*International Journal of Engineering Technology and Scientific Innovation*

Volume:01,Issue:03

www.ijetsi.org

# SIMULATION AND ANALYSIS OF DATA COMPRESSION ALGORITHMS EMPLOYEES IN DEFECT DETECTION SYSTEM

Cortez Jose Italo[1], Cortez Liliana[2], Hurtado Madrid J.Miguel[1], Trinidad Garcia Gregorio[1], Garcia Juarez Pedro[1], Luisillo Hernandez Miguel[1]

Research Laboratory Digital Systems and Renewable Energies.
[1]Faculty of Computer Science, [2]Faculty of Electronics
Benemerita Universidad Autonoma de Puebla
Puebla, Mexico

## ABSTRACT

This paper presents the results of the implementation of three compression algorithms used on a wired digital communication system, an analysis was performed for each algorithm to determine the efficiency obtained for any one. This analysis will determine the type best suited to the needs of the transmission channel for optimal communication lossless compression.

**Keywords:** Compression, Digital Communication, RLE, Huffman and Shannon Fano.

## INTRODUCTION

Data compression involves taking a web of symbols and transform them into codes or keys, to represent the signal in a smaller space. In digital communications and computer systems it is employed to reduce the data volume. The space occupied by the uncompressed encoded information is the ratio of the sampling frequency and resolution, the more bits are used the greater the file size.

The resolution is imposed by the digital system and cannot alter the number of bits arbitrarily; therefore compression is used to transmit the same amount of information which occupy a high resolution in a smaller number of bits, the compression is a special case of encryption, whose main characteristic is that the resulting code is smaller than the original.

Data compression means reducing the number of characters transmitted likewise the probability of transmission error is reduced, increasing the system performance [1].

## METHODOLOGY IN THE DIGITAL SIGNAL COMPRESSION

There are 2 types of algorithms which data compression, lossy and lossless compression are classified.

A lossy compression is based on eliminating signal data to shrink, which is usually reduced quality, bit rate (bit rate) may have a constant or variable length [3].

After the compression, you cannot get the original signal, although an approximation whose resemblance to the original depends on the type of compression. These algorithms are mainly applied in compressing images, videos and sounds [4].

Compression algorithms lossless data are characterized by transformations use information from mathematical methods substitution by employing a dictionary or through the use of statistics to obtain the compressed information, in such compression is possible to recover the original signal by the reverse process of compression [5].

### A. Huffman compression algorithm

The lossless compression algorithm Huffman is characterized by the use of a dictionary that is built from a statistical analysis of the data to be compressed, this algorithm has a variable length, as the word size for each value is based on the probabilities thereof; for decompression algorithm of this type is necessary for the receiving system data dictionary containing the same that the transmission system, because without this you cannot retrieve the information received [6].

The design of the dictionary for this algorithm is based on the development of a tree diagram, where a branch represents a symbol and the probability of occurrence [9].

The code for each message builds following the path from the start point to the branch of the tree representing the message.

Furthermore, if the decoder implements the same tree used for compressing, decoding will be only read and interpret the way bits for encoding, as in the Fig. 1 [10].

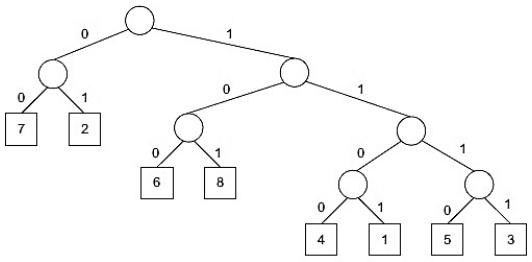


Fig. 1 Tree diagram for Huffman coding

Coding is inversely proportional to the probability of occurrence of the message, the symbol most likely to be assigned a shorter coding assigning less output alphabet symbols.

It considers that is necessary to select two nodes lower probability tree on each iteration, where a third node containing the summed probability symbols, relocating less likely nodes in the bottom diagram is obtained [11].

**B. RLE compression algorithm**

Compression Run Length Encoding (RLE) analyzes the data to determine consecutive repeated sequences of values and creates a transmission frame containing a single value more concurrent count the obtained data.

