Design and Development of a Secured Data Coding and Compression Method for 2D Digital Images

B. A. Weyori,
Dept. of Computer Engineering,
K. N. U. S. T- Kumasi, Ghana.
Dept. of Computer Science,
U. D. S, Navrongo, Ghana

P. N. Amponsah
Faculty of Information,
Communication Science and Technology,
C. U. C. G, Fiapre-Sunyani, Ghana

P. Yeboah
Faculty of Information,
Communication Science and Technology,
C. U. C. G, Fiapre-Sunyani, Ghana

ABSTRACT

Image compression addresses the problem of reducing the amount of data required to represent a digital image. The underlying basis of the reduction process is the removal of redundant data. In this paper, we carry out a study on the digital image coding and compression, and proposed a digital image coding scheme which is a development of the Huffman's data coding algorithm. The proposed coding algorithm is an efficient data coding and compression method that offers high-speed bit data transmission and a secured image processing. We demonstrate the efficiency of the proposed coding algorithm by showing that it generates optimal prefix that is simpler and generates shorter code length per character and less average code length compared to the Traditional Huffman's coding algorithm. The design of an encoder and a decoder pair is carried out in MATLAB programming language and the results of the MATLAB simulation demonstrate the security ability of the proposed image coding scheme.

Keywords

Compression, Coding, Redundant, Prefix, Huffman's Algorithm, Code length, Encoder, Decoder.

1. INTRODUCTION

Digital image processing consists of many topics, including digital image transforms, digital image filtering and enhancement, digital image compression, edge detection, image segmentation, and shape description. Digital image processing has been extensively used in desktop publishing, medical imaging, military target analysis, manufacture automation control, machine vision, geo-physical imaging, graphic arts and multimedia [1]. Most of these applications depend on the availability of compact and inexpensive hardware delivering the required high performance so that very large scale integrated (VLSI) technology is of vital importance for digital image processing [2, 3].

These processing involves the development of many signal processing algorithms and implementations. The VLSI implementation of digital image processing systems requires high-speed and low-power techniques. Further, a secure transmission of the digital image through the computer network needs to be considered in some applications [4, 5, 6]. Digital image data are so large that to extensive and effective store and work with or transmit them, they have to be compressed to save storage space and efficiently coded to secure the image data. The bigger the resolution of the image, the brighter and better the image quality which is as a result of the large image data, hence the need for the image data to be compressed.

The natural technology for handling the increased spatial resolutions of today's imaging sensors and evolving broadcast television standards needs to be compressed and made easy for transmission. Furthermore, image compression plays a major role in many important and diverse applications, including Tele-videoconferencing, remote sensing (the use of satellite imagery for weather and other earth-resource applications), document and medical imaging, facsimile transmission (FAX), and the control of remotely piloted vehicles in military, space, and hazardous waste management applications. In short, an ever-expanding number of applications depend on the efficient manipulation, storage, and transmission of binary, gray-scale, and color images [1, 7].

Data compression is often referred to as encoding as a general term is encompassing any special representation of data which satisfies a given need [8]. Thus, data compression may be viewed as a branch of information theory in which the primary objective is to minimize the amount of data to be transmitted. A simple characterization of data compression is that it involves transforming a string of character in some representation (such as ASCII) into a new string (of bits, for example) which contains the same information but whose length is as small as possible. Transforming data this way, image compression technique is significantly applied in data transmission and data storage

One important method of transmitting messages is to transmit in their place sequence symbols. If there are messages which might be sent than there are kinds of symbols available, then some of the messages must use more than one symbol. If it is assumed that each symbol requires the same time for transmission, then the time for transmission of the message is directly proportional to the number of symbols associated with the message [9, 10].

Encoding data (message) is the process of coding data or converting the data into codes in order to keep it secret for sending or storage. The string of character comprising the information is transformed into codes that are a coded form of the information or data. In Huffman's algorithm, once the Huffman's tree is constructed, the Huffman code is generated by placing the value of "1" to every right hand side of the tree's arc and "0" to every left hand side of the tree's arc, data may be encoded simply by replacing each symbol with its code.

