International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 4 Issue 04, April-2015

Survey of Dictionary Based Compression

Aarti Parekh
Department of Information Technology
Shri S'ad Vidya Mandal Institute of Technology
Bharuch 392-001, Gujarat, India

Mrunali Solanki
Department of Information Technology
Shri S'ad Vidya Mandal Institute of Technology
Bharuch 392-001, Gujarat, India

***Abstract*—Dictionary Based Compression is a useful technique through which we can encode variable-length strings of symbols as single tokens. There are number of algorithms available for Dictionary Based Compression. It uses less computing resources so it is very effective compression technique. The purpose of this paper is to present and analyze a variety of dictionary based algorithms.**

Keywords—Compression, Dictionary encoding, Text Compression , Lossy , Lossless

I. INTRODUCTION

Compression is representing information in a compact form rather than its original form. With increasing amount of data being stored, sufficient information retrieval and storage in the compressed area has become a major concern [4]. Compression is the process that will reduce the total number of bits needed to represent some information. There are lots of data compression algorithms which are available to compress files of different formats.

The aim of data compression is to reduce redundancy in stored or communicated data, so we can increase effective data density. Dictionary based encoding process is known as substitution encoding. In this process the encoder maintain a data structure known as 'Dictionary'[3]. The encoder matches the substrings chosen from the original text and finds it in the dictionary; if a successful match is found then the substring is replaced by a reference to the dictionary in the encoded file.

II. TYPES OF COMPRESSION

In data compression classification, There are two types:

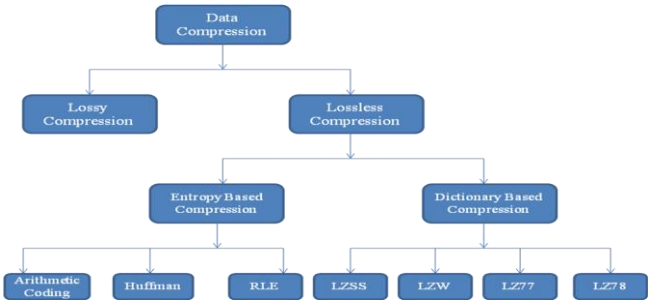

Fig 1: Types of Data Compression Techniques

A. *Lossy Compression*

As the name of Lossy Compression, There may be some loss of data in order to achieve higher compression. It is fundamentally different from lossless Compression [7]. It is generally done on analog data stored digitally, with the primary application being graphics and sound files. In some cases lossy method can produce much smaller compressed file.

B. *Lossless Compression*

Lossless data compression is used to compact files or data into a smaller form. No loss of data in order to achieve high compression. Lossless data compression has the constraint that when data is uncompressed, it must be identical to the original data that was compressed. Graphics, audio, and video compression such as JPEG, MP3, and MPEG on the other hand use lossy compression schemes which throw away some of the original data to compress the files even further.

III. DICTIONARY BASED TECHNIQUES

They are used to a type of adaptive dictionary when performing acronym replacements in technical literature. The standard way to use this adaptive dictionary is to spell out the acronym, then put its abbreviated substitution in parentheses. From then in the text should automatically invoke a mental substitution.

A. *Lempel Ziv Algorithms*

Generally compression schemes used statistical modeling. But in 1977 and 1978, Jacob Ziv and Abraham Lempel described a pair of compression methods using an adaptive dictionary. These two algorithms introduce new techniques that used dictionary-based methods to achieve new compression ratios.

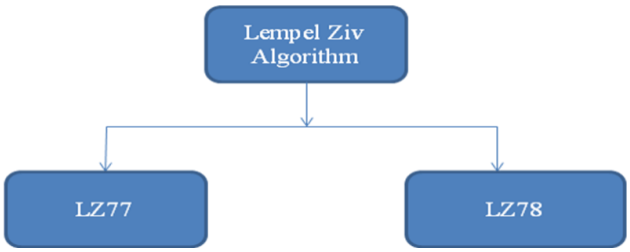

Fig 2: Family of Lempel Ziv Algorithm

1. LZ77 :

The first compression algorithm in 1997 produced by Ziv and Lempel is commonly referred to as LZ77. It is dictionary based & lossless data compression algorithm. No prior knowledge is required to solve [5].The input sequence is carried out by the encoder, putting into a sliding window which consist two parts: A search buffer which consist recently encoded sequence and a Look ahead buffer which consist the next portion of sequence to be encoded. Then longest match will found and put it into a search buffer. The output is in the form of a triple <offset, length, next symbol>.If no match found then null pointer is generated[1].

The length of offset to match and length must be limited to some extent. Usually the offset is encoded on 12-16 bits so it is limited from 0 to 65535 symbols. The match length is encoded on 8 bit. The algorithm for LZ77 is given below:

```
while (lookAheadBuffer not empty)
 {
get a reference (position, length) to longest match;
if (length > 0) {
output (position, length, next symbol);
shift the window length+1 positions along;
}
 Else
 {
output (0, 0, first symbol in the lookahead buffer);
shift the window 1 character along;
}
 }
```

In LZ77 , Most of compression time is used in searching for longest match whereas the decompression is quick as each reference is replaced with the string which it points to.