The first byte contains a number representing the number of times the character is repeated in the plot, the second byte contains the character. In other cases they are coded in a single byte: 1 bit (0 or 1) and 7 bits to specify the number of consecutive characters [7] [14].

**C. Compression algorithm Shannon - Fano**

The Shannon - Fano coding has a prefix code based on a set of symbols and their probabilities ensuring the length of the codeword to a bit of its theoretical ideal – log P(x) [8].

This algorithm generates an estimate based on the likelihood of each symbol contained in the compressed data required, coding each symbol with a binary variable length code.

In the Shannon-Fano coding, the symbols are sorted in descending order according to the probabilities of occurrence in a frame, then they regrouped into two subsets whose total probabilities are so close to being equal as possible [15].

The process of data analysis is described by the following algorithm.

- Frequencies of occurrence of symbols is calculated in a data frame.
- The list of symbols is ordered by frequency in descending order.
- The data are grouped into 2 subsets, ensuring that the probability of each subset approaching 50% of total probability.
- The first subset is assigned the binary digit 0 and the second subset is assigned the binary digit 1.
- This means that the code symbols in the first half begin with 0 and all codes in the second half start all 1.
- Each of the subsets is subdivided iteratively and bits (binary digits) are appended to the codes until each group consisting of a single symbol.
- The information entropy is calculated as:

$$X = \frac{Data\ Lenght}{Data\ Frequency} \qquad (1)$$

$$Entropy = Log\ 2\ (X) \qquad (2)$$

## DEVELOPMENT

Based on the references consulted and based on the most used for each compression type architectures, each algorithm design considering the specific requirements are made for each of the algorithms.

Statistical analysis necessary to develop dictionaries that were used in the Huffman and Shannon - Fano algorithms, RLE were made concerning an adaptation concerning the original algorithm was performed.

This modification consisted of the restructuring of the processing logic for the use of a finite arrangement, since the original algorithm requires a dynamic infinite fix, this presents a problem when the implementation because programmable logic devices used in this work are characterized by low memory and low consumption of resources, it is for this reason that the adaptation of this algorithm was necessary.

### A) Huffman compression

The Huffman compression algorithm was developed using compression architecture stability and use according to the references, taking this as a basis a flowchart (Fig. 2) is proposed based on this architecture.

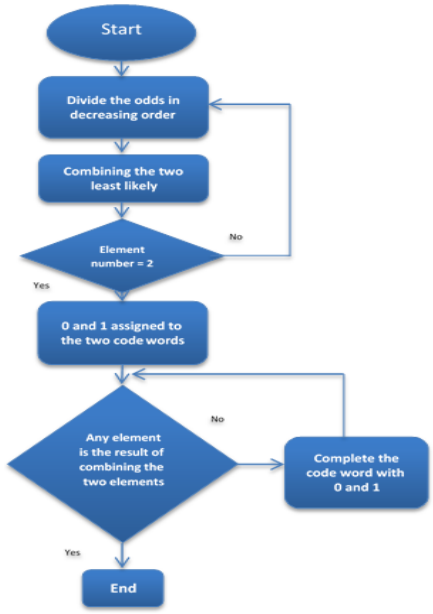


Fig. 2. Flowchart proposed in the implementation of Huffman compression algorithm.

### B) Run Length Encoding

The RLE compression algorithm was developed using compression architecture stability and use according to the literature review, this architecture is broken down more clearly in the following flowchart (Fig. 3).

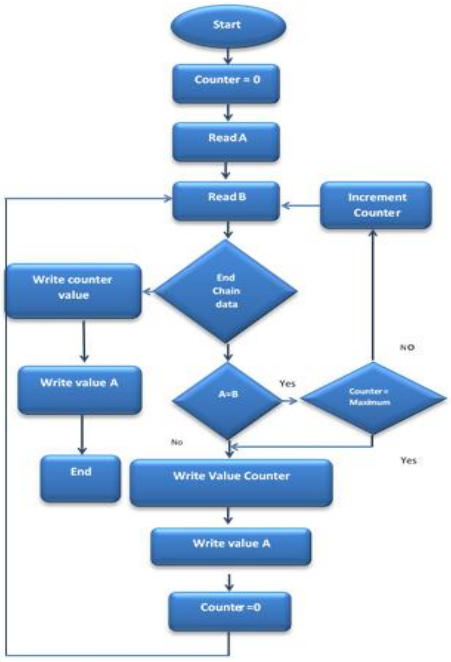


Fig. 3. Flow diagram corresponding to the architecture for the RLE algorithm.

