ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

A Text File Compression Using Enhanced Move to Front Algorithm

D.Selvanayagi [#1] Dr.S.Pannirselvam[*2]

[#1] *Ph.D Research Scholar, Department of Computer Science, Erode Arts and Science College(Autonomous) Erode, Tamilnadu, India*
[*2] *Associative Professor & Head, Department of Computer Science, Erode Arts and Science College (Autonomous) Erode, Tamilnadu, India*

#1 selvasubhika@gmail.com
*2 pannirselvam08@gmail.com

Abstract- Data compression is defined as the art of science and representation of information in a logical condensed form. Different methodologies have been used to perform compression. The compressed data takes minimum bits to represent, store or transmit file content without reducing the original data. Data compression is divided into two categories such as lossless compression and lossy compression. Text data compression normally belongs to lossless. This research paper is proposed Enhanced Move To Front [E-MTF] algorithm using to implement lossless compression. The proposed E-MTF takes large volume of text paragraph as input. Input can be classified different words according to the character length wise. The proposed E-MTF algorithm extracts all available character from the whole paragraph. Then the proposed algorithm designs code book for compression. The algorithm of E-MTF performs two character words and repeats the same procedure until long length character which is given in the document. Finally, the proposed algorithm gives the better results of compression ratio while compared to existing algorithm. The compression and decompression algorithm is implemented through MATLAB 2013a.

Index words-**Text compression, Lossy, Lossless compression, E-MTF, Codebook, CR.**

I. INTRODUCTION

Image compression plays a major role in applications like internet browsing, medical science, navy applications, TV broadcasting and many more applications. The main goal of compression is to reduce the storage quantity as much as possible and the decoded image displayed in the monitor can be similar to the original data as much as can be. This process may be useful if one wants to save the storage space. Also compressed files are much more easily exchanged over the internet since they upload and download much faster. Compression has the ability to reconstitute the original file from the compression version at any time. Data compression is method of encoding rules that allows substantial reduction in the total number of bits to store the transmit the file. Data Compression[1] is the process of encoding data to fewer bits than the original representation. So, that it takes less storage space and less transmission time while communicating over a network.

The most important criteria of compressionis whether the compression algorithm removes some part of data which cannot be recovered during decompression. On the basis of this criterion, the data compression techniques are divided into two major categories, “lossy” data compression techniques and “lossless”[4] data compression techniques. Data compression is possible because most of the real world data is very redundancy. Data compression is basically defined as a technique that reduces the size of the data by applying different methods that can either be Lossy or Lossless.

Lossy Data Compression

A lossy data compression method is one where the data retrieves after decompression may not be exactly same as the original data. And it is contrasted with lossless data compression. Lossy data compression [10] does not produce original information after decompression. Compression is reducing the file size and eliminating some redundant data while applying lossy data compression.

Losslessdata Compression

Lossless data compression algorithms[8] mostly used for reducing the amount of source information can be transmitted, when information is decompressed. It is possible because most real-world data have statistical redundancy and these algorithms exploit these statistical redundancies to represent data more concisely without losing information.

MTF coding[5] is an inherit part in the locally adaptive data compression scheme. The MTF scheme simply "Move-To-Front" the last symbol seen such that its index in the tables of codes becomes zero.

This paper isorganised as follows. Section I, presents the basic introduction of compression and its types. In Section II, the researcher reveals the literature review of lossless compression. In Section III, the proposed methodology is discussed. Section IV, presents experimental results and its findings. In Section V, conclusion and future directions are discussed.

II. REVIEW OF LITERATURE

Peter et al [1] discussed the burrows-wheeler transform coding of the zero runs from the MTF recoding stage. The author described some new interpretations and uses of the MTF transform, with new insights and approaches to lossless compression, perhaps including techniques from error correction. The final section has described investigations into novel combinations of a burrows wheeler transform with PPM compression. Although the results are not as good as with standard BWT compression. The author suggested have been made for future work, one exploiting the ability to generate many contexts of rather greater length than that which they were generated, and other suggesting the inclusion of error correcting coding methods into compression.

Brenton et al[2] introduced the sorting stage of the Burrows-Wheeler transform, an aspect, can have a significant impact on the size of the compressed data. The author showed in the experimental result that smaller compressed output was achieved with two modifications to the sorting. One is using a better alphabet, and ordering and reflecting the sorted strings as in binary. The author considered alternative alphabet ordering based on both hand selected, and structured selected. Both main techniques add to the compression time, but alphabet reordering adds almost nothing to the decompression time. Even without such improved algorithm, the efficient alphabet is needed.

