International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

TIFF Image Compression through Huffman Coding Technique

Anis Suliman Ali Bakouri

Om-Alaranb College of Science & Technology, Libya

Abstract: ***The goal of image compression is to eliminate the redundancy in a file's code in order to reduce its size. It is useful in reducing the image storage space and in reducing the time needed to transmit the image. Image compression is more significant for reducing data redundancy for save more memory and transmission bandwidth. An efficient compression technique has been proposed which combines DCT and Huffman coding technique. This technique proposed due to its Lossless property, means using this the probability of loss the information is lowest. Result shows that high compression rates are achieved and visually negligible difference between compressed images and original images...***

Keywords: Huffman coding, Huffman decoding, TIFF, PSNR, MSE

1. Introduction

Image compression is a technique in which large amount of disk space is required for the raw images which seems to be a very big disadvantage during transmission and storage. With increase in technology an efficient technique for image compression is needed. Even though there are so many compression techniques which is present already, but the need for better compression technique is required which is faster, memory efficient and simply suits the requirements of the user. To analyze the parameters of image compression Peak Signal to noise ratio and compression ratio is an important parameter it gives synthetic performance of the compression of images. Image is a 2-Dimensional signal represented by Digital system. Normally Image taken from the camera is in the analog form. However, for processing, transmitting and storage, images are converted in to digital form. A Digital Image is basically two-Dimensional arrays of pixels. Basically, compressing an image is different from compressing digital data. Data compression algorithm which is generally used for Image compression but it gives us result which is less than optimal. In remote sensing, bio medical and video processing techniques different types of images are used which require compression for transmission and storage. Compression is achieved by removing redundancy or extra bits from the image. There are numerous data compression algorithm which can be considered as universal number of universal compression algorithms that can compress almost any kind of data. These are the lossless methods they retain all the information of the compressed data. However, they do not take advantage of the 2-dimensional nature of the image data. Images have certain statistical properties, which can be exploited by encoders especially designed for them. Also, for the sake of saving a little more storage space and bandwidth some of the finer details in the image can be sacrificed. In this paper, a new technique to achieve image compression algorithm is proposed that combines a DCT transform and Huffman coding. Huffman coding is a well - known algorithm for generating minimum redundancy codes as compared to other algorithms. The Huffman coding has effectively used in text, image and video compression. The DCT transform is not new to image coding problems, some approaches based on DCT transforms have been recently reported in literature. The synergy of DCT is for better tackle the problem has been proposed. The method is shown to efficiently encode images in terms of high peak signal to noise ratio (PSNR) values.

2. Image Compression Techniques

The image compression techniques are broadly classified into two categories depending whether or not an exact replica of the original image could be reconstructed using the compressed image. These are:
1) Lossless technique
2) Lossy technique

2.1 Lossless compression technique

In lossless compression techniques, the original image can be perfectly recovered from the compressed (encoded) image. These are also called noiseless since they do not add noise to the signal (image). It is also known as entropy coding since it uses statistics/decomposition techniques to eliminate/ minimize redundancy. Lossless compression is used only for a few applications with stringent requirements such as medical imaging.

Following techniques are included in lossless compression:
1) Run length encoding
2) Huffman encoding
3) LZW coding

2.1.1 Run Length Encoding technique

This is a very simple compression method used for sequential data. It is very useful in case of repetitive data. This technique replaces sequences of identical symbols (pixels), called runs by shorter symbols.

2.1.2 Huffman Encoding

This is a general technique for coding symbols based on their statistical occurrence frequencies (probabilities). The pixels in the image are treated as symbols. The symbols that occur more frequently are assigned a smaller number of bits, while the symbols that occur less frequently are assigned a

relatively larger number of bits. Huffman code is a prefix code. This means that the (binary) code of any symbol is not the prefix of the code of any other symbol

2.1.3 LZW Coding

LZW (Lempel- Ziv- Welch) is a dictionary-based coding. Dictionary based coding can be static or dynamic. In static dictionary coding, dictionary is fixed during the encoding and decoding process. In dynamic dictionary coding, the dictionary is updated on fly. LZW is widely used in computer industry and is implemented as compress command on UNIX.

3. Huffman Algorithm

This coding technique is basically based on frequency of occurrence of a data item. The principle behind this technique is to use lower number of bits to encode the data that occurs more frequently. A Huffman code dictionary, which associates each data symbol with a code-word, has the property that no code-word in the dictionary is a prefix of any other code-word in the dictionary. The basis for this coding is a code tree according to Huffman, which assigns short code words to symbols frequently used and long code words to symbols rarely used for both DC and AC coefficients, each symbol is encoded with a variable-length code.

3.1 Huffman Coding

The Huffman encoding algorithm starts by constructing a list of all the alphabet symbols in descending order of their probabilities. It then constructs, from the bottom up, a binary tree with a symbol at every leaf. This is done in steps, where at each step two symbols with the smallest probabilities are selected, added to the top of the partial tree, deleted from the list, and replaced with an auxiliary symbol representing the two original symbols. When the list is reduced to just one auxiliary symbol (representing the entire alphabet), the tree is complete. The tree is then traversed to determine the code-words of the symbols.

