Lossless Image Compression Technique using Huffman Coding

[1]Aashish Nath Tiwari, *[2]Avinash Rajoriya

[1]*Research Graduate*, [2]*Assistant Professor*

Department of Electronics and Communication Engineering, Lovely Professional University, Punjab, India

Abstract - The problem of saving more amount of data in less space is solved using compression technique. Image and video compression technique are used in daily routine life such as DVD technology and MRI scan. In Lossless image compression technique, the decompressed image is identically same to the original image. This paper approached a lossless image compression technique by using Huffman encoding and decoding. Implementation of this technique is easy to use and utilize less memory. The MATLAB software used to implement the Huffman encoding and decoding function to perform the image compression.

Keywords: Huffman coding, redundant data, compression ratio.

I. INTRODUCTION

Image processing is a handling of images using mathematical operation, where the input can be image or combination of images or video and the output after processing an image is an image which is related to the image applied. Image compression is a specific application of data compression techniques, that encode the original image with fewer bits [1] The goal of image compression is to reduce the redundancy of the image. Image compression plays a very important role in many applications like storage of image database, communication with image, remote sensing (use of satellite image for weather prediction). There is different type of image compression techniques, broadly classified into two classes Lossless and Lossy image compression techniques. The lossless compression techniques mean no information regarding the image is lost or in another term, the reconstructed image from the compressed image is equal to the original image in every sense, whereas in Lossy image compression techniques, some of the information is lost. In other words, the reconstructed image from the compressed is comparable to the original image but not identical to original image. This paper, proposes lossless image compression and decompression using Huffman coding. Which compress the different types of image format (i.e. jpg, png, etc.). The performance parameter of image compression is CR, MSE (cumulative squared error between compressed image and decompressed image), PSNR (used to check the quality of reconstructed image) [2].

II. IMAGE MODEL

The main objective of this paper is to compress the image by reducing the redundant data in input image using Huffman encoding and reconstructing the image by Huffman decoding without any effect on quality of an image.

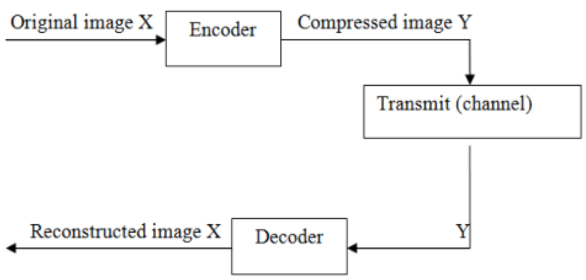

Fig. 1 Block diagram of image compression model

In image compression there are three basic data redundancies as follows [3]:

1) Coding redundancy: - A code is a system of symbols used to represent a set of the event. It occurs when less than optimal code words are used.

2) Interpixel redundancy: - It occurs when pixels are correlated spatially (i.e., the pixel is similar to or dependent on neighbouring pixels). This type of redundancy is also called spatial and temporal redundancy

3) Psychovisual redundancy. It occurs when some information is ignored by the human visual system.

Types of compression techniques

There are two types of compression techniques. These are Lossless and Lossy compression techniques. In. In Lossless compression, the decompressed image is identical to the original image, while in Lossy compression decompressed image is relatively equal to the input image. Though Lossy compression techniques provide higher compression ratio than lossless compression techniques but it losses the information. Some lossless and Lossy compression techniques are given below [4]:

Lossless compression techniques

a. Run length coding

b. Huffman coding

c. LZW coding

d. Arithmetic coding

Lossy compression techniques

a. Predictive coding

b. Transformation coding

c. Block Truncation coding

d. Sub-band coding

III. PREVIOUS WORK

There are many compression algorithms have been developed and used in previous year. Some of the algorithms are of general purpose use, i.e., it can be used to compress the different type of files (e.g., image files, text files, video files, etc.). Another algorithm compresses a particular type of files. It has been comprehended that, according to the form of data at which compression techniques is performed, below we are reviewing some of the literature review belonging in this area.

Dalvir Kaur and Kamaljit Kaur [4] present a lossless image compression with three components: Huffman coding, LZW coding, and retinex algorithm. These methods are called Huffman based LZW Lossless Image Compression using Retinex algorithm. this method works in fourth stages, first the image is compressed by Huffman encoding techniques and second stage Huffman codes are concatenated together and compressed by LZW encoding. We get the compressed image as a result.in the third stage, the compressed image is decoded by LZW techniques and further it is decoded by Huffman techniques the result is the decoded image. the fourth step is Retinex algorithm it is used for improving the quality and contrast of decoded image

Jagadish H. Pujar and Lohit M. Kadlaskar [5] propose a simple algorithm for image compression and decompression using Huffman coding which is used on text string as well as different image file formats for performance analysis. we find that the original image is identical to the decoded image. We also find the higher code redundancy helps to achieve more compression. This technique is used for scan testing to reduce test data volume, test data compression, and compression time. Comparing the result, it concluded that Huffman coding is an efficient technique for image compression and decompression to some extent. It is very useful techniques for compressing the text and general types of image.

Suresh Yerva, Smita Nair, Krishnan Kutty [6] present lossless image compression techniques in the spatial domain for the continuous image by image folding concept. This method is based on the use of adjacent neighbour redundancy for prediction. the compression process occurs in two ways, that are row folding and column folding. for column folding, the even column elements are subtracted from odd column elements, similarly applied on odd columns element. For row folding, the elements of even row are subtracted from the odd rows element and the resultant odd rows element are used for next level. The difference data is stored in a tile format., and this method is compared with the the Set Partitioning in Hierarchical Trees (SPHIT) algorithm.