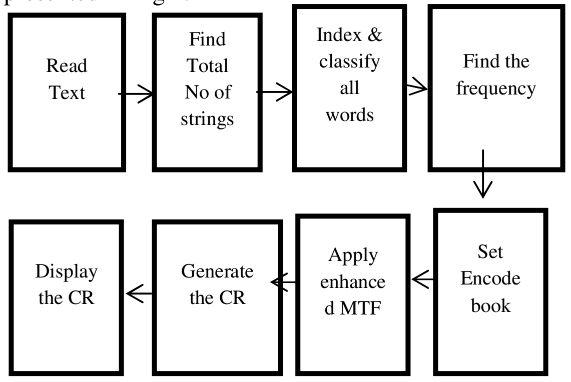
Sarada et al [3] developed many algorithms so far and still the process seems to be always increasing in search of a better compression scheme. The Burrows-Wheeler transform(BWT) has been a crucial tool for data compression. Normally the BWT earns maximum efficiency when the input is in text format. For this reason, this technique may be considered quite useful for data compression. The author proposed a modified BWT – based coding scheme that replaces a run length coding. Instead of transforming .mat to .text, the original format is used for any loss of data. Finally the author suggested that the MTF coder is more suitable as compared to other technique and the current compression ratio may further be improved.

N. Sangavan et al[13] introduced Laptops, computers, mobiles, everything is flooded with it.. Hence transferring, saving and securing it, is becoming a complicated and troublesome task. For considering transfer and saving of disk space, size needs to be reduced. Changing the data to meaningless form can make it immune to assailants. In this paper, an optimized approach has been proposed for dealing with both the issues of size and security. With this aim, we have combined Huffman compression with newly developed symmetric cryptographic algorithms and can be applied on text data. Paper consists of overall results of the methodology. The author suggested that new algorithm is better for text data.

III. PROPOSED METHODOLOGY FOR ENHANCED MOVE-TO-FRONT

That the input data are kept in their original format as are available in the source code. The choice of E-MTF[1] coding is important. While contexts are grouped together there is no statistical information kept, and so the encoder must rapidly adapt from the distribution of one context to the distribution of the next content. The E-MTF algorithm[13] has precisely this rapidly adapting quality. And this algorithm maintains list of all encoder and decoder. The current encoder is inversed into its position in the list and moved immediately to the original position. E-MTF list can be compressed into small integers.

Ideally an Enhanced MTF algorithm[7] should be changed whenever the context changes, but determining relevant context changes is hard enough for the compressor and even more difficult for the decompression. An algorithm that takes advantage of long runs of identical symbols is the MTF coding. The symbol [3] at the top of the list is assigned the number 0, the next is assigned the number 1 and so on. The first time a particular symbol occurs, the number corresponding to its place in the list is transmitted then it is mode to be top of the list. In this way, long runs of different symbols get transformed to a large number of 0's. Applying this technique is to proposed methodology. The proposed E-MTF method diagram is represented in Fig 1.



ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 20, April 2016

Fig 1. Compression encoded diagram for E-MTF

The Inverse method of decompression diagram is also presented as below in fig 2.

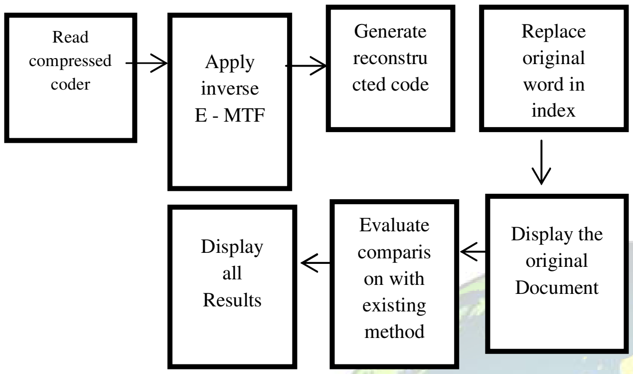

Read compressed coder
Apply inverse E - MTF
Generate reconstructed code
Replace original word in index
Display the original Document
Evaluate comparison with existing method
Display all Results

Fig 2. Decompression diagram for inverse E-MTF

From the above Fig 2 get the compressed coder from the document. Apply inverse MTF algorithms to generate reconstructed code to the replacement of original document.Finally evaluated the decompression code from the compressed document.