Robert Mario Fano and Claude Elwood Shannon in 1948 worked together on “The Mathematical Theory of Communication.” The work focused on how best to encode the information a sender wants to transmit. The Shannon-Fano coding is constructed as follows: the source message a(i) and their probabilities p(a(i)) are listed in order of non-increasing probability. This list is then divided in such a way as to form two groups of as nearly equal total probabilities as possible. Each in the first group receives a 0 as the first digit of its codeword; the messages in the second half have codeword beginning with 1. Each of these groups is then divided according to the same criterion and additional code digits are appended [11, 12].

The use of Huffman codes [13] affords compression, because distinct symbols have distinct probabilities of incidence. This property is used to advantage by tailoring the code lengths corresponding to those symbols in accordance with their respective probabilities of occurrence. Symbols with higher probabilities of incidence are coded with shorter codeword, while symbols with lower probabilities are coded with longer codeword. However, longer codeword still show up, but tend to be less frequent and hence the overall code length of all codeword in a typical bit string tends to be smaller due to the Huffman coding.

2. HUFFMAN’S ALGORITHM

Huffman’s algorithm is a technique for compressing data. But this technique is a greedy algorithm, which always makes the choice that look best at the moment. The greedy algorithm makes a locally optimal choice in the hope that this choice will lead to a global optimal solution. Huffman’s greedy algorithm looks at the occurrence of each character, and it is a binary string in optimal way.

David A. Huffman in 1952 published a paper on “A Method for the Construction of Minimum-Redundancy Codes.” Huffman’s algorithm expressed graphically, takes as input a list of non-negative weights (w[1],….w[n]) and constructs a full binary tree (a binary tree is full if every node has either zero or two children whose leaves are labeled with the weights [14] .

2.1 Huffman’s Algorithm

```
HUFFMAN(C)

1. n=[C]
2. Q=C
3. For i=1 to n-1
4. Do Z=ALLOCATE_NODE()
5. X=LEFT[Z]=EXTRACT_MIN(Q)
6. Y=LEFT[Z]=EXTRACT_MIN(Q)
7. F[Z]=F[X] +F[Y]
8. INSERT(Q, Z)
9. RETURN EXTRACT_MIN(Q)
```

The algorithm is used to construct the Huffman’s tree which helps in the generation of the codes of the data.

3. PROPOSED ALGORITHM FOR DATA CODING

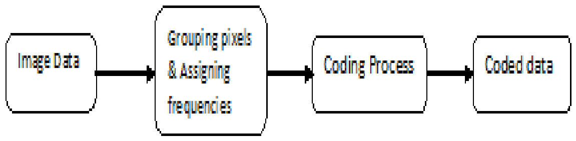
In order to achieve an efficient and effective encoding scheme that saves storage space and has a high speed bit transmission rate through a transmission channel the secured image compression and encoding technique is developed.

This algorithm is a development from the Huffman’s method of construction of minimum redundancy codes, which is a technique for compressing image data to yield more efficient codes than the Huffman’s technique.

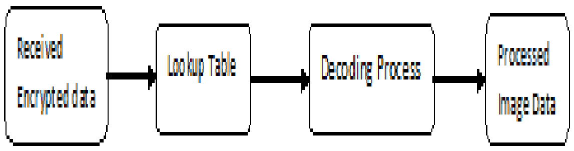
The improved code generation method derived from the Huffman’s algorithm is a method that aims at compressing data and saving about 5% to 70% more than the Huffman’s data coding method.

The scheme consists of a simplified encoder and a decoder pair for the encryption and decryption of image data. The image pixels are grouped according to the bits of the image data and their frequencies of occurrence assigned to them. The encoding process uses the frequencies or probabilities to construct a top-button tree from which the string codes are generated. The codes are arranged into an encrypted bit strings by a certain order. The security measure is that an unauthorized listener does not know that the bit strings are not binary set of numbers but encrypted strings generated from a tree. Hence, only the encoder with a lookup table with the corresponding codes and their frequencies can decode the bit stings back to the processed image data. This has a high level of security and very useful for vital and valuable image data coding.

3.1 Encoder



3.2 Decoder



3.3 APPLICATION OF THE PROPOSED ALGORITHM

The new method uses the greedy method to generate a tree which enhances the code generation of various characters in a message based on the frequencies or the probabilities of occurrence of the various characters. The total probability of the characters in the message is 1:

$$P_T = \sum_{i=1}^{n} P(i) = 1 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

The percentage weight of the encoded message is calculated by

$$\% \text{ W} = \sum_{i=1}^{n} P_i . N_i \times 100 \text{..........................(2)}$$

Where L_i is the code length of the ith character and P_i is the probability of occurrence of the ith character. Hence the less the percentage weight the more efficient the coding method and the faster the transmission rate of the message.

