PERFORMANCE COMPARISON OF HUFFMAN CODING AND LEMPEL-ZIV-WELCH TEXT COMPRESSION ALGORITHMS WITH CHINESE REMAINDER THEOREM

Mohammed Babatunde IBRAHIM[1], Kazeem Alagbe GBOLAGADE[2]
[1]Department of Computer Science,Kwara State University, Malete, Nigeria
[1]imbamok@gmail.com, [2]kazeem.gbolagade@kwasu.edu.ng

Keywords: Text, Compression, Huffman coding, LZW, Chinese Remainder Theorem (CRT)

Abstract: The need for speed and storage space is the focus of recent computer technologies. The growth of an increased number of applications and important data is giving rise to new methods of having efficient compression giving rise to greater performance. This research paper provides two data compression algorithms and compares their performances based on some given text samples with the introduction of a new algorithm known as Chinese Remainder Theorem. The performance metrics used are compression size, compression time and compression ratio. The results indicate that Huffman-CRT on text outperforms LZW-CRT for compression size by 55.64% to 57.11%, compression time for Huffman-CRT is 44.61s better than LZW-CRT with 55.39s while Huffman-CRT also outperformed LZW-CRT by 3.70x to 2.37x for compression ratio.

1. INTRODUCTION

Digital data transmission via the internet and cellular networks has brought about an unprecedented increase over the past decade. Data compression is offering asmarterapproach to reducing communication cost through the use of available bandwidth effectively. Text, image, audio and video represent digital data. With this trend expected to continue, it makes sense to pursue research on developing algorithms that can be most effectively used on available network bandwidth by maximally compressing data [1].

The use of battery-powered storage devices for wireless devices and smartphones is drawing attention to the world of research for monitoring and communication. This attention has led to the need for text compression to be a prevailing and an important research area. Consequently, text compression is a key concern for management and compressing of data without changes observed in the text [2].

The main objective of a compression algorithm is to compress source text up to an optimal level that requires minimal spaceand consumes relatively less time and low overhead. In this paper, a new algorithm known as the Chinese Remainder Theorem (CRT) was introduced to enhance the performance of the conventional text compression techniques of Huffman Coding and Lempel Ziv Welch (LZW).

2. RELATED WORKS

[3] presented an approach effective for short text compression for smart devices. Their studyafforded a light-weight compression scheme where the storage required for compressing text is very low. Statistical context model for prediction of single symbols was used along with Arithmetic coding.

In [4] thenotion of their compression algorithm was to describe an encryption technique to minimize all words in the dictionary so as to reduce every word in the dictionary by substituting certain characters in the words via some special character, and retaining same characters so that the word is retrievable after compressing.

A novelapproach proposed by [5] wascreatedby applying signal processing to source text compression on filesnamely, the Fourier transform and the wavelet. The size of the compressed data, Fourier transform and wavelet thresholdis studied along with two

factors: wavelet filters and decomposition levels, on compression factor of text files, are investigated. It was indicated that from the results available, the Fourier transforms and wavelet are provided a base for lossy text compression with non-stationary text signal files.

In [6] authors presented an intelligent, reversible transformation technique that can be applied to the source text that improves algorithm ability to compress and also offer a sufficient level of security to the transmitted data. In this paper, the authors present an encoding technique known as Efficient Compression Algorithm (ECA) which offers the highest compression ratios. The authors suggest that in an ideal channel, the reduction of transmission time is directly proportional to the amount of compression. But in a typical Internet scenario with fluctuating bandwidth, congestion and protocols of packet switching, this does not hold true. The authors conclude that their results have shown excellent improvement in text data compression and added levels of security over the existing methods. These improvements come with additional processing required on the server/node.

In [7]the author discussed the lossless text data compression algorithms such as Run Length Encoding, Huffman Coding and Shanon Fano coding. The authors have concluded the article by doing a comparison of these techniques. The authors have also concluded that the Huffman technique is most optimal for lossless data compression.

