Journal of Kerbala University , Vol. 6 No.3 Scientific. 2008

A Hybrid Compression Algorithm by Using Shannon-Fano Coding and Oring Bits

B.A.Al-hmmeary
م.م. بيادر عباس الحميري
جامعة بابل / كلية العلوم

Abstract:

Text compression plays an important role and it is an essential object to decrease storage size and increase the speed of data transmission through commmunication channels. This reseach aims to combine two compression methods in one data compression system. The first algorithm is Shannon-fano coding that is a stastical compression method for creating the code lengths of a integer-length prefix code, the second method is Oring Bits that deals with data streams of 0 and 1.

1.1 Introduction:

Data compression is the process of creating binary representations of data which require less storage space then the original data. There are two main data compression strategies: lossy and lossless.

Lossless data compression is used when the data has to be uncompressed exactly as it was before compression. Text files are stored using lossless techniques, since losing a single charcter can in the worst case make the text dangerously misleading. There are strict limits to the amount of compression that can be obtained with lossless compression. Lossless compression ratios are generally in the range of 2:1 to 8:1. [3]

Lossy compression in contrast, works on the assumption that the data doesn't have to be stored perfectly. Much information can be simply thrown away from images, video data, and audio data, and when uncompressed such data will still be of acceptable quality. Compression ratios can be an order of magnitude greater than those available from lossless mehods. Lossy compression ratios are in the range 20:1 to 30:1.[1]

There are many different ways that data compression algoritms can be categorized, such as:-

1- The algorithms can be classified as lossless like, RLE[7], Shannon[9], Huffman[10], LZW method[2] or can be classified as lossy like, JPEG[11], MPEG[11], CS&Q[10].

2- The algorithms can be classified according to a fixed or variable size of group taken from the original file and written to the compressed file. For example, the CS&Q algorithm is a fixed-input fixed-output scheme, the Shannon algorithm is a fixed-input variable-output scheme, the Arithmetic algorithm is a variable input and output scheme, and the RLE and LZW algorithms are variable-input fixed-output schemes[10].

2.1 Shannon-Fano Algorithm[4,8] :-

This is a common stastical method consists of a modeling stage followed by a coding stage. The model assigns probabilities to the input symbols, and the coding stage then actually codes the symbols based on those probabilities.

Starting with a set of *n* symbols with known probabilities(or frequencies) of occurrence, the Shannon-Fano algorithm works as the following:-

• Designed to define an effective Code table

• For a given list of symbols, develop a corresponding list of probabilities or frequency counts so that each symbol's relative frequency of occurrence is known.

• Sort the list of symbols according to frequency, with the most frequently occurring symbols at the top and the least common at the bottom

• Divide the list into two parts, with the total frequency counts of the upper half being as close to the total of the bottom half as possible.
• The upper half of the list is assigned the binary digit 0, and the lower half is assigned the digit 1. This means that the codes for the symbols in the first half will all start with 0, and the codes in the second half will all
start with 1.
• Recursively apply the same procedure to each of the two halves,
subdividing groups and adding bits to the codes until each symbol has
become a corresponding code leaf on the tree.

Table(1) illustrates the Shannon-Fano code for a seven-symbol alphabet (the symbols are not shown, only their probabilities).

Table (1) Shannon-Fano Example

	Probab.	Steps						Final code
1.	0.25	1	1					:11
2.	0.20	1	0					:10
3.	0.15	0		1	1			:011
4.	0.15	0		1	0			:010
5.	0.10	0		0		1		:001
6.	0.10	0		0		0	1	:0001
7.	0.05		0		0		0	0 :0000

The lengths of the codewords is equal to $\lceil \log_2(1/p_i) \rceil$, where p_i is the probability.

3.1 Oring Bits Algorithm [6,8]:-

This method starts with a sparse string L_1 of size n_1 bits. In the first step, L_1 is divided into k substrings of equal size. In each substring all bits are logically ORed, and the results (one bit per substring) become string L_2, which will be compressed in step 2. All zero substrings of L_1 are now deleted. In step 2, the same process is applied to L_2, and the result is the 4-bit string L_3.
And the method can continue by applying the above steps if there is another data.
An example for compressing the following sparse string L_1:-

L_1= 0000 | 0000 | 0000 | 0100 | 0000 | 0000 | 0000 |
1000 | 0000 | 0000 | 0000 | 0000 | 0010 | 0000 |
0000 | 0000

L_1 is a 64-bit string divided into 16 substrings of size 4 each. After Oring each 4-bit substring we get the 16-bit string L_2.

L_2= 0001 | 0001 | 0000 | 1000.

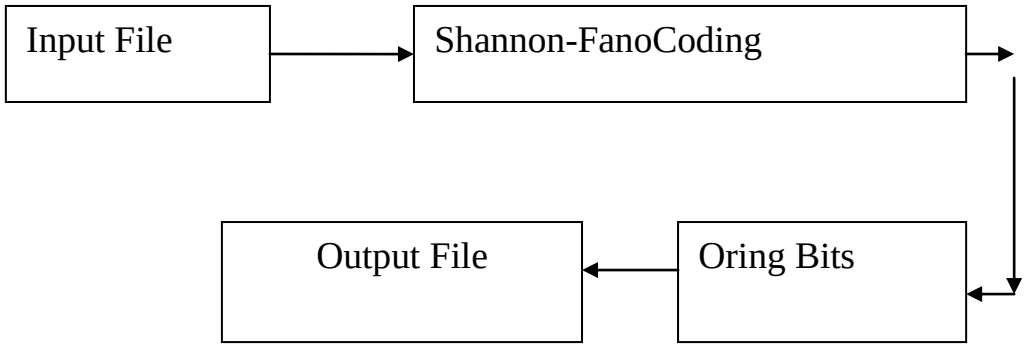
which is short enough so no more compression steps are needed. After deleting all zero substrings in L_1 and L_2 we end up with the three short strings

L_1 = 0100 | 1000 | 0010, L_2= 0001 | 0001 | 1000
L_3= 1101.