3.3 Algorithm of Image Data Coding

PROCEDURE [A]

```
1.  n=[A]
2.  for  i=1 to n-1 do
3.  for  j=n down-to i+1 do
4.  if (A[j-1] < A[j]) then
5.  Temp=A[j-1]
6.  A[j-1] = A[j]
7.  A[j]=Temp
8.  Right-Child[Z]=A[1]
9.  Left-Child[Z]=A[2]
10. Root=Z=(P(A[1])+P(A[2]))
11. Insert (Z, A[1], A[2])
12. R=3
13. i=1
14. While (R $\leq$ n) Do
15. Parent=A[i]
16. If((R=n) AND (n mod 2)$\neq$0) then
17. Right-Child(A[1])=A[R]
18. Insert (A[R])
19. Elseif ((R=n) AND (n mod 2)=0) then
20. Right-Child[Z]=A[R]
21. Left-Child[Z]=A[R+1]
22. Else
23. Right-Child[Z]=A[R]
24. Left-Child[Z]=A[R+1]
25. Insert(A[R], A[R+1])
26. R=R+2
27. End
```

Suppose we have characters likes; a, b, c, d, e, f, g and h with their associated probabilities as: P (a), P (b), P(c), P(d), P(e), P(f), P(g), P(h) respectively, using the new method, encode the data into binary string of 0s and 1s.

1. We first arrange the probabilities in descending order

   P (a)>P (b)>P(c)>P (d)>P (e)>P (f)>P (g)>P (h)……………...1

2. We develop the tree using the top-bottom approach which is a greedy method

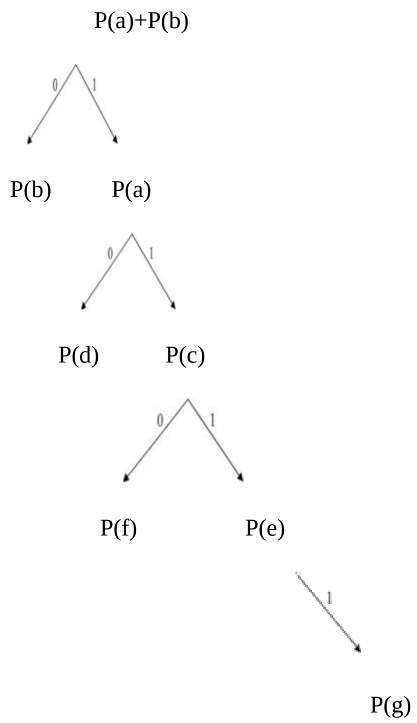


Figure 1: Code Generation Tree

3. Code generation from the tree (every move to the left is assigned a 0 and that to the right is assigned a 1).

Table A: Procedure for coding data

Character	Probability	Code	Code length
A	P(a)	1	1
B	P(b)	0	1
C	P(c)	11	2
D	P(d)	10	2
E	P(e)	111	3
F	P(f)	110	3
G	P(g)	1111	4

The total length of a tree to be constructed depends on the total number of characters in the message to be encoded.

If the number of characters in a message is odd then the length of the tree is calculated as:

$$L_T = \frac{(N+1)}{2} \text{...................................3}$$

4. ILLUSTRATION OF THE PROPOSED ENCODING SCHEME

If the number of characters in a message is even then the length of the tree is calculated as:

$$L_T = \frac{N}{2} \text{...} 4$$

A part of the cameraman grayscale image is used to illustrate the applicability of the proposed secured data coding and compression method. This method is much secured and error free. The algorithm uses string of zeros and ones to encode the image data. The image data when encrypted cannot be corrupted since the coded data is not binary but strings hence no error messages can damage the strings.

Table 1: A table showing an 8×8 part of cameraman image data

107	108	107	106	99	101	102	107
109	106	108	107	103	102	103	110
107	106	110	110	106	107	107	120
106	107	108	108	108	108	108	114
105	108	109	109	108	106	107	110
105	108	109	110	108	108	109	109
108	109	109	109	108	109	110	109
107	107	108	108	107	110	110	109

Suppose alphabets are used to represent the individual image pixels and their probabilities are assigned to them. Since this proposed coding scheme uses the number of occurrence of individual pixels to code the data. A part of the image data is used to demonstrate the applicability of the proposed coding algorithm.