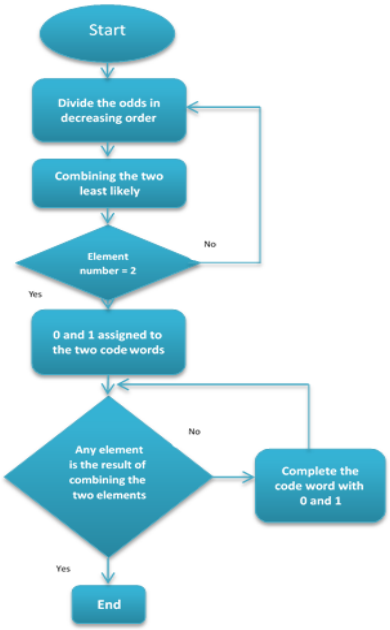


Fig. 4. Flowchart for implementing the Shannon - Fano algorithm

### C) Compression Shannon - Fano

The compressive modulus which is based on Shannon-Fano algorithm is designed based on the unique architecture reported for this type of algorithm, the flowchart (Fig. 4) was made from this architecture.

## RESULTS

The results of this work is broken down into three parts, first the results of the methodology used to design dictionaries for each algorithm is listed, then the measurements of the practical experiments are described.

Finally a statistical analysis between the results of simulations carried out and the results measured in practical experiments is pointed out, considering the parameters that measure the efficiency and optimizing each algorithm can determine the algorithm with higher performance.

Once applied the methodology and statistical analysis necessary information

that you want to compress the construction of dictionaries is performed, all symbols to design the dictionary, after the statistical analysis of the data was contemplated was found that the range of values It is determined by the closed interval [384 405], based on the result reported for this job only symbols used in compression algorithms for Huffman and Shannon - Fano.

A comparison between the uses of the channel in data transmission occurs, uncompressed data and after applying the RLE compression algorithm, numerically presents the efficiency of the compression algorithm in Table 1.

Table 1 Comparison of use of the transmission line a RLE.

| Data | Shannon-Fano | Data | Huffman |
|---|---|---|---|
| 384 | 111110 | 384 | 111011 |
| 385 | 11111100 | 385 | 1110101 |
| 388 | 0 | 388 | 0 |
| 389 | 10 | 389 | 10 |
| 391 | 11110 | 391 | 11100 |
| 392 | 1111111100 | 392 | 11101000 |
| 400 | 1110 | 400 | 1111 |
| 401 | 11111101 | 401 | 11101001 |
| 404 | 110 | 404 | 110 |
| 405 | 1111111101 | 405 | 1110101110 |

The results obtained by testing a sample of 1000 considering experimental data are presented in Fig. 5, which shows in red the number of symbols uncompressed, while the graph in blue represents the number of symbols archived using the algorithm RLE compression, the compression factor obtained by this algorithm, applying the criterion of compression rate is obtained that the compression ratio is obtained for this algorithm is highlighted 25.8 %.

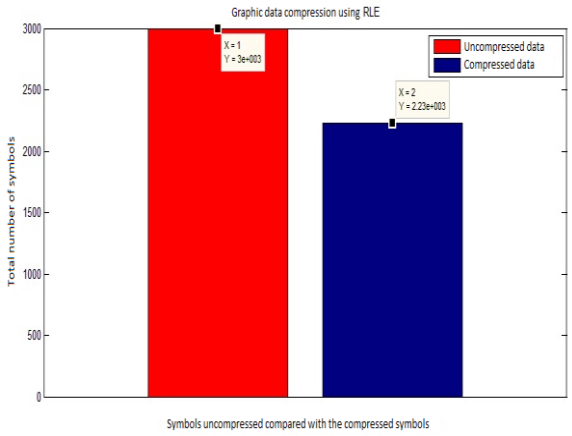


Fig. 5 Result of RLE data compression

Table 2. Comparison of use of the transmission line a RLE.

| | |
|---|---|
| Number of symbols uncompressed | 3000 |
| Number of symbols compressed | 2226 |
| Number of data | 1000 |
| Compression ratio data | 0.7420 |
| Percent Data Compression (%) | 25.8000 |

In the table 3 is considered a fragment of the transmission frame obtained by RLE.