2.1 HUFFMAN CODING

Huffman code is constructed by an optimal prefix code invented by Huffman. An algorithm is constructed in a bottom-up approach by building a tree T corresponding to the optimal code. It starts with a set of C leaves performing a C-1 merging procedure to build a final tree. It is assumed C is a set of n characters where each character is a defined frequency f[c] as an object. Q designated as a priority queue is keyed on f use to classify the two least frequency objects to merge together. The product of the merger of two objects is a new object whose frequency is the sum of the frequencies of the two objects that merged. O(n log n) is denoted as the total running time of Huffman on a set of n characters. Huffman algorithm is simple to implement and producing lossless images of compression as well as being optimal and compact code. Huffman code depends on the statistical model of data and is quite slow. The different code lengths make decoding a tedious process and also includes an overhead due to the Huffman tree [2].

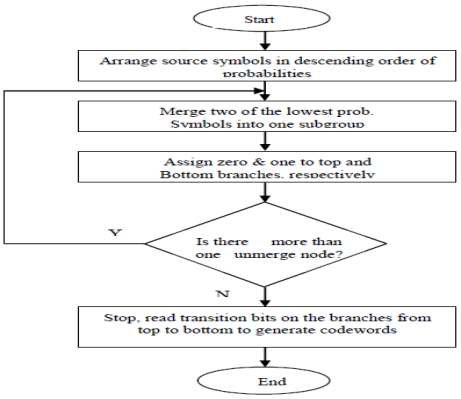

Fig. 1: Flowchart of Huffman coding [8]

2.2 LEMPEL ZIV WELCH (LZW)

The LZW algorithm is simple to implement thereby attaining a high throughput in hardware implementations. It is also a lossless data compression algorithm based on the dictionary model. Data is coded by referencing a dictionary as a replacement for tabularizing character counts and tree building as done by Huffman coding. Therefore, to encode a substring, a single code number corresponding to the substring's index in the dictionary is written to the output file. The LZW is mostly used for compressing text files although other file formats can be applied on it as well. Initially, the LZW dictionary comprises 256 entries (ASCII codes) of single characters. The pattern that is longest for each segment of the source text is identified which is then encoded by the indices in the current dictionary. This results in a new entry if no such match is found in the current dictionary. A match is found in the dictionary if the same segment is found in the future[2].

The LZW algorithm is presented below by [9]:

Step 1: At the start, the dictionary contains all possible roots, and P is empty;

Step 2: C: = next character in the char stream;

Step 3: Is the string P+C present in the dictionary?

(a) if it is, P:= P+C (extend P with C);
(b) if not,
–output the code word which denotes P to the code stream;
– add the string P+C to the dictionary;
–P: = C (P now contains only the character C);
(c) Are there more characters in the charstream?
–if yes, go back to step 2;
–if not:
Step 4: Output the code word which denotes P to the code stream;
Step 5: END.

2.3 CHINESE REMAINDER THEOREM

The basic operation of Chinese Remainder Theorem (CRT) is to generate a single integer through its residue modulo within moduli set (Xiao, Huang, Ye and Xiao, 2018). The CRT is popular referred to as theorem of number theory that states that when the remainder of Euclidean division of an integer *n* by many integers, then, it is possible to certain exclusively the remainder of the division of *n* by taking the product of these integers having satisfied the condition that the divisor are pairwise coprime (Zhang, Cui, Zhong, Chen and Liu, 2017). Given that $m_1, m_2, m_3, \ldots, m_n$ are the pairwise relatively prime positive numbers. And, the modular multiplicative inverse of an integer $P_i \mod p_i$ is expressed as P_i^{-1} and must conform to Equation 1 (Yan, Lu, Liu, Liu and Yang, 2018):

$$P_i . P_i^{-1} \equiv 1 (mod\ p_i) \qquad (1)$$

where, i = {1,2,3, …, n}. For any given n positive integers, $b_1, b_2, b_3, \ldots, b_n$, the CRT states that the pair of congruences are represented in Equation 2:

$$Y \equiv b_1 mod\ m_1, Y \equiv b_2 mod\ m_2, \ldots, Y \equiv b_n mod\ m_n \qquad (2)$$