Table 2: Example 1

Image Pixels	Character	Probability
99	A	0.05
101	B	0.09
106	C	0.12
107	D	0.13
108	E	0.16
111	F	0.45

At the first processing stage the frequencies or probabilities are arranged in descending order. Next, a tree is generated; every movement to the left of the tree is assigned a ‘zero’ and that to the right of the tree is assigned a ‘one’.

Table 2(a): Code generated using the newly developed coding algorithm

Character	Code
A	110
B	111
C	10
D	11
E	0
F	1

Table 2(b): Calculation of code length and average code length per character computation

Character	Probability	Code	Code length	P(i)L(i)
A	0.05	110	3	0.15
B	0.09	111	3	0.27
C	0.12	10	2	0.24
D	0.13	11	2	0.26
E	0.16	0	1	0.16
F	0.45	1	1	0.45

L_{av}= 1.53

Table 2(c): Huffman Method of Code Generation and average code length per character computation

Character	Probability	Code	Code length	P(i)L(i)
A	0.05	1100	4	0.20
B	0.09	1111	4	0.36
C	0.12	100	3	0.36
D	0.13	101	3	0.39
E	0.16	111	3	0.48
F	0.45	1	1	0.45

L_{av}= 2.24

Suppose alphabets are used to represent the individual image pixels and their probabilities are assigned to them. Since this proposed coding scheme uses the number of occurrence of individual pixels to code the data, it is difficult understand and decrypt the code. A more practical example is used here, the image data is larger than the 8×8 2D image data shown above, just a part of the data has been sampled and used for the demonstration of the coding process of the proposed scheme in the example shown below.

Table 3: Example 2

Image Pixels	Character	Probability
93	A	0.21
95	B	0.18
99	C	0.14
101	D	0.13
103	E	0.10
105	F	0.08
107	G	0.07
109	H	0.04
110	I	0.03
111	J	0.02

Example 3: Code generated using the newly developed coding algorithm

Character	Probability
A	1
B	0
C	11
D	10
E	111
F	110
G	1111
H	1110
I	11111
J	11110

Table 3(b): Calculation of code length and average code length per character computation

Character	Probability	Code	Code length	P(i)L(i)
A	0.21	1	1	0.21
B	0.18	0	1	0.18
C	0.14	11	2	0.28
D	0.13	10	2	0.26
E	0.10	111	3	0.30
F	0.08	110	3	0.24
G	0.07	1111	4	0.28
H	0.04	1110	4	0.16
I	0.03	11111	5	0.15
J	0.02	11110	5	0.10

L_{av}= 2.16

Table 3(c): Huffman Method of Code Generation and average code length per character computation

Character	Probability	Code	Code length	P(i)L(i)
A	0.21	11	2	0.42
B	0.18	011	3	0.54
C	0.14	001	3	0.42
D	0.13	000	3	0.39
E	0.10	101	3	0.30
F	0.08	0101	4	0.32
G	0.07	0100	4	0.28
H	0.04	1000	4	0.16
I	0.03	10011	5	0.15
J	0.02	10010	5	0.10

L_{av}= 2.72

The total code length per probability of the proposed method for example 1 is 1.53 and 2.16 for example 2 as compared to 2.24 and 2.72 respectively for that of the Huffman's algorithm, meaning approximation about 70% reducing in example 1 and approximately about 50% reduction in storage space in example 2.

5. EXPERIMENTAL RESULTS

In order to test the performance of the proposed algorithm, experiments were conducted on the cameraman image, the pout image and the aircraft image shown in Figure 1, Figure 2 and Figure 3 with the encrypted image displayed. Intensive simulations on MATLAB 7.40(R2007a) platform were carried out on several versions of the images with 8bits. Figure 1(b), Figure 2(b) and Figure 3(b) shows the results achieved by the proposed algorithm.

