

# Applying Huffman Coding to Images and Data for Compression

K. Surekha<sup>1</sup> Mr. S. Muni Kumar<sup>2</sup>

<sup>1</sup>Student <sup>2</sup>Assistant Professor

<sup>1,2</sup>Department of Computer Applications

<sup>1,2</sup>KMM Institute of PG Studies, Tirupati, India

**Abstract**— Huffman coding is a method of data compression that is independent of the data type that is the data could represent an images and data. This compression scheme is used in JPEG. Huffman coding works by looking at the data stream that makes up the file to be compresses. The image compression is a technology in the field of digital image processing. Images are very important documents nowadays; to work with them in some applications they need to be compressed, more or less depending on the purpose of the application and it reduces the size of the files. The need for an efficient technique for compression of Images ever increasing because the raw images need large amounts of disk space store seems to be a big disadvantage during transmission & storage.so that compression techniques are used. There are various algorithms that performs this compression in different ways; some are lossless and keep the same information as the original image, some others loss information when compressing the image. In this paper we have to propose the Lossless technique method of image compression and decompression using a simple coding technique called Huffman coding. This lossless technique is simple in implementation and utilizes less memory.

**Keywords:** Huffman coding algorithm, Digital Image Processing

## I. INTRODUCTION

Compression refers to reducing the quantity of data used to represent a files, images without excessively reducing the quality of the original data in this. It also reduces the number of bits required to store and transmit digital media using compression. To compress something means that you have a piece of data and images you decrease its size. Image compression is the applications of its data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form .[9]Another one is to find out which parts of the data are not really important and just leave those away Huffman coding is a lossless data compression technique. Huffman coding is based on the frequency of the occurrence of a data items i.e. pixel in images. The technique is to use a lower number of bits to encode the data in to binary codes that occurs more frequently and decode the data. Data compression has become requirement for most applications in different areas such as computer science, Information technology, communications, medicine etc. In computer science, Data compression is defined as the science or the art of representing information in a compact form. [8]The comparison between two compression algorithms helps us to use a suitable technique for various applications. A good Image will be analyzed in terms of better picture quality parameters along with memory space requirement which is very important for an image analysis. Image compression is one very important application of Digital Image Processing. Image Compression is the application of data compression on

digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form [9].Data compression has become requirement for most applications in different areas such as computer science, Information technology, communications, medicine etc. In computer science, Data compression is defined as the science or the art of representing information in a compact form [10].

## II. HUFFMAN CODING

Image compression technique basically divided into two parts, loss technique and lossless technique for loss technique, as its name suggest that a little or small part of information is vanished out or converted into noise factor during the process of image compression. In case of lossless technique will get compressed image without loss of information. That's why lossless technique is better as compare with loss technique.[4] So, in order to have better characteristics of image compression, author using Huffman coding in this paper, which come under loss less technique with more prominent features in various application [5]such as medical survey and analysis, technical drawing etc. As we know that Huffman coding algorithm is a step by step process and involves the variable length codes to input characters & it is helpful in finding the entropy and probability of the state. Huffman algorithm is comparatively easier because of its simpler mathematical calculation in order to find the various parameter. Original image can be reconstructed with the help of digital image restoration. [6] Huffman coding is based on the frequency of occurrence of a data item and images. The principle is to uses a lower number of bits to encode the data that occurs more frequently. Codes are stored in a Code Book which may be constructed for each image or a set of images and also data. In all cases the code book plus encoded data must be transmitted to enable decoding. For a set of symbols with a uniform probability distribution and a number of members which is a power of two, Huffman coding is equivalent to simple binary encoding e.g., ASCII coding. The aim of image compression is to remove unwanted information from image so that it can be able to transmit or store data in an efficient form. Compression basically means removing unwanted information from image which only lead to the enhancement of memory space requirement without affecting quality of image. [7]

### A. Construction of Huffman Coding Algorithm:

The algorithm builds the tree t corresponding to the optimal code represent in a bottom up manner. It begins with a set of |c| leaves and perform |c|-1merging operations to create the final tree. In the Huffman algorithm n denotes the number of set of characters denotes the parent node and x&y are the left&right child of z respectively.

