to noisy environment.

In general two types of errors can arise during the digital transmission of information: propagating error and non- propagating error. Propagating error occurs at the receiver side when the decoded bit-stream contains codewords of different lengths and affects the decoding of subsequent code words and causes a loss of synchronization. If the decoded codeword matches with a valid codeword and doesn't affect the decoding of subsequent code words then it is referred to as a non-propagating error. Even a single bit error can cause propagation error, due to which the data received after the bit error position become useless and results in a serious problem. In such case, the decoder may loose synchronization. In the case of VLC being used in video coding and image coding standards, propagating error arises. Therefore in order to reduce the effect of error propagation and to improve the error resilience capability, RVLC have been used widely in multimedia applications.

With the increasing demand on multimedia and data services in third and fourth generation wireless networks which possess higher error rate and low bandwidth, RVLCs with good error resilience capability can be largely applied. RVLC can also be defined as a combination of variable length prefix followed by a fixed length suffix. They are also called affix-free codes or bi-prefix since they are both prefix-free and suffix-free. Other than error resilience property, RVLCs can also be used in searching the data because of its capability to search in two directions backward and forward direction, impossible with VLC.

RVLCs are of two types: symmetrical RVLCs and asymmetrical RVLCs. Symmetrical RVLC uses the same code table for decoding in both the forward and the backward directions since the prefix and suffix parts are identical whereas, asymmetrical RVLCs require two different decoding tables. Therefore the memory requirement for symmetrical RVLC is less than that of asymmetrical RVLC. However, asymmetrical RVLCs have more flexible codeword selection mechanism; they offer better coding efficiency than symmetrical RVLCs. MPEG-4 [3] video coding standard uses asymmetrical RVLC whereas H.263+ [4] uses symmetrical RVLCs.

Over the years many algorithms had been proposed [1][6][7][8][9][10][11] for generating RVLCs. Here we propose a modified RVLC construction algorithm using Takishima's symmetrical RVLC algorithm [1] as a baseline.

Let us first discuss a brief about RVLC, with the help of an example. Consider a DMS (Discrete Memory-less Source) emits five symbols A, B, C, D and E with the probability of occurrence as 0.30, 0.27, 0.20, 0.13 and 0.10 respectively. Table 1.1 shows the obtained codewords and the average codeword lengths for Huffman code, Takishima's symmetrical RVLC and asymmetrical RVLC. It may be observed that RVLCs being affix free, take more number of bits in code assignment, thus average codeword length is more as compared to Huffman.

Table 1.1: An example to illustrate RVLC's.

Sym.	Sym. Prob.	Huffman Code	Takishima Symm. RVLC	Takishima Asymm. RVLC
A	0.30	00	00	00
B	0.27	01	11	01
C	0.20	11	010	11
D	0.13	100	101	1010
E	0.10	101	0110	10010
Average codeword length		2.23	2.53	2.56

The organization of paper is discussed here. Section II consists of different RVLC algorithms designed so far, section III presents the mathematical analysis of the proposed algorithm, section IV gives our contribution, section V concludes the paper and section VII presents appendixes.

2. Symmetrical RVLC

Takishima's Symmetrical Algorithm [1] was the first algorithm designed for the construction of Reversible Variable Length Code. It has been seen that the number of symmetrical code words on a full binary tree at level L is given as

$m_o(L) = 2^{\lfloor (L+1)/2 \rfloor}$

Where $\lfloor Z \rfloor$ is the largest integer less than or equal to *Z*.

All of the symmetrical code words in $m_0(L)$ cannot be the target symmetrical code words due to the prefix condition for allowing instantaneous decoding; i.e. across a path from root node to the available node of codeword there will be single symmetrical codeword that can be assigned. Consequently, all of the symmetrical code words at level L that violate the prefix condition must be eliminated from $m_o(L)$. Using Takishima's symmetrical RVLC construction algorithm, RVLC may be obtained by following the below mentioned steps:

(i) The bit length vector of the target symmetrical RVLC is initialized by that of the optimum VLC i.e. Huffman code.

$$n_{rev}(i) = n(i),$$

(ii) The number of code words of length i, $n_{rev}(i)$, is restricted by the bit alignment of code words of length less than i.