ALGORITHM

Encoder- Enhanced MTF Algorithm

```
Input: Text document from the file
Output: Encoded all the string
Step 1: Call Read textproc()
Step 2: Find the Total String
Step 3: Give Unique Index Number to string
Step 4: Call classify Proc()
Step 5: Calculate frequency item of repeated string
Step 6: Call code book proc()
Step 7: Apply Enhanced MTF procedure on input String
Step 8: Display compression table with CR
Step 9: Stop
```

Decoder- Enhanced Inverse MTF Algorithm

```
Input: Get compressed coded
Output: Generate Original document
Step 1: Read Compressed coded table
Step 2: Call inverse E- MTF_Proc()
Step 3: Generate final reconstructed Code
Step 4: Call Word Replacement_Proc()
Step 5: Display the original doc with compression ratio.
Step 6: Stop
```

IV. EXPERIMENTAL RESULTS

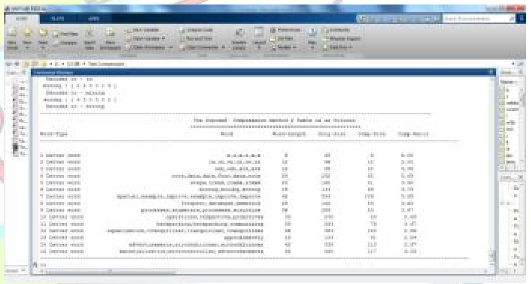
The E-MTF coder maintains a list of all symbol. The current symbol is recoded in to its position in the list and the node immediately to be head of the list, forcing all intervening symbols back by one position. The resulting working set of active symbols nearby head of the MTF list can be recoded into small integers, to be compactly encoded by the final statistical encoder. For example, it considers permuted sequence "ppkkkuaaaaaa". All symbols of a run, except the first, recode to 0 with MTF. The proposed algorithm reads the following text:

Text Compression Input is as below:

"The term data Compression refers to the process of reducing the amount of data required to represent a given quantity of information in this definition data and information are not the same thing data are the means by which information is conveyed because various amounts of data can be used to represent the same amount of information representations that contain irrelevant or repeated information are said to contain redundant data".

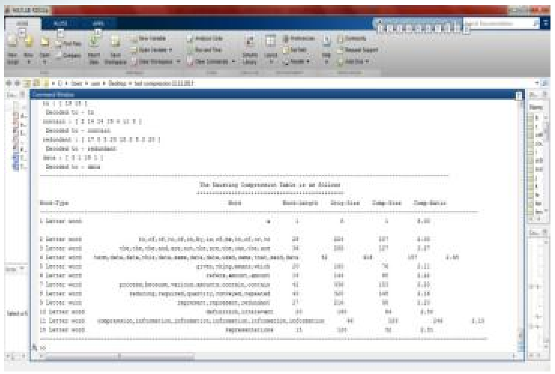
The proposed algorithm produces the following result for the above text input data. The existing method results follows in fig 3.

Fig.3 Compression table for Proposed1



```
The Proposed Compression method 1 Table is as follows
Word-Type        Word        Word-Length   Orig-Size   Comp-Size   Comp-Ratio
[illegible]
```

The corresponding proposed result by E-MTF is mentioned in table 2.



```
to : [ [illegible] ]
 Decoded to - to
contain : [ [illegible] ]
 Decoded to - contain
redundant : [ [illegible] ]
 Decoded to - redundant
data : [ [illegible] ]
 Decoded to - data

The Existing Compression Table is as follows
Word-Type        Word        Word-Length   Orig-Size   Comp-Size   Comp-Ratio
1 Letter word    a           1             8           1           8.00
[illegible]
```

Fig. 4 Compression table for Proposed E-MTF

The same procedure for compression and decompression is repeated in five document file. The results of compression ratio with existing and proposed method are indicated in the table 1.

Table 1. Comparison between Existing MTF and Proposed Method E-MTF

Doc Name	Original File Size(in bits)	Exist MTF Compression size(in bytes)	Proposed Enhanced MTF Compression size(in bytes)	Reduced Compression size(in bytes)	Compression Ratio
Doc 1	28919	25076	20678	308	1.39
Doc 2	38912	34671	25467	350	1.53
Doc 3	49561	44063	28563	404	1.73
Doc4	109202	98281	52092	602	2.10
Doc 5	147001	103652	56587	462	2.60

The proposed reads all Character from text file. The E-MTF improves the compression ratio whole compared to the existing method.

