© 2017 IJSRST | Volume 3 | Issue 2 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X

National Conference on Advances in Engineering and Applied Science (NCAEAS)
16th February 2017
**In association with**
**International Journal of Scientific Research in Science and Technology**

# Comparative Study of Data Compression Techniques

**Supriya Gabhane[1], Songa Bhattacharyya[2], Snehalata Shende[3]**

[1,2,3]Department of Electronics & Telecommunication, S. B. Jain Institute of Technology Management & Research, RTMNU, Nagpur, Maharashtra, India

## ABSTRACT

Source Coding is used to reduce the redundancy for transmission of data. This leads to need for 'data compression' that is, which requires less storage with high transfer rate .Thus the cost of storage hardware and bandwidth decreases. The paper deals with the number of data compression techniques named Huffman Coding, LZ coding and Shannon-Fano Coding and these coding are used for lossless data compression. Lossless data compression techniques regenerate the original data from the compressed file. The paper provides a detail survey and comparison of the different lossless data compression techniques.

**Keywords :** Lossless Data Compression, Huffman Coding, L-Z W, Shannon- Fano Coding.

## I. INTRODUCTION

In today's era of technological advancement, there is demand for multimedia that has made the use of wireless data occur frequently (i.e. redundancy). The transmission of such data requires a lot of bandwidth and for its reliable communication. The limited accessibility of these requirements is constraints in the reliable design and efficient communication system. Thus, the technique of data compression is used. This technique is used for transmission of the required information and to reduce the redundancy of the data. The conversion of source (i.e. analog signal) into digital or binary form (i.e. binary code) is called as source coding. This technique also reduces the size of the data & transmits it more efficiently whereas in channel coding the size of the data is increased by adding redundant bits. The source encoder does the process of conversion from analog to digital. The source encoder will get the input from the original source (which is generally analog signal) and the output of the source encoder is in digital form. Thus, source encoder does data compression. Data compression is also known as source coding or bit-rate reduction. There are two types of data compression techniques namely Lossy and Lossless.

## II. METHODS AND MATERIAL

### 2. Compression Techniques

In compression, the original data is recovered back when we restore it. There are two types of data compression techniques. They are as follows:

- Lossy Compression.
- Lossless Compression.

### 2.1 Lossy Compression

It describes coding algorithms that are characterized by an irreversible loss of information. Only an approximation of the original source data can be reconstructed. Lossy coding is the primary coding type for the compression of speech, audio, picture, and video signals, where an exact reconstruction of the source data is not required. In this technique, the data is not restored back to 100% of the original signal. In lossy compression, the data is permanently lost especially the redundant data but the user may not notice it. For example, the Joint Photographic Experts Group (JPEG), which is an image format, uses this type of compression technique.

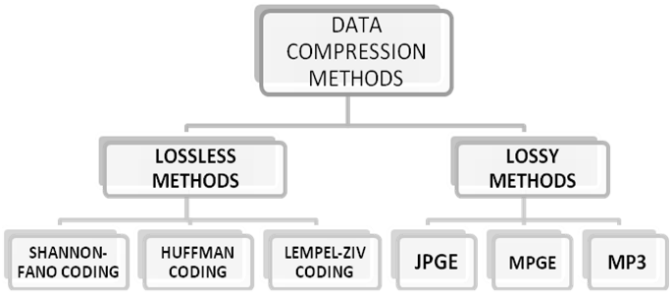



**Figure 1.** Block diagram of Data Compression

### 2.1 Lossless Compression

It describes coding algorithms that allow the exact reconstruction of the original source data from the compressed data. Lossless coding can provide a reduction in bit rate compared to the original data, when the original signal contains dependencies or statistical properties that can be exploited for data compaction. It is also known as noiseless coding or entropy coding. Lossless encoding methods guarantee to reproduce exactly the same data, as was input to them. In lossless compression every single bit is restored back to get the same data which was transmitted at the transmitter side. For example, the Graphics Interchange File (GIF), which is an image format, uses this type of compression technique.

## 3. Lossless Compression Techniques

There are three types of lossless compression techniques they are as explained below:

### 3.1 Shannon Fano Coding

Shannon–Fano coding is named after Claude Shannon and Robert Fano, is a technique for constructing a prefix code based on a set of symbols and their probabilities (estimated or measured). It was invented in 1949. It does guarantee that all code word lengths are within one bit of their theoretical ideal. This is one of the three earliest technique for data compression that was invented by Claude Shannon & Robert Fano in 1949.In this technique a binary tree is generated that represents probabilities each code symbol occurring. The symbols are encoded in a way such that the most frequent symbol appears at the top of the tree and the least likely symbols appear at the bottom. An efficient code can be obtained by the following procedure **Algorithm:**

**Step 1**: For given list of symbols, develop a corresponding list of probabilities or frequency counts so that each symbol's relative frequency of occurrence is known.

**Step 2:** Sort the list of symbols according to the frequency, with the total frequency with the most frequently occurring symbols at the left and the least common at the right.

**Step 3:** Divide the list into two parts with the total frequency counts of the left part being as close to the total of the right as possible.

**Step 4**: The left part of the list is assigned the binary digit 0, and the right part is assigned a digit 1.This means that the codes for the symbols in the first part will all start with 0, and the codes in the second part will all start with 1.