If $n_{rev}(i) \leq m(i)$, $n_{rev}(i)$ is not changed.

Otherwise one bit is added to some ccde words, i.e.:

$$n_{rev}(i+l) = n_{rev}(i+l) + n_{rev}(i) - m(i) ,$$

$$n_{rev}(i) = m(i) .$$

(iii) Step (ii) is repeated until the bit allocation is finished for all code words.

Tsai[6] modifies the above algorithm, after the available candidate terms have been achieved they are arranged in the ascending order according to the maximum length of their symmetrical bit suffixes excluding the first bit of each candidate codeword then the selection procedure is carried forward. It overcomes problem in variations of the bit alignment patterns.

A more simplified algorithm was later developed by Jeong and Ho in 2003 [8], it reduces the complexity of the search method. It is based on the concept of generating half number of the required code words on the half binary tree and then applying bit inversion to get the next set of remaining symmetrical code words. It is also based on Huffman code. This change in the adaptation process was helpful in reducing the average code word length since no symmetrical code word could be missed.

In 2008 an improved algorithm [10] was developed, it states another codeword selection mechanism based on graph model to represent the prefix and suffix relationship among RVLC candidate codewords. It presents Dependency index (DI) and common dependency index (CDI) which depicts the joint influence of the set of assigned codewords on the number of available codewords of longer length which is explained in detail in section III. Based on this graph model, this paper is an effort to modify the above Takishima's Symmetrical Algorithm.

3. Mathematical Analysis

Given a(i) and a(i+1) which denote the set of i^{th} level and $(i+1)^{th}$ level codewords. Let C_{i+1} denotes a corresponding subset of a(i+1) and V_{i+1} denotes the corresponding vertex set of C_{i+1}. For a codeword C_i of a(i) and corresponding vertex v_i of $V_{a(i)}$, where $V_{a(i)}$ denotes the set of vertices whose corresponding codeword are in a(i), then the dependency index (DI) of C_i with respect to C_{i+1} is defined as the number of v_i edges whose adjacent vertices are in $V_{a(i)}$. Thus selecting codewords with smaller DIs will contribute more codewords in a(i+1). It is therefore reasonable to consider codeword with small DIs. The common dependency index (CDI) of C_i with respect to C_{i+1} is defined as the number of v adjacent vertices which are of degree 2 and in $V_{a(i)}$, where degree of the vertex v represents the number of codewords of a(i) which are not prefix-free (or suffix-free) with respect to C_{i+1}. According to [3], It is resonable to select codeword with large CDI. Fig.1 shows the prefix and suffix relationship graph for level 3 And 4. It defines the preference order of 2-tuples (DI, CDI) as follows:

$(t1, t2) > (t3, t4)$ if $t1 < t3$ or ($t1 = t3$ and $t2 > t4$)
$(t1, t2) = (t3, t4)$ if $t1 = t3$ and $t2 = t4$
$(t1, t2) < (t3, t4)$ if $t1 > t3$ or ($t1 = t3$ and $t2 < t4$)

It tries to select those available codeword that have highest (DI, CDI). Fig.1 shows the Prefix Suffix Relationship Graph (PSRG) of a(3) and a(4) after the codeword 000 is selected. The (DI, CDI) are on the left side of the corresponding vertices.

