IMPROVED IMAGE COMPRESSION USING LOSSLESS HUFFMAN ENCODING (I2COM)

Nirajan Bist
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
neonbest12@gmail.com

Suraj Joshi
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
surajjoshi0123@gmail.com

Abhishek Karki
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
abhishekkarki17@gmail.com

Bharat Raj Joshi
Department of Computer and Electronics Engineering
Kantipur Engineering College
Lalitpur, Nepal
br14joshi@gmail.com

***Abstract*—With the development of information technology, image has become the mainstream of information transmission. Even if the storing capacity of storage devices has increased it is not sufficient for enormous pool of those images. The purpose of this work is to demonstrate robustness of simplistic implementation of Huffman encoding in lossless image compression. Some of the formats like JPEG 2000, PNG, etc. are available for lossless compressed image formats are highly used in practice currently but I2COM provides better compression ratio than these available formats in case of small sized images. We have compressed image data along with mapping table which increased the compression Ratio about 30% as compared to compression without compressing the mapping table. I2COM is lightweight image compressor which takes uncompressed BMP image file and compresses it using Huffman Encoding among various available lossless compression technique. Compression is based on various redundant information in file and I2COM uses the frequency of color as a redundant information. Resulting compression ratio (ratio of original size to compressed size) can be normally expected up to 7:1 or more depending upon the number of colors used in the image.**

***Index Terms*—Lossless Image Compression, Huffman Encoding, Compression Ratio**

I. INTRODUCTION

Compression of images is a very useful application for saving the storage data. Main motive of compression is to diminish superfluity or redundant information of the image data in order to make storage or transmission more efficient [1]. The goal of the compression is to keep the image quality intact and minimum processing resources. Images may contain spatial redundancy, coding redundancy, grayscale redundancy and frequency redundancy [2]. This project focus on frequency redundancy of colors to reduce the image file size. The compression may be Lossy or Lossless on the fact that the original image reconstruction ability after compression. Lossy compression involves quantization and entropy coding whereas Lossless compression involves entropy coding only [3]. Huffman encoding serves as lossless compression technique that exploits sequences statistics by assigning frequently used symbols fewer bits than rare symbols [4]. Appropriate headers are to be added to the compressed file. The field of image compression continues to grow at a rapid pace. As we look to the future, the need to store and transmit images will only continue to increase faster than the available capability to process all the data. Image compression involves reducing the size of image data files, while retaining necessary information. Retaining necessary information depends upon the application.

II. METHODOLOGY

A. Background

Huffman coding is a method for lossless data compression and it is independent of data type. Coding process begins by collecting all the probabilities for particular color in descending order. The process starts from bottom i.e leaf nodes and finally making a Huffman tree. At every leaf of tree, color nodes are present with each node have color frequency and color value in RGB format. The process follows in steps. In each step, two color nodes having smallest probabilities are extracted from the sorted list of color nodes. The color nodes are linked by a new node with its probability as sum of probability of those two nodes and is added to the tree. The process goes on until only one node remains which becomes root node for Huffman tree. In this paper the Huffman technique will be accomplished or applied after breaking the image into small parts (rows and columns) and apply this technique to each part.

The problem can be divided into three major parts: Image Acquisition: In our work, image is initially read from user to be decompressed through User Interface (UI) and then it is converted into numpy array with the aid of Python Imaging Library (PIL) module in Python.

Image Compression: Frequency table of the unique colors used in the image is formed using Priority Queue. Finally table along with mapped binary values of individual image pixels are written in compressed file.

Decompression: First table is loaded in a dictionary, and reverse mapping of the image pixels is performed to get actual color of individual pixel which gives a numpy array. Finally we write the image from the array to a file using Python Imaging Library (PIL) module.

B. *Algorithm*

Algorithm 1 Encoding Algorithm

```
Read image to be compressed.
Initialize Priority Queue (P) empty.
for all pixel in image do
    if pixel color not in P then
        Form node(N) with frequency = 1
        Put N into P
    else
        Increase color frequency by 1
    end if
end for
Prepare Huffman Tree using the Priority Queue
Create Mapping Dictionary containing color codes
    for each color from the tree
Save Mapping Table to file.
for all pixel in image do
    Write corresponding codes of the pixel color in
        file using dictionary
end for
```

III. Image Compression

Huffman coding can be demonstrated clearly by compressing a raster image. Suppose we have a 5x5 raster image with 24-bit color(8-bit for each channel). The uncompressed image will take 5 x 5 x 24 = 600 bits of storage.