The exclusive solution $mod\ \partial_g = m_1 m_2 \ldots m_i = \prod_{i=1}^{n} (m_i)$. The solution realised from the key server can be expressed by Equation 3.

$$X \equiv b_1 + b_2 + \cdots + b_n \left(mod\ \partial_g\right) = \sum_{i=1}^{n} b_i \beta_i \gamma_i \left(mod\ \partial_g\right) \qquad (3)$$

where, $\beta_i = \frac{\partial_g}{m_i}$, and $\beta_i \gamma_i \equiv 1\ mod m_i$.

3. MEASUREMENT OF PERFORMANCE METRICS

Performance metrics are used in determining which techniques is better off according to some criteria. The nature of application determines the metrics to be used for compression algorithm performances [13]. Time and space efficiency are some of the factors to be considered when performances are to be measured [14]. The study intends to use the following performance metrics in analyzing the results. MATLAB 2015a was used in designing the system for simulation.

- Compression Time (CT): This measures the rate of compressing data bits within a fraction of time (per second). It is measured in seconds.

- Compression Ratio (CR): This is measured as the ratio between uncompressed size (US) of text and compressed size (CS) of text in relation to its bits. It is denoted by

$$CR = \frac{US}{CS} \qquad (1)$$

The System screenshot is shown in figure 2 below:

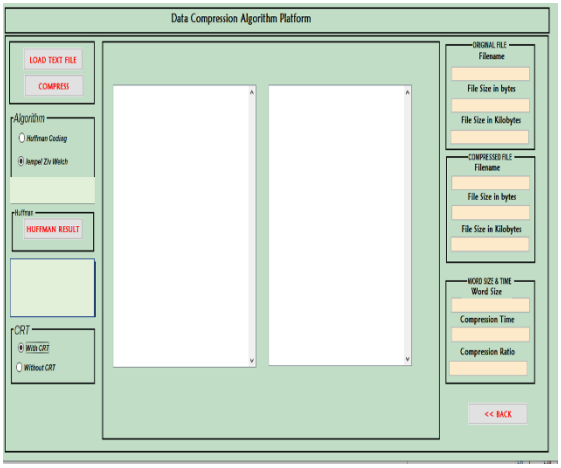

Fig.2: Data Compression Algorithm Platform System Graphical Interface

4. RESULTS

Since the compression behaviour depends on the redundancy of symbols in the source file, it is difficult to measure the performance of the compression algorithm in general. The performance of data compression depends on

the type of data and the structure of the input source. The compression behaviour depends on the category of the compression algorithm either lossy or lossless. The results are shown below:

Compression Size:

Table: 1.1: The sizes of texts for Huffman coding and LZW compression schemes

S/No.	Text filename	Original size (KB)	Huffman-CRT (KB)	LZW-CRT (KB)
1	nty.txt	1314	762	762
2	hufdes.txt	5452	682	880
3	license.txt	2628	1682	2032
4	output.txt	13456	1682	1844
5	paper44.txt	26572	15720	15016
6	result.txt	15720	15720	16668

From Table 1.1, the compression values of text files sizes for Huffman coding and LZW are same for text s/no. 1 only. Text s/no. 6 for Huffman-CRT retained the same size as the original size, while for the same text s/no., LZW-CRT was higher. Therefore, the compression values of Huffman-CRT compression algorithm slightly performed better when compared to LZW-based on CRT operation performed independently. After the compression, the resultant sizes of Huffman-CRT and LZW-CRT to the original text file are 55.64% and 57.11% respectively.