- 1)  $n=|c|$
- 2)  $Q=c$

- 3) For i=1 to n-1
- 4) Do
- 5) Z=allocate \_Node()
- 6) X=left[z]=extract\_Min(Q)
- 7) Y=right[z]=extract\_Min(Q)
- 8) F[z]=f[x]+f[y]
- 9) Insert(Q,Z)
- 10) Return extract\_Min(Q)

**B. Analysis:**

- 1) The Q is initializes as a priority queue with the character c
- 2) Q=C can be performed by using build heap in O(n) tree.
- 3) For loop takes (|n|-1) times because each heap operation requires O(log n) time.
- 4) Hence the total running time of Huffman code on the set of n characters is O(n log n).
- 5) system setting and user key generation

**C. Method:**

The procedure is applied for the construction a Huffman tree:

- 1) Search for the two nodes having the lowest frequency, which are not yet assigned to a parent node.
- 2) Couple these nodes together to a new interior node
- 3) Add both the frequencies and assign this value to the new interior node.
- 4) The procedure has to be repeated until all nodes are combined together in a root node.

| Symbol | Probability | code word |
|--------|-------------|-----------|
| A1     | 0.2         | 01        |
| A2     | 0.4         | 1         |
| A3     | 0.2         | 000       |
| A4     | 0.1         | 0010      |
| A5     | 0.1         | 0011      |

Table 1: Huffman coding using probability

A source generates 4 different symbols{A1,A2,A3,A4,A5} with probability {0.2,0.4,0.2,0.1,0.1}.Generate a binary tree from left to right taking the two less probable symbols, putting them together to form another equivalent symbol having the probability that equals the sum of the two symbols. Keeps on doing it until you have just one symbol. Then read the tree backwards, from right to left, assigning different bits to different branches in this tree

|     |     |     |     |     |
|-----|-----|-----|-----|-----|
| A2  | A1  | A3  | A4  | A5  |
| 0.4 | 0.2 | 0.2 | 0.1 | 0.1 |

Table 2: Example probability.

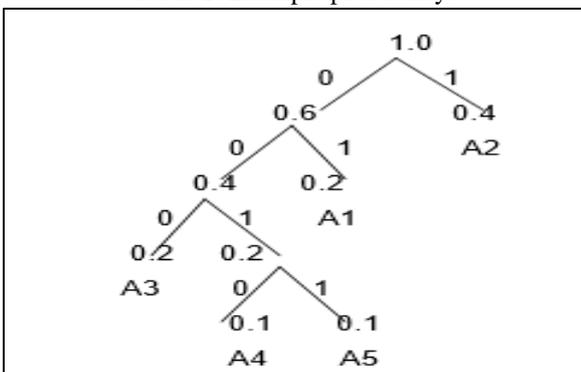


Fig. 1: huffman coding tree using probability

**D. Data Compression:**

Each node of the tree are represented with a byte symbol and the frequency of that byte on the data. Using text ABCBAACD as example. The creation of the Huffman coding tree have the following steps:

- 1) Scan the data and calculate the frequency of occurrence of each and every byte
- 2) Insert those nodes into the reverse priority queue based on the frequencies.
- 3) Starts a loop until the queue is empty
- 4) Remove the two nodes from the queue and combine them into a internal node with the frequency equal to the sum of the two nodes frequencies
- 5) Insert a two nodes removed from the queue as children of the created internal node
- 6) Insert the created to the internal node into the queue

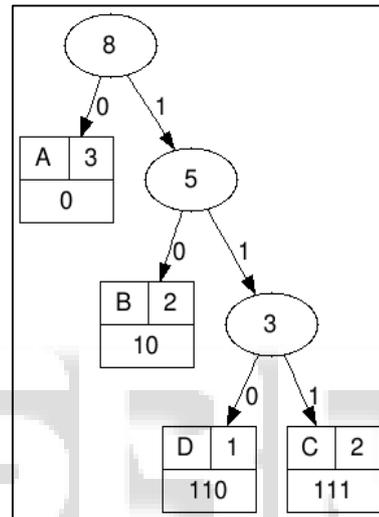


Fig. 2: Huffman tree based on text

So the new representation of the bytes on the text are:

- A: 0
- B: 10
- C: 111
- D: 110

**III. APPLICATIONS**

- 1) Arithmetic coding can be viewed as a generalized of Huffman coding; indeed, in practice arithmetic coding is often preceded by Huffman coding, as it is easier to find an arithmetic code for a binary input than for a non-binary input of Huffman coding.
- 2) Huffman coding is in wide use because of its simplicity, high speed, performance and lack of encumbrance by patents.
- 3) Huffman coding today is often used as a "back-end" to some other compression method. Multimedia codecs such as JPEG and MP3 have a front-end model and quantization followed by Huffman coding.

#### IV. RESULT AND ANALYSIS

##### A. For String:

```
Command Window
New to MATLAB? See resources for Getting Started.

Enter data ['t','a','k','e','o','f','f']
aeffkot
0.1429
0.1429
0.2857
0.1429
0.1429
0.1429

binarycode(1) =
    0    1    1

binarycode(2) =
    0    1    0
```

```
Command Window
New to MATLAB? See resources for Getting Started.

binarycode(3) =
    0    0

binarycode(4) =
    1    0    1

binarycode(5) =
    1    0    0

binarycode(6) =
    1    1

fx >> ['t','a','k','e','o','f','f']
```

##### B. For Image:



##### C. For Character:

```
Command Window
New to MATLAB? See resources for Getting Started.

Enter symbols- 'ABCDCB'
Enter value of probability- [0.2,0.3,0.4,0.5,0.6,0.1,0.4]
```

```
Command Window
New to MATLAB? See resources for Getting Started.

code =
D

code =
D    01

code =
D

code =
D    11

code =
C
```

```
Command Window
New to MATLAB? See resources for Getting Started.

code =
C    100

code =
B

code =
B    100

code =
B

code =
B    110
```

```
Command Window
New to MATLAB? See resources for Getting Started.

code =
B    110

code =
A

code =
A    0010

code =
C

code =
C    1010

fx >>
```

#### V. CONCLUSION

I have studied compression techniques on the basis of their use in different applications and their advantages and disadvantages. The Huffman coding is very efficient for more

frequently occurring sequences of pixels with fewer bits and reduces the file size dramatically. Huffman algorithm is based on statistical models which adds to overhead. The above discussed algorithms use lossless compression technique. JPEG technique which is used mostly for image compression is a lossy compression technique. The Huffman coding assign an integral number of bits to each symbol. It is not universal, it is only valid for one particular type of source.

#### REFERENCES

- [1] A. K. Jain, Image data compression: A review", in Proc. IEEE, vo169, pp. 349-389, 1981.
- [2] Gilbert Held and Thomas R. Marshall. Data and Image Compression: Tools and Techniques
- [3] Compressed Image File Formats: JPEG John Miano, August 1999
- [4] A.K.Jain,Image data compression: Areview", in proc.IEEE,Vo169,pp.349-389,1981
- [5] R.G.Galleger,"variations on a theme byHuffman,"IEEE trans.inform.theory,vol.IT-24,pp.668-674,Nov.1978
- [6] "The past, present, and future of image and multi-dimensional signal processing", 1981mdsp\_review 1998
- [7] R. C. Gonzalea and R. E. Woods, "Digital Image Processing", 2nd Ed, Prentice Hall, 2004.
- [8] Sachin Dhawan,"A Review of Image Compression and Comparison of its Algorithms", Dept. of ECE, UIET, Kurukshetra University, Haryana, India. IJCET VOL.2, Issue 1, March 2011
- [9] Sachin Dhawan,"A Review of Image Compression and Comparison of its Algorithms", Dept. of ECE, UIET, Kurukshetra University, Haryana, India. IJCET VOL.2, Issue 1, March 2011.
- [10] M. Pu, Fundamental Data compression. Butterworth Heinemann, 2006