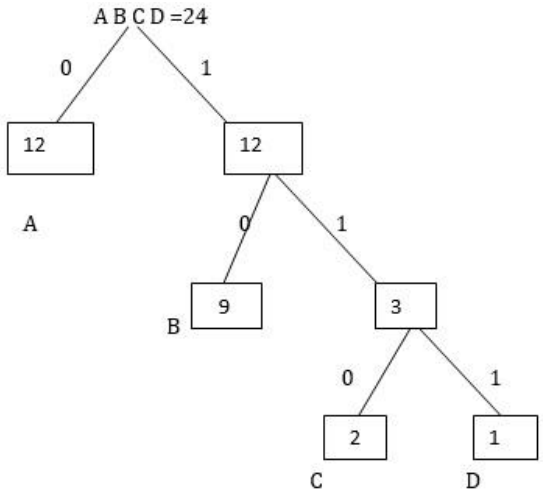
**Step 5:** Recursively apply the step 3 & 4 to each of the two halves, sub diving groups and adding bits to the codes until each symbol has become a corresponding code leaf on the tree.

For Example, Let us consider a data stream as given below:

Source Stream:

ABBBAAAABCABAACBBABBDAAA

| Source | A | B | C | D |
|---|---|---|---|---|
| Frequency | 12 | 9 | 2 | 1 |




| Source | Code |
|---|---|
| A | 0 |
| B | 10 |
| C | 110 |
| D | 111 |

No. bits required for uncompressed data: 24 * 8 = 192 bytes
No. bits required for compressed data :9 * 8 = 72 bytes

### 3.2 HUFFMAN CODING TECHNIQUE

Huffman Coding was named after its inventor, David Huffman in year 1950 while he was student in MIT. In general, Huffman coding results in optimum code. Thus, the code has the highest efficiency.

The symbol occurring more number of times will have least code as compared to that of the symbol, which does not, occur more number of times. The symbol that occurs same no of times will have same length. In general, Huffman coding results in an optimum code. Thus, the code has the highest efficiency. The Huffman encoding procedure is as follows:

**Step 1:** Put all the nodes sorted order of their frequencies/ probabilities.
**Step 2:** Repeat the following steps until we are left with a single node.

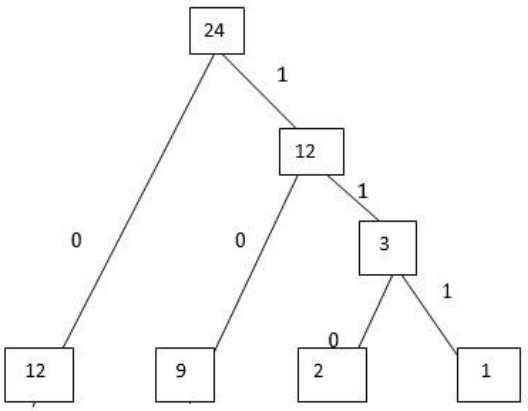
(i) From the sorted list pick two nodes having least frequencies/ probabilities and create a parent node of them.
(i) Assign the sum of children's frequencies/ probabilities to the parent node and insert it into the list in such a way that its order is maintained.
(ii) Delete the children node from the list.

**Step 3:** Assign code 0, 1 to the two branches of the tree on the path from the root.

For Example, Let us consider a data stream as given below:

Source Stream:
ABBBAAAABCABAACBBABBDAAA

| Source | A | B | C | D |
|---|---|---|---|---|
| Frequency | 12 | 9 | 2 | 1 |




A B C D
Code: 0 10 110 111

No. bits required for uncompressed data: 24 * 8 =192 bytes
No. bits required for compressed data : 9 * 8 = 72 bytes

### 3.3 Lempel-Ziv Coding Technique

Terry Welch created the Lempel- Ziv-Welch (LZW) algorithm in 1984. It removes the characters, which occurs more no of times in the output (i.e. redundant character) and includes every character before starting compression and employees other techniques to improve compression.

**Lempel Ziv Coding Algorithm**

**Step 1:** Initialize the dictionary to contain all blocks of length one (D={a ,b}).
**Step 2:** Search for the longest block **W** that has appeared in the dictionary.
**Step 3**: Encode **W** by its index in the dictionary. **Step 4**: Add **W** followed by the first symbol of the next block to the dictionary.
**Step 5** : Go to Step 2.

For Example, Let us consider a data stream as given below:

Source Stream:

ABBBAAAABCABAACBBABBDAAA

| Numeric Position | Sub-Sequence | Codes |
|---|---|---|
| 1 | A | A |
| 2 | B | B |
| 3 | BB | 2B |
| 4 | AA | 1A |
| 5 | AAB | 4B |
| 6 | C | C |
| 7 | AB | 1B |
| 8 | AAC | 4C |
| 9 | BBA | 3A |
| 10 | BBD | 3D |
| 11 | AAA | 4A |

No. bits required for uncompressed data: 24 * 8 = 192 bytes
No. bits required for compressed data :19 * 8 = 152 bytes

## III. CONCLUSION

In this paper we compared three techniques of data compression and we concluded that Shannon-Fano technique and Huffman Coding Technique is more efficient than Lempel-Ziv Coding Technique. In the example given in the paper Lempel – Ziv Technique requires maximum number of bits for transmission than that of Huffman Coding Technique and Shannon – Fano Coding Technique and gives better results.

## IV. REFERENCES

[1]. Dr.Sanjay Sharma, "Information Theory in the Communication Systems(Analog and Digital) book" published by S.K. Kataria & Sons ,New Delhi.
[2]. K.A Ramya and M. Pushpa. 2016. International Journal of Scientific Engineering and applied science (IJSEAS) – Volume-2, Issue-1,January 2016.ISSN: 2395-3470.