3.2 Huffman Decoding

Before starting the compression of a data file, the compressor (encoder) has to determine the codes. It does that based on the probabilities (or frequencies of occurrence) of the symbols. The probabilities or frequencies have to be written, as side information, on the output, so that any Huffman decoder will be able to decompress the data. This is easy, because the frequencies are integers and the probabilities can be written as scaled integers. It normally adds just a few hundred bytes to the output. It is also possible to write the variable-length codes themselves on the output, but this may be awkward, because the codes have different sizes. It is also possible to write the Huffman tree on the output, but this may require more space than just the frequencies. In any case, the decoder must know what is at the start of the compressed file, read it, and construct the Huffman tree for the alphabet. Only then can it read and decode the rest of its input. The algorithm for decoding is simple. Start at the root and read the first bit off the input (the compressed file). If it is zero, follow the bottom edge of the tree; if it is one, follow the top edge. Read the next bit and move another edge toward the leaves of the tree. When the decoder arrives at a leaf, it finds there the original, uncompressed symbol, and that code is emitted by the decoder. The process starts again at the root with the next bit. Decoding a Huffman-compressed file by sliding down the code tree for each symbol is conceptually simple, but slow. The compressed file has to be read bit by bit and the decoder has to advance a node in the code tree for each bit.

4. Proposed Algorithm

Algorithm for compression of image using Huffman techniques with Discrete Cosine Transform has been proposed in this section. The algorithm of the proposed method is:

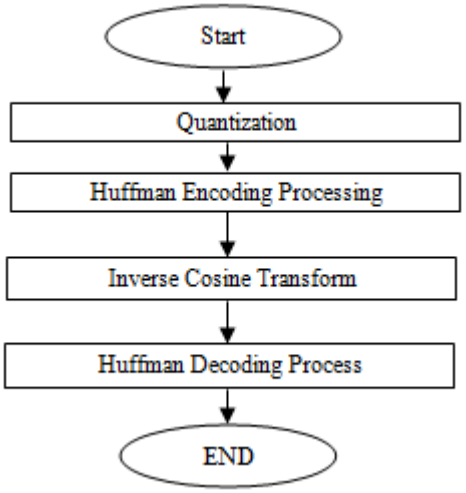

Figure 1(a): Flowchart of Huffman Algorithm

4.1 Simulation Parameters

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB stands for matrix laboratory, and was written originally to provide easy access to matrix software developed by LINPACK (linear system package) and EISPACK (Eigen system package) projects. MATLAB is therefore built on a foundation of sophisticated matrix software in which the basic element is array that does not require pre dimensioning which to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of time.

5. Result and Discussions

In this paper MATLAB Simulator has been used to evaluate the performance of Huffman coding technique in the field of image compression behalf of compression parameters. The proposed algorithm has been applied on the different images having TIFF and JPEG format.

In this paper MATLAB Simulator has been used to evaluate the performance of Huffman coding technique in the field of image compression behalf of compression parameters. The proposed algorithm has been applied on the different images having TIFF format.

Figure 2(a): cat original image with image size 512 ×512.

Figure 2(b): cat compressed image with 180×160

Figure 2(a) shows cat original image with image size 512 ×512. in TIFF format When proposed algorithm has been applied on the cat image which is showing in figure 4(a) then obtained the cat compressed image with 180×160 which is showing in figure 2(b).

Compression rate shows that how much an image can be compressed from its original size. There are two error metrics which is used to compare the quality of image compression, that are known as MSE and PSNR.

The MSE represents the cumulative squared error between the compressed and the original image whereas PSNR represents a measure of the peak error. Lower the value of MSE lowers the error and PSNR tells us about the quality of image, more the PSNR value better will be the result.

$$\mathbf{MSE}=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j)-K(i,j)]^2$$

Where m, n is the size of the original image.

$$PSNR_{db} = 10log_{10}\frac{(2n-1)2}{MSE}$$

Table 1: Compression ratio with PSNR using Huffman Coding based on Histogram Information and Image Segmentation

Compression ratio	PSNR
CR= 25%	24..9670
CR= 35%	27.3075
CR= 45%	29.7695
CR= 55%	32.0086
CR= 65%	35.7995
CR= 75%	38.0142

Table 2: Compression ratio with MSE using Huffman Coding based on Histogram Information and Image Segmentation

Compression Ratio	MSE
5	4.864
10	3.215
15	2.973
20	2.468
25	2.096

Table (1) shows that the compression parameters for both the images which is shown in figure.

6. Conclusion

After observing the results, it can be concluded that the objective of the research is achieved with following observation. It is concluded that Image Compression is an important technique in digital image processing. There are different types of compression techniques but Huffman Coding technique is a good compression technique in lossless image compression. Huffman compression is a variable length type of compression technique. In Huffman, the coding redundancy can be eliminated by assigning the codes in better way.

References

[1] Buro M (1993) "*On the maximum length of Huffman codes*", Information Processing Letters, Vol. 45, No.5, pp. 219-223
[2] Chen H, Wang Y, Lan Y (1999), "*A Memory Efficient and Fast Huffman Decoding Algorithm*" Information Processing Letters, Vol. 69, No. 3, pp. 119- 122.
[3] Ganvir N, Jadhav A (2010), "*Explore the Performance of the ARM Processor Using JPEG*" International Journal on Computer Science and Engineering, Vol. 2(1), pp 12-17.
[4] K. Novak, A Comparison of Two lmage Compression Techniques for Softcopy Photogrammetric, Photogrammetric Engineering & Remote Sensing, Vol. 62, No. 6, June 1996, pp. 695-701..
[5] Y. shen, A Medical Image Compression Scheme Based on Low Order Linear Predictor and Most-likely Magnitude
[6] Huffman Code, IEEE International Conference on Mechatronics and Automation June 25 - 28, 2006, Luoyang, China.