First, we count up how many times each color occurs in the image. Then we sort the colors in order of decreasing frequency. We end up with a row that looks like Fig. 1.(b)

Now we put the colors together by building a tree such that the colors farthest from the root are the least frequent. The colors are joined in pairs, with a node forming the connection. A node can connect either to another node or to a color. In our example, the tree might look like Fig. 1.(c)

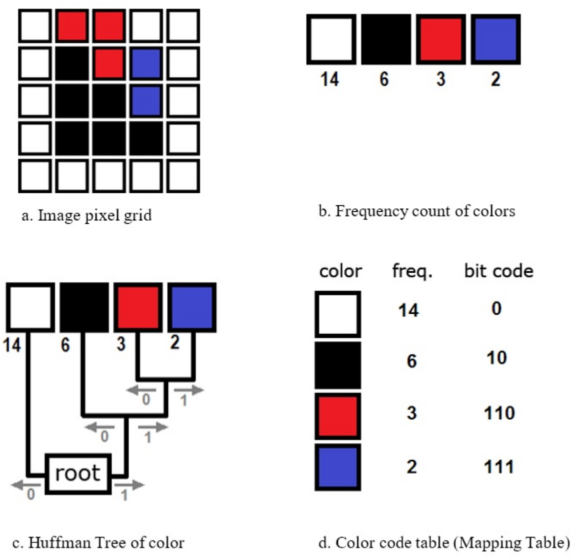

Fig. 1. Example of image compression process [5]

Our result is known as a Huffman tree. It can be used for encoding and decoding. Each color is encoded as follows. We create codes by moving from the root of the tree to each color. If we turn right at a node, we write a 1, and if we turn left 0. This process yields a Huffman code table in which each symbol is assigned a bit code such that the most frequently occurring symbol has the shortest code, while the least common symbol is given the longest code.

Because each color has a unique bit code that is not a prefix of any other, the colors can be replaced by their bit codes in the image file. The most frequently occurring color, white, will be represented with just a single bit rather than 8 bits. Black will take two bits. Red and blue will take three. After these replacements are made, the 600-bit image will be compressed to 14 x 1 + 6 x 2 + 3 x 3 + 2 x 3 = 41 bits.Further for decompression we need to save huffman color mapping table, and image size.

A. *Comparative Measures*

The compression ratio is defined as the ratio of original file size to the compressed file size.

$$C_R = \frac{s1}{s2} \tag{1}$$

Where s1 is the data rate of original image and s2 is that of the encoded bit-stream.

TABLE I
Comparative Compression Ratio Table

Image Name	Compression Ratio(CR)		
	I2COM	*JPEG 2000*	*PNG*
threecolor.bmp	12.01	10.12	10.32
leaves.bmp	11.34	10.11	9.27
ovalcircle.bmp	12.09	10.33	9.24
car.bmp	6.59	8.74	7.84
roads.bmp	3.71	7.64	6.52

Fig. 2. Images used in comparison

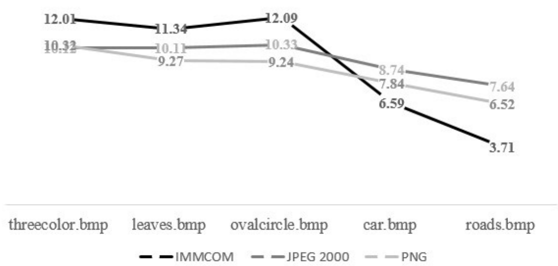

Fig. 3. Comparison Chart for Compression Ratio

TABLE II
UNCOMPRESSED BMP FILE INFORMATION TABLE

Image Name	***Size(KB)***	***Width(px)***	***Height(px)***
threecolor.bmp	187	270	236
leaves.bmp	4110	1600	900
ovalcircle.bmp	910	513	758
car.bmp	5774	1908	1044
roads.bmp	8593	2048	1260

IV. Conclusion and Future Enhancements

The method of splitting an image into different rows and columns will not only provides better results in image compression but also helpful for security purpose of the image transmission. Huffman Encoding method for the image compression has given fair compression ratio average 7:1. This has advantage over complex iterative and progressive algorithms that it is straight forward algorithm to focus on color frequency redundancy. It has turned out to be simple but robust method of image compression. We have compressed image data along with mapping table which increased the compression ratio about 30% as compared to compression without compressing the mapping table. Thus we got some improvement in the compression size in expense of computational speed.

Further we can see that the images compressed with I2COM technique is more efficient in compressing the small sized images than than larger images in comparison to two other lossless compression techniques JPEG 2000 and PNG.

This project can be extended to support lossless compression of videos as well. Other algorithms like Lempel-Ziv-Welch ,a highly advance universal lossless compression, coordinated with this project may help in compressing the larger images efficiently.

Acknowledgment

The authors acknowledge Er. Bishal Thapa and Er. Dipesh Shrestha for co-operating with us and providing us valuable guidance in completing this project .

References

[1] A. K. Jain, Image data compression: A review", in Proc. IEEE, vo169, pp. 349-389, 1981.

[2] G.C Chang Y.D Lin (2010) An Efficient Lossless ECG Compression Method Using Delta Coding and Optimal Selective Huffman Coding IFMBE proceedings 2010, Volume 31, Part 6, 1327-1330, DOI: 10.1007/978-3-642-14515-538.

[3] . Dr. T. Bhaskara Reddy, Miss.Hema suresh yaragunti, Dr.S.kiran, Mrs.T.Anuradha A novel approach of lossless image compression using hashing and Huffman coding ,International Journal of Engineering research and technology ,vol.2 issue 3,march-2013.

[4] M. Yang and N. Bourbakis. An Overview of Lossless Digital Image Compression Techniques, IEEE, pp. 1099-1102. 2005

[5] Huffman Coding - Base of JPEG Image Compression. [online] Available at: https://www.print-driver.com/stories/huffman-coding-jpeg