Table 3. Hatching transmission obtained for the algorithm RLE.

| Encoded data |
|---|
| 1404138913881389238813891404138813912389138 81 |
| 401138823892388238913881389238813892388140014 |

The results obtained by the Huffman compression is shown in Fig. 6, the number of symbols without compression is displayed in red colored, while the number of symbols obtained by Huffman compression are shown in blue, concerning this section is highlighted the compression ratio obtained where the value is 30.6 %.

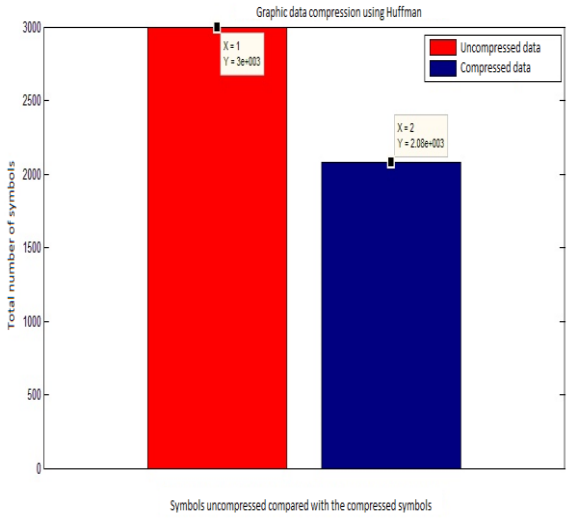


Fig. 6 Result data compression Huffman

Table 4 Comparison of use of the transmission line a Huffman compression.

| | |
|---|---|
| Number of symbols uncompressed | 3000 |
| Numberofsymbols compressed | 2082 |
| Number of data | 1000 |
| Compression ratio data | 0.6940 |
| Percent Data Compression (%) | 30.6000 |

Below it is part of the fabric of Huffman algorithm simulation:

| Encoded data |
|---|
| 1101001000101100111001010011101001010100010 1 |
| 0010001000111111010101100010001000000000101 00 |

Finally the results derived from Shannon - Fano algorithms is, applying the approach of compression rate, corresponding to a 30.3 % compression value was obtained as shown in Figure 7.

In Table 4 the numerical results are broken down, which highlights the compression ratio and the compression ratio, these values are close to the results obtained by the Huffman coding theorems which corroborates compression dictionaries based on where asserts that any dictionary-based algorithm will have a compression ratio bounded by an interval.

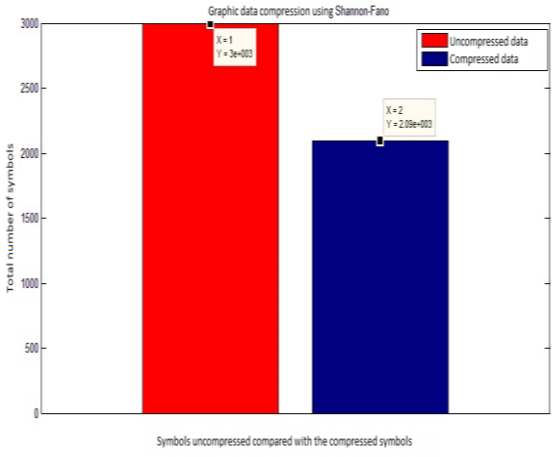


Fig. 7 Result data compression Shannon Fano

Table 5. Comparison of use of the transmission line compression Shannon Fano.

| | |
|---|---|
| Number of symbols uncompressed | 3000 |
| Number of symbols compressed | 2091 |
| Number of data | 1000 |
| Compression ratio data | 0.6970 |
| Percent Data Compression (%) | 30.3000 |

A part of the plot of the simulation algorithm is presented Shannon Fano:

| Encoded data |
|---|
| 1101001000101100111101010011111101010001 |
| 0100100010001110110101011000100010000000000 |

## CONCLUSIONS

In this work a data compression system in a software platform, which uses lossless compression algorithms, these are characterized by the use of dictionaries based on statistical data or information was developed to compress, this ensures these algorithms are optimized for this type of information.

