ICEIC'95

# Symmetric Huffman Coding

*Kyong-Sik Om, Woong-Hee Lee, Se-youn Jung, Jae-Ho Chung*

Dept. of Electronic Engineering, College of Engineering, Inha Univ.
Address : 253 Yonghyondong, Inchon, 402-751, KOREA
TEL : 82-032-860-7415
FAX : 82-032-868-3654
E-Mail : ks-om@dragon.inha.ac.kr

*Index terms - Huffman code, symmetric code, variable-length coding, variance, maximum code length ML, average code length AL.*

## Abstract

Symmetric Huffman codes characterized by the feature that first bit of a code is 0 or 1 having the probability of about 0.5 and the rest bits have symmetry were analyzed. We have analyzed the performance of symmetric Huffman codes based on the average code length *AL*, maximum code length *ML*, codeword variance. Finally, we will discuss the method to utilize the merits of the proposed symmetric Huffman coding scheme in realization of decoder in hardware.

## I. INTRODUCTION

The Huffman coding is optimal binary coding scheme. Consequently, a variety of modified Huffman codes have been developed and applied to digital signal processing such as image and speech signal processings. This paper proposes another coding scheme based on the Huffman coding and symmetry.

We have named the new designing method as symmetric Huffman coding which is characterized by the feature that first bit of a code is 0 or 1 having the probability of about 0.5 and the rest bits have symmetry.

The paper's organized as follows. In section II, we review Huffman coding. In section III, we discuss proposed symmetric Huffman coding. In section IV, we analyze the effect of maximum code length. In section V, we show experimental results of image data. Finally in section VI, the conclusions are stated.

## II. HUFFMAN CODING

Huffman coding is a variable-length code. The advantage of a code in which the message symbols are of variable length is that sometimes the code is more efficient in the sense that to represent the same information we can use fewer bits on average. If the probabilities of the frequencies of occurrence of the individual symbols are sufficiently different, the variable-length encoding can be significantly more efficient than block encoding [1].

From now we will discuss about the characteristics and theorems of Huffman code.

For any instantaneous code ( in other word, prefix condition code ) the following Theorem 1 must be satisfied.

*Theorem 1 (Kraft inequality):* A necessary and sufficient condition for the existence of an instantaneous code $S$ of n symbols $S_i$ (i=1, . . . , n) with encoded words of length $l_1 \le l_2 \le l_3 \le \cdots \le l_n$ is

$$\sum_{i=1}^{n} \frac{1}{r^{l_i}} \le 1 \tag{1}$$

where $r$ is the radix (number of symbol) of the alphabet of the encoded symbols.

*Definition 1:* We define average code length $AL$ and maximum code length $ML$ as,

$$AL = \sum_{i=1}^{n} P_i l_i \tag{2}$$

$$ML = \max_i (l_i) \tag{3}$$

where $P_i$ is the probability of the ith symbol, $l_i$ is the length of the ith symbol, we can say that average code length $AL$ is efficiency of the code words.

*Theorem 2:* Let $S$ be a discrete source consisting of letters $a_1, a_2, \_ , a_n$ with probabilities of $P_1 \ge P_2 \ge P_3 \ge \_ \ge P_n$. Let $C = \{c_1, c_2, c_3, \_ , c_n\}$ be a code for the source $S$ and $|c_1|, |c_2|, |c_3|, \_ , |c_n|$ the lengths of its codewords. As is well known, for any discrete source $S$ with entropy $H$, there exists a prefix code with average codeword length $AL$ bounded by

$$AL < H + 1 \tag{4}$$

The code with the minimum average codeword length can be constructed using the Huffman procedure [1][2].

*Theorem 3:* The entropy supplies a lower bound on the average code length $AL$ for any instantaneous decodable system

$$H \le AL \tag{5}$$

*Lemma 1:* The Huffman encoding process is not unique, but in any cases the average length of the encoding of messages will be the same.

*Definition 2:* We define the variance of codes *Var*.

$$Var = \sum_{i=1}^{n} P_i (l_i - AL)^2 \tag{6}$$

*Theorem 4:* The more variable the $l_i$, the more harm (or good) the errors in the estimates of the $P_i$ could cause in the average of the symbol length.