2. LZ78:

In 1978, Jacob Ziv and Abraham Lempel introduce new dictionary based scheme which is known as LZ78.This compression algorithm maintains explicit dictionary. Both the side dictionary has to be built for encoding and decoding and they must follow the same rule to ensure that they used an identical dictionary. The output by the algorithm consist of two elements:<i, c> where 'i' is an index referring to the longest matching dictionary entry and first non matching symbol. The algorithm for LZ78 is given below:

```
w := NIL;
while (there is input){
K := next symbol from input;
if (wK exists in the dictionary) {
w := wK;
} else {
output (index(w), K);
add wK to the dictionary;
w := NIL;
}
}
```

The Encoding Done By LZ78 Is Fast Compare To LZ77. The Important Property Of LZ77 That LZ78 Preserves Is Decoding Is Faster Than Encoding .The Decompression Is Faster Compared To The Process Of Compression.

IV. COMPARATIVE ANALYSIS

We focus on to compare the performance of LZ77 and LZ78.To find the efficiency of any compression algorithm is achieved by two important parameters like how much amount of compression achieved and time used by encoding and decoding algorithms. We have testing some practical performance on above mentioned two techniques LZ77 and LZ78.Table 1 shows comparison between LZ77 and LZ78 based on BPC measurement.

Compression Ratio

Compression ratio is the ratio between the original size of the file and the compressed size of the file it is calculated as

$$\text{Compression Ratio} = \frac{\text{Original Size}}{\text{Compressed Size}}$$

Table 1.Comaprison of BPC for LZ77 and LZ78 algorithms

SR NO.	FILE NAME	FILE SIZE	LZ77 BPC	LZ78 BPC
1	Mybook1	768771	4.57	3.92
2	Mybook2	610856	3.93	3.81
3	Mybook3	246814	3.81	4.68
4	paper1	39611	3.84	4.6
5	Paper2	49379	2.93	3.84
6	Paper3	93695	2.98	3.92
7	Object1	21504	5.41	5.58
8	Prog1	111261	3.75	3.95
9	Prog2	377109	4.37	4.33
10	Prog3	82199	4.1	4.24
Average			3.969	4.287

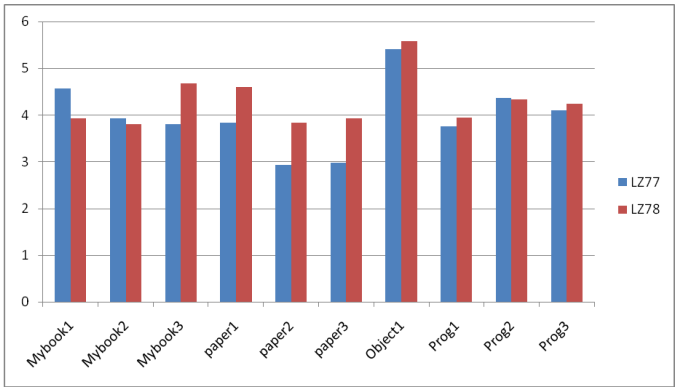

Fig 3: Comparison of compression ratio of LZ77 & LZ78 algorithm

V. CONCLUSION

Dictionary based Compression is an important field of research due to its wide range of this paper we performed a survey on various lossless dictionary based compressing. Techniques. Paper focus mainly on algorithm LZ77 and LZ78. Comparative analysis is provided for the discussed techniques based on the compression ratio achieved by each technique.

REFERENCES

[1] Amit Jain, Kamaljit I. Lakhtaria, "comparative study of dictionary based compression algorithms on text data,", International Journal of Computer Engineering and Applications, Volume VI, Issue II, May 14.

[2] Rupinder Singh,Brar Bikramjeet singh, " A Survey on Different Compression Techniques and Bit Reduction Algorithm for Compression of Text/Lossless Data", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 3, March 2013

[3] *Burrows M., and Wheeler, D. J. 1994. A Block-Sorting Lossless Data Compression Algorithm. SRC Research Report 124, Digital Systems Research Center.*

[4] Mark Nelson, Jean-Loup Gailly, "The Data Compression book" 2nd Edition

[5] Ziv. J and Lempel A., "A Universal Algorithm for Sequential Data Compression", IEEE *Transactions on Information Theory 23 (3), pp. 337–342, May 1977*

[6] Przemyslaw Skibinski, "Reversible Data transforms that improve effectiveness of universal lossless data compression", Ph.D thesis, Department of Mathematics and Computer Science, University of Wroclaw, 2006

[7] *Mohammad Banikazemi, "LZB: Data Compression with Bounded References", Proceedings of the 2009 Data Compression Conference, IEEE Computer Society, 2009.*

[8] *Fiala E.R., and D.H. Greene, "Data Compression with finite windows", Communications of the ACM 32(4):490-505, 1989.*

[9] Arup Kumar Bhattacharjee, Tanumon Bej, Saheb Agarwal "Comparison Study of Lossless Data Compression Algorithms for Text Data" IOSR Journal of Computer Engineering (IOSR-JCE).

.