Compression Time

Table 1.2: Texts compression time for Huffman-CRT and LZW-CRT

S/No.	Text filename	Huffman-CRT (CT)	LZW-CRT (CT)
1	nty.txt	1.61	0.057
2	hufdes.txt	1.68	0.23
3	license.txt	2.46	0.20
4	output.txt	2.26	0.57
5	paper44.txt	16.85	22.46
6	result.txt	16.87	28.20

In Table 1.2, the compression time of text files for Huffman-CRT and LZW-CRT algorithm were significantly correlated in values. The LZW-CRT compression time was the best individually because it took the smallest time to perform compression operations for the selected texts except for Text S/No. 5 and 6. Upon successful compression operations based on CRT, the performances in terms of CT for Huffman-CRT and LZW-CRT algorithm on the original text files are 44.61s and 55.39s respectively. Therefore, the CRT-Huffman coding is best suitable due to the minimal time required for achieving compression of the text files sampled.

Compression Ratio

Table 1.3: Texts compression ratio for Huffman-CRT and LZW-CRT

S/No.	Text filename	Huffman-CRT (CR)	LZW-CRT (CR)
1	nty.txt	1.93	1.72
2	hufdes.txt	7.99	6.08
3	license.txt	1.56	1.29
4	output.txt	8.00	2.31
5	paper44.txt	1.69	1.77
6	result.txt	1.00	1.02

In Table 1.3, the compression ratio of text files for Huffman coding and LZW showed considerable improvements using CRT enhancement operations. In particular, LZW-CRT was better for text files compression ratio than Huffman-CRT accordingly. After successful compression operations based on CRT, the Huffman-CRT and LZW-CRT algorithms reduced their original data representations by 3.70x and 2.37x averagely in that order. Consequently, the Huffman-CRT outperformed LZW-CRT algorithms understudy in terms of capability to compress the original data representation by 3.70 times on the average.

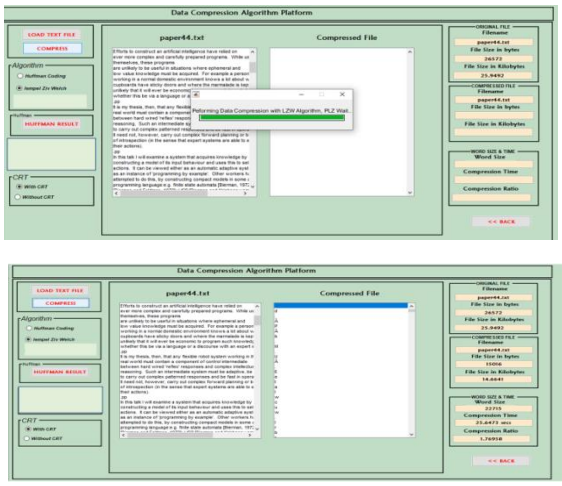

Fig 3a & b: Compressed Text file using CRT with LZW Algorithm

Table 1.4: Comparisons of Huffman-CRT and LZW-CRT based text compression

Evaluation Parameters	Huffman-CRT	LZW-CRT
Compression Size (CS) (KB)	55.64	57.11
Compression Time (CT) (s)	44.61	55.39
Compression Ratio (CR)	3.70	2.37

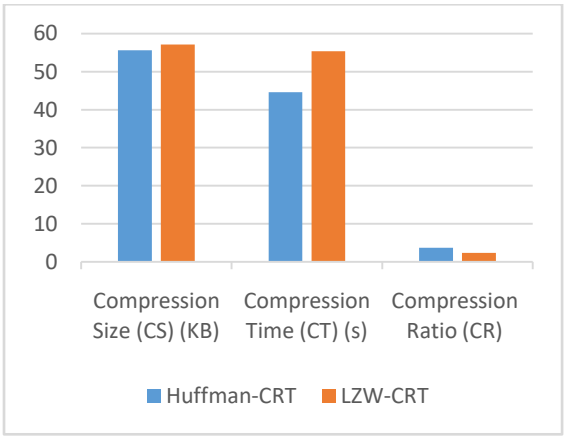

Fig. 4: Graphical view of text compression comparison.

Table 1.4 shows the summary of comparisons between both algorithms which is graphically represented in Figure 4.