*proof* : see pp. 74-77 in [1].

As stated in Lemma 1 there can be varieties of Huffman encodings. From Theorem 4 it is desirable to select the one with the smaller variance.

## III. SYMMETRIC HUFFMAN CODING

The proposed symmetric Huffman coding procedure is like as following.

step 1. Arrange source symbols $S = \{ S_1, \cdots, S_n \}$ according to the probability magnitude of $P_1 \geq P_2 \geq P_3 \geq \cdots \geq P_n$.

step 2. Construct Huffman code with the odd symbols.

step 3. For odd symbols the Huffman code is constructed by adding 0 to the head of the constructed half Huffman code, and 1 for even symbols.

step 4. If the total number of source symbols is odd, one Huffman codeword can be shortened by one bit (see example 2).

As proposed symmetric Huffman code is also uniquely decodable and instantaneous code, it satisfies the inequalities of (4) and (5).

*Definition 3:* The *external path length* of a tree is the sum of the lengths of all paths from the root to a leaf ; it will be denoted by *epl*.

$$epl = \sum_{i=1}^{n} l_i \qquad (7)$$

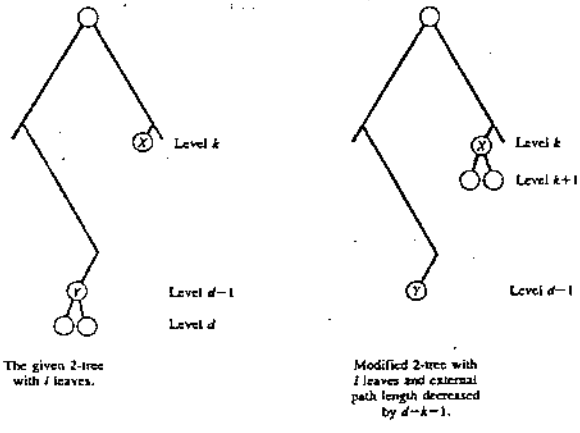


Fig. 1. Decreasing external path length.

*Lemma 2:* Among 2-trees with $l$ leaves, the *epl* is minimized only if all the leaves are on at most two adjacent levels.

*proof :* Suppose we have a 2-tree with depth $d$ that has a leaf X at level k, where $k \leq d-2$. We will exhibit a 2-tree with the same number of leaves and lower *epl*. Choose a node Y at level d-1 that is not a leaf, remove its children, and attach two children to X. (See Fig. 1. for an illustration.) The total number of leaves has not changed. The *epl* has been decreased by 2d+k, because the paths to the children of Y and the path to X are no longer counted, and increased by 2(k+1)+d-1 = 2k+d+1, the sum of the lengths of the paths to Y and the new children of X. There is a net decrease in the *epl* of 2d+k-(2k+d+1) = d-k-1 > 0, since $k \leq d-2$. ∎

*Lemma 3:* If source number is large and source probabilities are nearly equal to each other, the maximum code length of symmetric Huffman code is shorter or at least equal to that of Huffman code.

*proof :* We can easily see from Lemma 2.

*Lemma 4:* If source number is large and source probabilities are nearly equal to each other, the variance of symmetric Huffman code is smaller than that of Huffman code.

*proof :* From Lemma 2 the fact that $epl = \sum_{i=1}^{n} l_i$ is decreased also means that $\sum_{i=1}^{n} l_i^2$ is also decreased.

$$\begin{aligned}
Var &= \sum_{i=1}^{n} [P_i \ (l_i - AL)^2] \\
&= \sum_{i=1}^{n} [P_i \ (l_i - \sum_{i=1}^{n} P_i l_i)^2] \\
&\cong P \sum_{i=1}^{n} (l_i - P \sum_{i=1}^{n} l_i)^2 \ , \ (P_i \simeq P) \\
&= P \sum_{i=1}^{n} (l_i^2 - 2P l_i \sum_{i=1}^{n} l_i + P^2 (\sum_{i=1}^{n} l_i)^2) \\
&= P \, [ \sum_{i=1}^{n} (l_i^2) - 2P (\sum_{i=1}^{n} l_i)^2 + nP^2 (\sum_{i=1}^{n} l_i)^2 ] \\
&\cong P \, [ \sum_{i=1}^{n} (l_i^2) - 2P (\sum_{i=1}^{n} l_i)^2 + P (\sum_{i=1}^{n} l_i)^2 ] \ , (P \simeq 1/n) \\
&= P \, [ \sum_{i=1}^{n} (l_i^2) - P (\sum_{i=1}^{n} l_i)^2 ] \\
&\cong P \, [ \sum_{i=1}^{n} (l_i^2) - \frac{1}{n} (\sum_{i=1}^{n} l_i)^2 ] \\
&\geq 0
\end{aligned} \qquad (8)$$