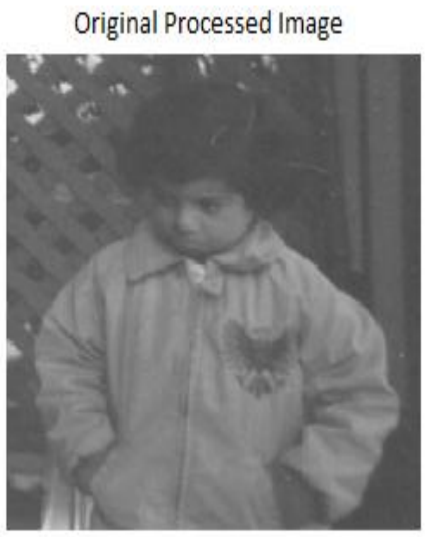


Figure 2: Original Image of Pout

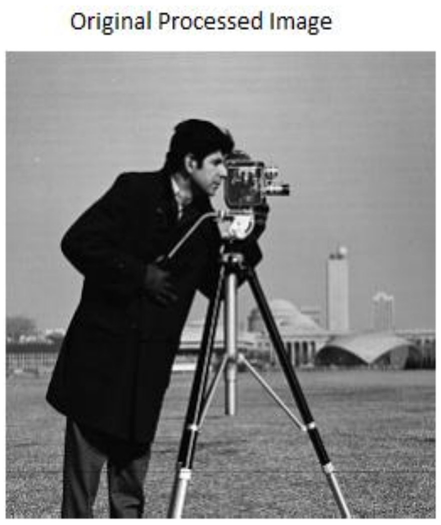


Figure 4: Original Image of Cameraman

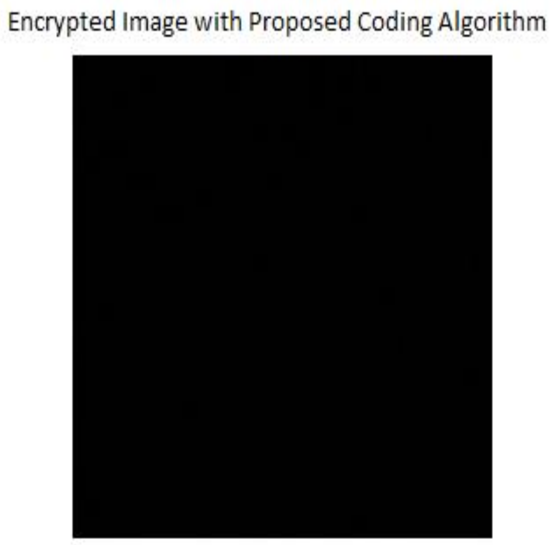


Figure 3: Encrypted image pixels using the newly developed secured image coding scheme

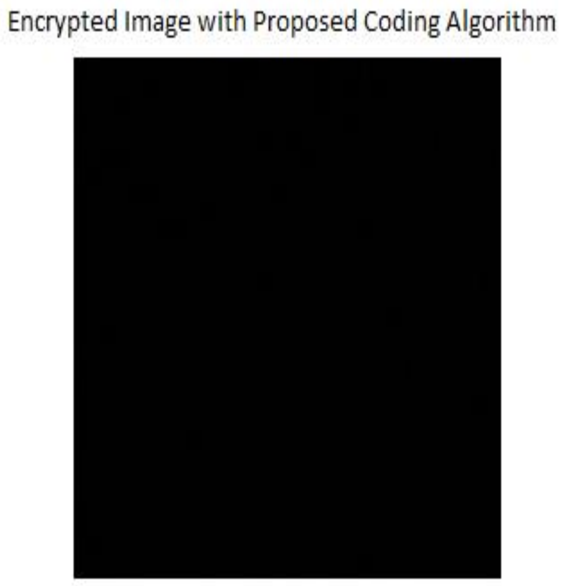


Figure 5: Encrypted image pixels using the newly developed secured image coding scheme

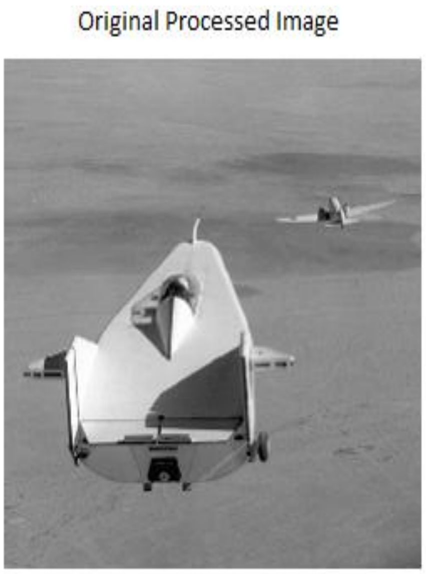


Figure 6: Encrypted image pixels using the newly developed secured image coding scheme