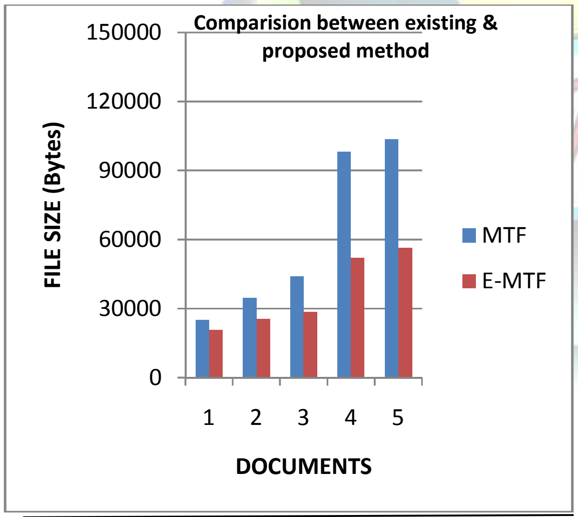


Fig 5 Comparison between Existing MTF and Proposed Method E-MTF

V. CONCLUSION& FUTURE WORK

From the above tested documents ensure that the higher rate redundancy helps to achieve more compression. In above presented enhanced MTF compression and Enhanced inverse MTF decompression is very efficient technique for lossless compression scheme. As for as concerning compression ratio, whenever text document size is increased, the compression ratio is also increased due to repeated character in a single string. It is observed that the proposed Enhanced MTF outperforms the improvements of the compression ratio in text document. The efficient proposed technique can be extend to gray, colour image subject to single pixel repeated more than one time with in block size.

REFERENCES

1. Peter Fenwick, "Burrows-Wheeler Compression: Principles and reflections" Theoretical Computer Science, Science Direct, 2007.
2. Brenton Chapin, Stephen R. Tate, department of computer science, denton, "Higher Compression from the Burrows-Wheeler transform by modified Sorting"
3. Sarada Prasad Dakura and JyotindersinghSahambi, "Lossless ECG compression for event recorder based on Burrows-Wheeler Transformation and Move-To-Front coder",International journal of Recent Trends in Engineering, Vol 1,No.3, may 2009.
4. ApoorvVikram Singh, Garima Singh, "A survey on Different text Data Compression Techniques", International Journal of Science and Research, vol 3, issue 7, July 2014.
5. Travis Gagie, Giovanni Manzini, "Move-To-Front, Distance Coding, and Inversion Frequencies Revisited", Theoretical Computer Science, Science Direct 2010.
6. ManjeetKaurEr. UpasnaGarg, "A Review of Various Data Compression Techniques to form a New Technique for Text Data Compression" Imperial Journal of Interdisciplinary Research (IJIR) Vol-1, Issue.5, 2015.
7. Adeshkumar, PrakshiRastogi, PragyanSrivastava, "Design and FPGA Implementation of DWT, Image Text Extraction Technique, Science Direct 2015.
8. K.Mahendrababu, Dr.P. Sathyanarayan, "A comparative Analysis of image Compression using various Compression Methods", international Journal of advanced Research in Electrical, Electronics and Instrumentation Engineering. Jan 2014.
9. R. R. Baruah, V.Dekal, M.P.Bhuyan,"Enhancing Dictionary Based Pre-processing For Better Text Compression", International journl of Computer Trends and Technology, vol 9, mar 2014.
10. Enas Abu Jrai, "Efficiency lossless Data Techniques for Araabic Text Compression", international journal of Computer Science, Information Technology. Vol 6, no 5, oct 2014.
11. HaroonAlterawneh, /mohammadAltarawneh,"Data Compression Techniques on text Files: A comparison Study", International Journal of Computer Applications, Volume 26, july 2011.
12. N. Sangwan"Combining Huffman text compression

with new double encryption algorithm", Emerging trends in communications, control, signal processing & computing 2013 Oct.

13. Manjeet Kaur Er. Upasna Garg "A Review of Various Data Compression Techniques to form a New Technique for Text Data Compression", Imperial Journal of Interdisciplinary Research (IJIR) Vol-1, Issue.5, 2015.