This can be verified easily from the *Cauchy-Schwarz inequality* that is like as following if $b_1 = b_2 = \cdots = b_n = 1$.

$$(a_1^2 + a_2^2 + \cdots a_n^2)(b_1^2 + b_2^2 \cdots + b_n^2) \geq (a_1 b_1 + a_2 b_2 + \cdots + a_n b_n)^2 \qquad (9)$$

The Cauchy-Schwarz inequality means that the effect of the variation of left side term is greater than or equal to that of right side one.

Both $\sum_{i=1}^{n} (l_i^2)$ and $\frac{1}{n} (\sum_{i=1}^{n} l_i)^2$ are decreased if the coding scheme is changed from Huffman procedure to symmetric Huffman procedure, but as $\sum_{i=1}^{n} (l_i^2)$ is decreased much more than $\frac{1}{n} (\sum_{i=1}^{n} l_i)^2$ does, so in general the codeword variance of symmetric Huffman code is smaller than that of Huffman code if source number is large and source probabilities are nearly equal to each other. ∎

The lower codeword variance means it is robust to noise as seen in Theorem 4.

*Lemma 5:* Even if there are many cases of symmetric Huffman codes constructed from the half symbols - from Lemma 1 the average code length is equal in any case -, the final symmetric Huffman codes also have the same average code length.

*proof* : If $AL'$ is the average code length of half Huffman codes and $AL$ is the final average code length of symmetric Huffman code,

$$AL' = \sum_{i=1}^{n} P_i l_i \quad \textit{(where } i \textit{ is odd number)} \tag{10}$$

$$AL = \sum_{i=1}^{n} (P_i + P_{i+1})(l_i + 1) \tag{11}$$

$$= \sum_{i=1}^{n} P_i (l_i + 1) + \sum_{i=1}^{n} P_{i+1}(l_{i+1} + 1), \quad (l_i = l_{i+1})$$

$$= \sum_{i=1}^{n} P_i l_i + \sum_{i=1}^{n} P_i + \sum_{i=1}^{n} P_{i+1} l_{i+1} + \sum_{i=1}^{n} P_{i+1}$$

$$= \sum_{i=1}^{n} P_i l_i + \sum_{i=1}^{n} P_{i+1} l_{i+1} + 1$$

$$= AL' + AL'' + 1$$

*(where i is odd number).*

Thus, if $AL'$ is equal in every cases, the final average code length $AL$ is also equal, as the other half Huffman average code length $AL''$ is also equal from Lemma 1. ∎

From Lemma 5 we can consider only the half Huffman codes in constructing symmetric Huffman code. So we can say that how good a half Huffman code is a ultimate goal in Symmetric Huffman code.

Coding complexity is also a big issue for signal compression. The symmetric Huffman coding procedure satisfies the following Property 1.

*Property 1*: Symmetric Huffman coding procedure is simple and faster than Huffman coding procedure as we can consider only the half sources.

Here we show two simple cases for the symmetric Huffman procedure.

*( example 1 )* Simple even source number case.

If the source probabilities are 0.3, 0.2, 0.15, 0.15, 0.1, 0.1, then we can design symmetric Huffman code as shown in Fig 2.

| | | |
|---|---|---|
| 0.3 | 0 0 | $S_1$ : 00 |
| | 1 0 | $S_2$ : 10 |
| 0.15 | 0 10 | $S_3$ : 010 |
| | 1 10 | $S_4$ : 110 |
| 0.1 | 0 11 | $S_5$ : 011 |
| | 1 11 | $S_6$ : 111 |
| (a) | | (b) |

Fig 2. The symmetric Huffman procedure and final code of example 1.
(a) Huffman code constructed by half sources.
(b) The final code.