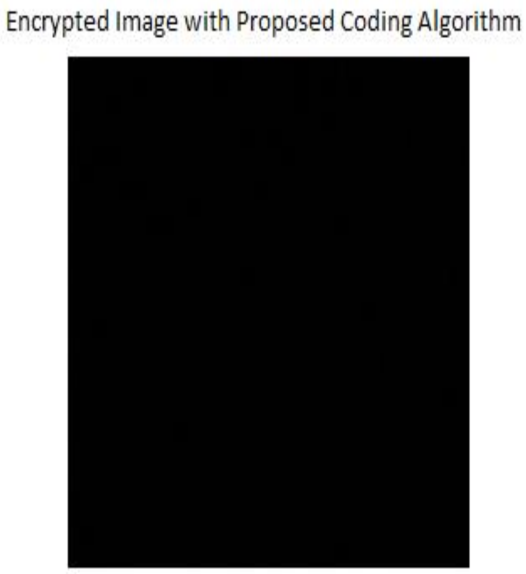


Figure 7: Encrypted image pixels using the newly developed secured image coding scheme

CONCLUSION

A secured image coding and compression scheme using an efficient coding algorithm is presented in this paper. The proposed coding algorithm can be used as the basis for full security and compression multiple access pictorial communication.

Further the proposed method is developed based on the top-bottom approach and it is simpler. As compared to the Huffman's data coding technique, the proposed scheme is more efficient in terms of compression and security. Thus, the proposed coding algorithm has a fast data transmission per image pixel through a transmission medium.

The proposed method also has a very simple decoder design to recover the image pixel. A lookup table is used in storing the image pixels and their respective characters and codes.

6. REFERENCES

[1] P. Pirch and H-J. Stolberg, "VLSI Implementations on Image and Video Multimedia Processing Systems", IEEE Transaction on Circuits and Systems for Video Technology, Vol. 8, No. 7, Nov., 1998.

[2] K. Konstantinides and V. Bhaskaram, "Monolithic architectures for image processing and compression", IEEE Computer Graphicsand Application, pp.75-86, Nov. 1996.

[3] Wei Wang, M.N.S. Swamy and M.O. Ahmad, "RNS Application for Digital Image Processing", 4th IEEE international workshop on System-on-chip for Real-time Application, pp. 77-80, July 2004.

[4] Geoff Dougherty, Digital Image Processing for Medical Applications, Cambridge University Press, 2007.

[5] C.Gonzalez and R.E. Wood, Digital Image Processing, Prentice-Hall, India, Second Edition, 2007.

[6] R.C.Gonzalez, R.E. Wood and Steven L. Edwin, Digital Image Processing Using MATLAB, Prentice-Hall, India, Second Edition, 2008.

[7] Willian K. Pratt, Digital Image Processing; Third Edition, Wiley-intersecience Publication, New York, U.S.A., Copyright 2001.

[8] Faller, N., "An Adaptive System for Data Compression", Record of the 7th Asilomar Conference on Circuits, Systems and Computers (Parcific Grove, Ca., Nov.), pp. 593-597, 1973.

[9] Cormack, G. V., "Data Compression on Database System", Commum. ACM 28, 12(Dec.), pp. 1336-1342, 1985.

[10] Cormack, G. V. 1985. Data Compression on Database System. Commum. ACM 28, 12(Dec.), pp. 1336-1342.

[11] Connell, J.B., "Huffman-Shannon-Fano Code. Proceedings", IEEE 61, pp. 1046-1047, July 1973.

[12] Fano, R. M., Transmission of Information. M.~I.~T. Press, Cambridge, Mass, 1949

[13] Y. C. Hu, C. C. Chang*, "A New Lossless Compression Scheme Based on Huffman

[14] Coding for Image Compression," *Signal Processing: Image Communication*, Vol. 16, No. 4, pp. 367-372, Nov. 2000.

[15] David A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes", Proc. IRE, 40, 1952.