Hyoung Joong Kim et al [7] present a new lossless data compression technique. which can be used for image and video? This method is based on CABAC (Context Adaptive Binary Arithmetic Coder) algorithm. the basic idea of this method is hiding of the data reversibly with a location map

IV. PROPOSED METHODOLOGY

Huffman Coding

Huffman coding techniques is based on the following observation:

1) First create a list of symbol, and sorted by frequency or probability from largest to smallest

2) Combine two symbol which has the smallest frequency

3) Re-sort the probability list. And repeat the step 1,2,3 until the all of the probability have been added up

The final number at above of the tree should be the sum of all probability. Let us consider an example shown in table 1.

Symbol	Probability	code	Average length
A	0.36	1	0.36
B	0.15	001	0.45
C	0.13	010	0.39
D	0.11	0011	0.44
E	0.09	0010	0.36
F	0.07	0001	0.28
G	0.05	00001	0.25
H	0.03	000001	0.18
I	0.01	000000	0.08

In the first step, select the two symbol that has two lowest probabilities, add them and form a new code with combined probability we assign '1' to one of the two nodes and '0' to another. And continue this process until it is left with one symbol with probability equal to '1'. To determine the code, assigned the '0' and '1' on every node. The average length of the code is calculated below: Lavg=0.36*1+(0.15+0.13)*3+(0.11+0.09+0.07)*4+0.05*5 +(0.03+0.01) *6=2.77 bits/symbol and calculate the entropy source is given by equation shown below and result is 2.68 bits/symbol and resultant efficiency of the code is 2.68/2.77 is 96.75%

$$H = -\sum_{k=0}^{L-1} P(r_k)\log_2 P_r(r_k)$$

Developing the algorithm:

Step1: Read the image on MATLAB

Step2: Convert the colour image into grey level image and resize it into 512*512

Step3: call the function which will find the symbol and calculate the probability of each value in input image data.

Step4: arranged the probabilities in descending order.

Step5: Add the two probabilities which has lowest value and repeat this step until only two probabilities are left

Step6: Assign longer code word for lowest probability symbol, and shorter code for highest probability symbol

Step7: This generate a code dictionary binary tree.

Step8: Call the function which performs the encoding process resultant image is called compressed image

Step9: The compressed image is decompressing by the decoding function (with the help of generated binary tree) resultant image is called decompressed image.

Step10: Find the compression ratio, the compression ratio is defined as

$$\text{CR}= \frac{\textit{original image size}}{\textit{compressed image size}}$$

Step11: Calculate the MSE (mean square error) and it is given as

$$\text{MSE} = \frac{1}{\text{N}\times\text{N}}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}[\text{f}(x,y) - F(x,y)]^2$$

Step12: Calculate the PSNR, and it is given as:

$$\text{PSNR} = 10\log_{10}\frac{\text{Peak Value}^2}{\text{MSE}}$$

Peak value is the maximum possible value of the image [8].

V. EXPERIMENTAL RESULTS

A number of test images used are a lena, a mandrille, a building and a barbara. An algorithm used is Huffman coding which is used to compress the image. A size of an image was taken is 512x512. Compression of the data is depending on the redundant data present in the image as shown in Table 2. This study includes the variety of images as mentioned in table 2 and shown the image "building" in figure 2: Original image 2(a), Decompressed image 2(b)

Table 2: Result shows Compression Ratio, CT (compression Time), MSE (mean square error), PSNR (peak signal to noise ratio), RD (redundant data).

S NO.	Image	CR	CT	RD(%)	MSE	PSNR
1	Lena	2.0564	1.3419 s	51.37	0	Inf
2	Mandrill	4.3096	4.4551 s	76.80	0	Inf
3	Building	3.4125	1.3708 s	70.77	0	Inf
4	Barbara	1.8678	4.4512 s	46.46	0	Inf

Figure 2: 2(a) Original image 2(b) Huffman decoded image

The figure 2 and table2 shows the decoded image is identical to the original image in every sense

IV. CONCLUSION

The above table 2 shows that the achievement of higher compression ratio depends on the percentage of redundant data available in an image. It shows that the higher redundant data implies higher compression ratio. The experimental result shows up to a 76.80% data saving space. The experimental results show the MSE is 0 and PSNR is infinite. The decompressed image is exact Lossless with Huffman coding.

V. FUTURE SCOPES

As the future work includes improvement of compression ratio using new techniques and trying to minimize the compression time. The proposed work would be done on different kind of data such as text, audio and video as till now it is restricted to the image.

REFERENCES

[1] Cooper, T. K., & Desoky, A. H. (2015). Huffman Coding Analysis of XOR Filtered Images, 237–241.

[2] KUMAR, S., Bhopale, S. D., & Naik, R. R. (2014). Implementation of Huffman Image Compression and Decompression Algorithm, 2(4), 1356–1358.

[3] Alarabeyyat, A., Al-Hashemi, S., Khdour, T., Btoush, M. H., Bani-Ahmad, S., & Al-Hashemi, R. (2012). Lossless Image Compression Technique Using Combination Methods. *Journal of Software Engineering and Applications*, *5*(October), 752–763

[4] Kaur, D., & Kaur, K. (2013). Huffman Based LZW Lossless Image Compression Using Retinex Algorithm, *2*(8), 3145–3151

[5] Pujar, J. H., & Kadlaskar, L. M. (2010). A new lossless method of image compression and decompression using Huffman coding techniques. *Journal of Theoretical and Applied Information Technology*, *15*(1), 18–23

[6] Yerva, S., Nair, S., & Kutty, K. (2011). Lossless Image Compression based on Data Folding, 999–1004.

[7] A NEW LOSSLESS DATA COMPRESSION METHOD Hyoung Joong Kim Korea University Seoul 136-701, Korea. (2009), 1740–1743.

[8] Kulkarni, A. (n.d.). Gray-Scale Image Compression Techniques: A Review, *131*(13), 22–25.