*( example 2 )* Simple odd source number case.

If the source probabilities are 0.3, 0.15, 0.15, 0.15, 0.1, 0.1, 0.05, then we can design symmetric Huffman code as shown in Fig 3. As the source number isn't a even one we can shorten a tag bit corresponding to the virtual code.

| | | |
|---|---|---|
| 0.3 | 0 0 | $S_1$ : 00 |
| | 1 0 | $S_2$ : 10 |
| 0.15 | 0 10 | $S_3$ : 010 |
| | 1 10 | $S_4$ : 110 |
| 0.1 | 0 110 | $S_5$ : 0110 |
| | 1 110 --> 1 11 | $S_6$ : 111 |
| 0.05 | 0 111 | $S_7$ : 0111 |
| | 1 111 (virtual code) | |
| (a) | | (b) |

Fig 3. The symmetric Huffman procedure and final code of example 2.
(a) $S_5$ can be shorten by 1 bit considering the virtual code.
(b) The final code ( see $S_6$ ).

## IV. THE EFFECT OF ML

The encoder of VLC (Variable Length Code) is easy to construct in hardware. The source symbol can be matched by look-up table of codewords. But the decoder is not so easy.

There is two possible ways to design a decoder of VLC. One is a *serial processing method* and the other is a *parallel processing method*. In a serial processing method each bit is put by tree searching method in code-tree and final symbol can be found, in practical by adjusting the address of ROM in which logic gate and tree structure are located. However this serial processing method is hard to implement in real time processing for the requirements of high clock rate.

To overcome this problem, the parallel processing method is presented which can be operated in low clock-rate [6]. In parallel processing method every possible bit-pattern is saved and the input bit-stream can be matched to the corresponding symbol. This method is good for real-time decoding system. But as the maximum code length increases, the requiring memory is increased exponentially as memory cost function of definition 4. So shorter maximum code length is desirable.

*Definition 4:* We define memory cost function as following

$$M = 2^{ML} \tag{12}$$

where $ML$ is the maximum code length.

From these view points we can find the proposed symmetric Huffman code is appropriate for establishing serial and parallel decoding hardware from Lemma 3.

Also using the Property that proposed symmetric Huffman codewords have the symmetry - first bit of a code is 0 or 1 having the probability of about 0.5, and the rest ones have symmetrical form between odd codes and even ones - we can reduce memory requirements more in parallel processing decoder. For example we can design a parallel processing decoder memory like Fig 4. then the final symbol is like (13) and we can see Property 2, i.e.,

source symbol = matched codeword except first bit + first bit (13)

*Property 2:* The memory cost function of symmetric Huffman code can be expressed like as following.

$$M = 2^{ML-1} \quad (14)$$

where $ML$ is the maximum code length of symmetric Huffman code.

| codeword except first bit | symbol |
|---|---|
| $C_1$ | $S_1$ |
| $C_3$ | $S_3$ |
| $C_5$ | $S_5$ |
| ... | ... |
| $C_n$ | $S_n$ |

Fig 4. Simple example of parallel processing decoder memory using the symmetry of symmetric Huffman code.

## V. Experimental Results

We have tested proposed symmetric Huffman procedure at image data (image size : 512×512, 256 graylevels). We selected the procedure used by Lu and Chen for generating the Huffman code which produces increasing code length according to source symbol $S = \{ S_1, \cdots, S_n \}$ where probability magnitude is $P_1 \geq P_2 \geq P_3 \geq \cdots \geq P_n$ [9].

Table 1 shows the results. It shows that $AL$ of symmetric Huffman procedure is slightly lower than that of Huffman procedure, but the efficiency is nearly 99 %. $ML$ of symmetric Huffman procedure is always smaller than that of Huffman procedure, which tells us very important merits as discussed in section IV. Finally the $Var$ is higher or lower than that of Huffman code depending on the statistics of the source. In general the histograms of Baboon, Boat, Tiffany spread out somewhat uniform, but the ones of Lena, Splash, Pepper are crowded to a few sides, and this fact seems to be related to $Var$.