The compression ratio as a first comparison parameter considered. In this section, the Huffman algorithm showed the best behavior with a compression ratio of 30.6 %.

Second comparison parameter as the compression rate required for each algorithm , in this comparison , the RLE algorithm presented the best performance with a speed greater than the remaining execution was taken algorithms, this algorithm because this does not require searching the dictionary for each value to compress.

As the third parameter to consider is the entropy, which provides information on using the canal. In this section, the RLE algorithm was presented the best performance, with a value of 4 very close to the ideal value for this information whose ideal value is 4.0576, which means that the algorithm has an optimum use of the channel for this type of information, parameters that were considered for this work, the best performing algorithm is the algorithm RLE.

## REFERENCES

[1] Denecker Koen and Van Overloop Jeroen, An Experimental Comparison of several Lossless Image Coders for Medical Images, 1068-0314/97, 1997 IEEE.

[2] Kenneth C. Adkiiis, Mary Jo Shalkhausert, Steven B. Bibyk, Digital Compression Algorithmsf For Hdtv Transmission, Dept. of Electrical Engineering The Ohio State University Columbus, Ohio 43210, CH2868 8/90/0000-1022$1.00 0 1990 IEEE.

[3] Gary Breed, Bit Error Rate: Fundamental Concepts and Measurement Issues, High Frequency Electronics, Summit Technical Media, LLC, 2003.

[4] Jose E. Briceño M., Transmision de Datos, Publicaciones de la Facultad de Ingenieria Escuela de Ingenieria Electrica, Taller de Publicaciones de la Facultad de Ingenieria, ULA, 2005.

[5] Compresion de fuente, notas de clase, Departamento de Ingenieria Telematica ETSET de Barcelona Universidad Politecnica de Cataluña.

[6] Luis Gabriel Rueda, Manuel Osear Ortega, Un Modelo de Codificacion Dinamica del Metodo de Huffman, para la Compresion de Datos en Lenea, 1er. Congreso Argentino de Ciencias de la Computacion, Domicilio: Ignacio de la Roza y Meglioli (5400) San Juan.

[7] David Salomon, Giovanni Motta, David Bryant, *Compresion de datos, La referencia completa*, Springer-Verlag London Limited, Cuarta Edicion, Primera en español, ISBN-10: 1-84628-602-6, 2007.

[8] MacKay D., Information *Theory, Inference, and Learning Algorithms*. Cambridge University Press. Version 7.0 (third printing) August 25, 2004.

[9] Viraktamath S.V., Attimarad G. V., Bhate Gaurav, *Impact of Selection of Source Coding Technique on the Efficiency* , International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN 2011).

[10] Anderson Jennings T., *the Mathematics of Cryptography & Data Compression,* Mathematics, Computer Science, & Engineering, Carroll College 2012.

[11] Amir Said William A. Pearlman, *Digital Signal Compression*. Cambridge University Press, United States of America, 2011.

[12] C. H. Messom, S. Demidenko2, K. Subramaniam and G. Sen Gupta, *Size/Position Identification in Real-Time Image Processing using Run Length Encoding,* EEE Instrumentation and Measurement, Technology Conference Anchorage, AK, USA, 21-23 May 2002.

[13] Kussay Nugamesh Mutter, Zubir Mat Jafri, Azlan Abdul Aziz, *Automatic Fingerprint Identification Using Gray Hopfield Neural Network Improved by Run-Length Encoding,* Fifth International Conference on Computer Graphics, Imaging and Visualization, 978-0-7695-3359-9/08 © 2008 IEEE.

[14] Jie Liang Chengjie Tu and Trac D. Tran. *Adaptive Runlength Coding*. The Johns Hopkins University Department of

Electrical and Computer Engineering, Baltimore, MD 21218.

[15] Xiaoyu Ruan and Rajendra Katti, *Using Improved Shannon-Fano Codes for Data Encryption,* Department of Electrical and Computer Engineering, Seattle, USA, July 9 14, 2006.

[16] N. Abramson, *Teorea de la Informacion y Codificacion,* Paraninfo, Quinta edicion, Madrid, 1981.