5. CONCLUSION

Data compression is an important technique in the area of multimedia computing. This is due to the fact that reduction in the size, transmission and storage are faster and cheaper when compared to uncompressed data.This Study provided a performance comparison between two texts compression algorithm while incorporating Chinese Remainder Theorem (CRT) into them so has to see which performed better. For compression size, Huffman-CRT slightly performed better than LZW-CRT by 55.64Kb to 57.11Kb, for compression time Huffman-CRT also performed better than LZW-CRT by 44.61s to 55.39s likewise for compression ratio, Huffman-CRT also performed better than LZW-CRT by 3.70 times to 2.37 times. Therefore, the outcomes revealed the prospects of improving texts representations with CRT and other file formats. In future works, there is a need to deploy other compression algorithms such as Run Length Encoding, Discrete wave Transform, Adaptive Huffman coding and Arithmetic Encoding with CRT.

6. REFERENCES

[1]. Pratibha, W.,Agya, M.,*"Lossless Speech Compression Techniques: A Literature Review"*,International Journal of Innovative Research in Computer Science & Technology (IJIRCST), vol. 3, no. 3, pp. 25–32, 2015.

[2]. Yamini, A.,Harshit, G.,Parul, Y., Shikhar, N.,*"Literature Survey on Image and Text Compression Techniques"*,International Journal of Science Technology &Engineering (IJSTE), vol. 3, no. 9, pp. 626-630, 2017.

[3]. Islam, M. R., Ahson Rajon, S. A., *"An Enhanced Scheme for Lossless Compression of Short Text for Resource Constrained Devices"*,International Conference on Computer and Information Technology (ICCIT), 2011.

[4]. Franceschini, R., Mukherjee, A.,*"Data Compression using Encrypted Text"*, Digital Libraries, 1996. adl '96, Proceedings of the Third Forum on Research and Technology Advances.

[5]. Al-Dubaee, S. A., Ahmad, N.,*"New Strategy of Lossy Text Compression"*,Integrated Intelligent Computing (ICIIC), 2010.

[6]. Jain, A., Patel, R., *"An Efficient Compression Algorithm (ECA) for Text Data"*,International Conference on Signal Processing Systems, 2009 IEEE.

[7]. Rastogi, K.,Sengar, K.,*"Analysis and Performance Comparison of Lossless Compression Techniques for Text Data",* International Journal of Engineering Technology and Computer Research,vol. 2, no. 1, pp. 16-19, 2014.

[8]. Shahbahrami, A.,Bahrampour, R., Rostami, M. S., Mobarhan, M. A.,*"Evaluation of Huffman and Arithmetic Algorithms for Multimedia Compression Standards",*International Journal of Computer Science, Engineering and Applications (IJCSEA), vol. 1, no. 4, pp. 34-47, 2011.

[9]. Rubaiyat Hasan, Md., *"Data Compression using Huffman Based LZW Encoding Technique",*International Journal of Scientific &Engineering Research (IJSER), vol. 2, no. 11, pp. 1-7, 2011.

[10]. Xiao, H., Huang, Y., Ye, Y., Xiao, G.,"*Robustness in Chinese Remainder Theorem for multiple numbers and remainder coding",* IEEE Transaction on Signal Processing, 1-16, 2018.

[11]. Zhang, J., Cui, J., Zhong, H., Chen, Z., Liu, L., *"PA-CRT: Chinese Remainder Theorem Based Conditional Privacy-preserving Authentication Scheme in Vehicular Ad-hoc Networks"*, Journal of LATEX Class Files,vol. 14, no. 8, pp.1-14, 2017.

[12].Yan, X., Lu, Y., Liu, L., Liu, J., Yang, G.,*"Chinese Remainder Theorem-based Two-in-one Image Sharing with Three Decoding Option",*Digital Signal Processing,vol. 82, pp. 80-90, 2018. https://doi.org/10.1016/j.dsp.2018.07.015

[13]. Pooja, S.,*"Lossless Data Compression Techniques and Comparison between the Algorithms",International Research Journal of Engineering and Technology (IRJET),* vol. 2, no.2. pp. 383–386, 2015.

[14]. Cormak, V.,Horspool, S.,*"Data Compression Using Dynamic Markov Modeling*", Comput. J., 30, pp. 541–550, 1987.