For Pepper image in a parallel decoder we can reduce memory size to be 1.5625% from the fact that normal Huffman coding : $M = 2^{23}$ vs. symmetric Huffman coding : $M = 2^{18-1}$. This is a very interesting results !.

Table 1. Test Results.

| 512 × 512 | | $AL$ | efficiency | $Var$ | $ML$ |
|---|---|---|---|---|---|
| Baboon | N | 7.521904 | 99.937409 | 0.687676 | 23 |
| | S | 7.526615 | | 0.654680 | 21 |
| Boat | N | 7.674099 | 99.792446 | 0.752015 | 20 |
| | S | 7.690060 | | 0.693476 | 17 |
| Tiffany | N | 6.753498 | 98.505217 | 1.071107 | 23 |
| | S | 6.855980 | | 0.920283 | 22 |
| Lena | N | 7.626995 | 99.909548 | 0.574006 | 23 |
| | S | 7.633900 | | 0.604721 | 21 |
| Splash | N | 6.923027 | 98.769587 | 2.618350 | 19 |
| | S | 7.009270 | | 2.699987 | 14 |
| Pepper | N | 7.515961 | 98.934701 | 1.376653 | 23 |
| | S | 7.596943 | | 1.569925 | 18 |

where N : Normal Huffman coding procedure.
S : Symmetric Huffman coding procedure.
efficiency : % $AL$ of symmetric Huffman codes compared with that of normal Huffman codes.

## VI. Conclusions

We have analyzed the performance of symmetric Huffman codes based on the average code length $AL$, maximum code length $ML$, codeword variance. The proposed symmetric Huffman codes are characterized by the features that first bit of a code is 0 or 1 having the probability of about 0.5 and the rest bits have symmetry.

As we have seen, proposed symmetric Huffman code shortens the maximum code length.

Finally we have discussed the method to utilize the merits of the proposed symmetric Huffman coding scheme in realization of decoder in hardware.

The proposed scheme can be extended to radix r ($r \geq 3$) by adjusting step 2 : selecting rth source and step 3 : adding 0,1,2,...,r.

If the source number is large and there is not big difference of source probabilities, the proposed symmetric Huffman coding procedure will be useful as shown in Lemmas 3, 4 and section V. Experimental results and the $AL$ of symmetric Huffman codes are nearly equal to that of normal Huffman code as the source number increases. But we think it is necessary that deeper analysis must be performed. For example, the upper bound and lower bound compared to Huffman code and the effect of odd source number must be deeply studied.

## References


[1] R. W. Hamming, *Coding and Information Theory*, Prentice-Hall, 1986.

[2] D. A. Huffman, " A method for the construction of minimum redundancy codes," *Proc. IRE*, Vol. 40, No. 2, pp.1098-1101, Mar. 1952.

[3] Sara Baase, *Computer Algorithms*, Addison-Wesley, 1988.

[4] Rafael C. Gonzalez, Richard E. Woods, *Digital Image Processing*, Addison-Wesley, 1992.

[5] *Numerical Recipes in C*, Cambridge university press, 1992

[6] S. B. Choi, M. H. Lee, C. Y. Park " A Jong-Nang VLC CODEC For Image Data Compression ", in *Proc. Korean Signal Processing Conference* (Inchon, Korea), Vol.7, No.1, pp.198-204, 1994.

[7] W. J. Kim, Y. H. Kim, S. D. Kim, " VLD Implementation using Codeword Partitioning", in *Proc. Korean Signal Processing Conference* (Inchon, Korea), Vol.7, No.1, pp.646-650, 1994.

[8] R. M. Capocelli, A. D. Santis, G. Persiano, " Binary Prefix Codes Ending in a "1" ", *IEEE Trans. on Information Theory*, Vol. 40, No.4, July, 1994.

[9] M. Lu and C. Chen, "A Huffman-type Code Generator with Order-N Complexity," *IEEE Trans. on Acoustics, Speech and Signal Processing*, Vol.38, No.9, Sept. pp. 1619-1626, 1990.