The output stream consists of seven 4-bit substrings instead of the original 16 substrings.
The decoder works differently (this is an asymmetric compression method). It starts with L_3 and considers each of its 1 bits a pointer to a substring of L_2 and each of its 0 bits a pointer to a substring of all zero that's not stored in L_2. This way string L_2 can be reconstructed from L_3, and string L_1, in turn, from L_2.

4.1 Proposed compression method:-

Figure (1) illustrates the block diagram of the proposed compression method.



Figure(1) Block Diagram of Proposed Method

From the above figure, we can see that the proposed compression method works as the following:-

1- Taking the input file, that is a random sequence of English alphabet symbols, computing the probability for each symbol.

2- Applying the Shannon-Fano coding of the sequence of probabilities, the result is a string of 0 and 1 bits.

3- Applying the Oring Bits method on the string of 0 and 1 bits(the result of Shannon-Fano method). Here the Oring Bits method divides the string using a 8-bit blocks.

4- The final compressed file contains:-
   a- the number of symbols.
   b- the symbols.
   c- the code words of each symbol.
   d- the final strings resulted from applying the Oring Bits method on the final code after substituting the codeword of each symbol in the input file which is divided into 8-bit blocks (inorder to apply the Oring Bits method) and each resulted 8-bit substring is stored in a byte.

4.2 Decompression of the proposed compression method:-

The number of symbols, each symbol, and the code words for each is distinguished.
The rest of the compressed file contains on the Oring Bits strings. After applying the decompression of this method, we get the final code that is read bit by bit in a sequential order and is compared with the code words to get the original text.

5.1 Results:-

In all the compression methods there are two main criteria should be taken into account, these are the compression ratio and the quality of results (the amount of error between the original data and the output data).

In this research the test sets of data files includes a sequence of English alphabet letters (random sequence).The compression ratio is calculated by

$$\left(100-\frac{\text{(File size (1))}}{\text{(File size (3))}}\right)*100\%$$

Where, file size (1) represents the original file size including the random sequence of English alphabet letters, while file size (3) represents the results after applying the proposed compression method.
The amount of errors is zero.
The following table illustrates the test sets of data which is used on the proposed compression method.

Table (2) Test sets of data

File	File size (1)	File size (2)	File size (3)	Compression ratio
File1	81 byte	38 Byte	28 byte	65.432%
File2	600 byte	295 Byte	235 byte	60.883%
File3	884 byte	200 Byte	198 byte	77.602%
File4	100 byte	48 Byte	43 byte	57.0 %
File5	427 byte	308 Byte	255 byte	40.281%
File6	292 byte	81 Byte	38 byte	72.260%
File7	600 byte	207 Byte	203 byte	66.167%
File8	97 byte	48 Byte	29 byte	70.103%
File9	204 byte	87 Byte	51 byte	75.00 %
File10	250 byte	78 byte	58 byte	76.80 %
File11	296 byte	120 byte	82 byte	72.297%
File12	436 byte	154 byte	125 byte	71.33 %

In the table, file size (2) represents the results after applying the Shannon algorithm on the data files .

6.1 Discussion & Conclusion:-

1- The proposed compression method is lossless here both of the Shannon-Fano and Oring Bits methods are lossless which is useful in text compression since losing a single character can in the worst case make the text dangerously misleading. This means increasing compression ratio without losing information.
2- According to the structure of Shannon-Fano coding, it produces a sparse string that makes it an effective method to combine with Oring Bits method. The distribution of 0 and 1 on the probabilities in Shannn-Fano method has a high effect on the Oring-Bits method. When the upper division is substituted by 1 and the lower division is substituted by 0, that is, the code word of the symbol with the highest probability is shorter than the code word of the symbol with the lower probability (i.e. a longer sequence of zeros). Then it will produce better results when combined with Oring Bits method.
3- When the compressed file contains on a longer sequence of frequented symbols, it has a high effect on the compression ratio as the Oring Bits gives better results.

References:

1- Armstrong A. & Jiang J., "**A Design of Genetic Vector Antisation Benchmarked by Competitive Learning Neural Network**", School of Computing University of Glamorgan, United Kingdom, 2000.
2- Blelloch G. E., "**Introduction to Data Compression**", Computer Science Department, Carnegie Melln University, Octobar 16, 2001.
3- Goebel G., "**Introduction/Lossless Data Compression**", 2005.
4- Lng. Andreas S.,**" Information Theory & Data Compression"**, International School of New Media at the University of Lubeck, 1999.
5- Muller R., "**Image Compression**" Part II, Image Processing Computer Graphics and Image Processing, Winter Semester 2003/04.
6- Nelson M. & Gailly J., "**The Data Compression Book**" (Second Edition), M&T Books, 1996.
7- Omer F. Acifel, "**Lossless Compression of Telemetry Data**", Technical Report, New Mexico State University, 1995.
8- Saloman D., "**Data Compression**", Department of Computer Science, California State University, USA, 1997.
9- Sergio V.,'**Fifty years of Shannon Theory"**,IEEE Transactions on Information Theory, Vol. 44, No. 6, Octaber 1998.
10- Smith S. W., A Sample Chapter from: "**The Scientist and Engineer's Guide to Digital Signal Processing**", 1997.
11- Smith W., "**Digital Signal Processing**", California Technical Publishing, 1999.