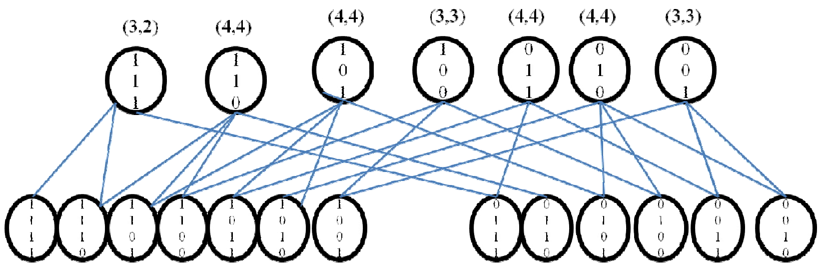

Fig.1. PSRG of a(3) and a(4)

4. Our Contribution

Hence we presents an algorithm to generate RVLCs based on Takishima’s algorithm as a baseline algorithm and Lin’s graph model. Major modification lies in appropriately choosing a codeword at initial level of smaller length which allows more codewords of longer length at subsequent levels. To satisfy prefix condition at each level of the Huffman tree we ensures that theparticular node of the available codeword should not be the children node of the assigned codewords.

Implementing the DI and CDI for Takishima’s RVLC algorithm, It is found that DI and CDI are large for 010 than 111 after selecting 000 as shown in above Fig.1.

For the analysis of RVLCs we have considered different distribution which are highly used in cryptography. Table 1.2 campares average codeword length for different RVLC construction algorithm obtained for 1-gram distribution, Tsai’s distribution and Buttigieg distribution defined for 26-english alphabet.

Table 1.2: Compares RVLC construction algorithm on the basis of average codeword length

Construction algorithm / Probability distribution	Huffman code	Takishima’s RVLC	Proposed construction algorithm based on graph model
1-gram distribution	4.155	4.7143	4.6635
Tsai's distribution	4.1557	4.6965	4.6487
Buttigieg distribution	4.2045	4.7753	4.7279

5. Conclusion

Here, we presents an algorithm to generate improved RVLC by using Takishima’s algorithm as a baseline and applying Lin’s graph model on it . This algorithm reduces the average codeword length which increases the coding efficiency and the size of the buffer required decreases.

6. References

[1] Y Takishima, M. Wada, and H. Murakami, Reversible variable length codes, IEEE Transaction on Communication, vol. 43, pp. 158–162, Feb–Apr. 1995

[2] Wallace G.K., “The JPEG Still Picture Compression Standard”, IEEE Transactions on Consumer Electronics, Vol. 38, No. 1, Feb. 1992.

[3] Q. Chen and K. Subbalakshmi, "An Intergrated Source Channel Decoder for MPEG-4 Coded Video," *IEEE Semi-Annual Vehicular Technology Conference,* Oct 2003.
[4] Study Group 16-Contribution, "Draft for "H.263++" Annexes U, V, and W to Recommendation H.263", International Telecommunication Union, Microsoft Corporation, United States, Study Period 1997 – 2000.

[5] I.-T. Recommendation, "JPEG Standard (JPEG ISO/IEC 10918-1)," vol. T.81, pp. 1-186, 1993.
[6] 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]
[7] Ksenija Lakovic and John Villasenor, An algorithm for construction of efficient fix-free codes, IEEE Communication. Letters, vol.7, No.8, pp.391–393, August 2003
[8] Jeong, W.H, Ho, Y.S., 2003b. Design of Symmetrical Reversible Variable-length Codes from the Huffman Code. Picture Coding Symposium, p.135-138
[9] 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), Received Dec. 12, 2005; revision accepted Feb. 22, 2006.
[10] Chia-wei Lin, Ja-Ling Wu and Yuh-jue Chuang,Two algorithms for constructing efficient Huffman-code based reversible variable length codes, IEEE Transaction on communication ,vol. 56,no.1,January 2008.
[11] V. Buttigieg, .Variable-Length Error-Correcting Codes,. Ph.D. Thesis, Department of Electrical Engineering, University of Manchester, 1995.
[12] Alan G Konheim, "One gram probability distribution", Cryptography – A Primer John Wiley, pp 16.