

New Algorithm to Find the Missing Strings During Compressed Image Transmission using Huffman Encoding

M. A. P. Manimekelai,

Professor, Dept ECE, Karunya University, Coimbatore-641114

Mariam Stephina Punnose, Rinda Regi, Sherin Alphonsa John

Students, ECE, Karunya University, Coimbatore, India-641114

Abstract - Usually Images require a large disk space during transmission and storage. This is disadvantageous as it consumes more memory and becomes slower and suits the requirements of the user. There are many methods of image compression and decompression but we have used a simple coding technique called Huffman coding in MATLAB platform. This technique is simple in implementation and utilizes less memory but major drawback is some strings of symbols will be missing during the transmission of the compressed image. In this paper we propose a new algorithm to overcome this problem.

Keywords: Huffman coding, PSNR, MSE, Compression Ratio

1. INTRODUCTION

An efficient method for image compression is necessary as normal images take large amount of disk space and transmission speed will be very low. There are many compression techniques. There are two main types of image compression, i.e. lossy and lossless. Lossless compression is preferred for archival purposes and often for medical imaging, technical drawings, clip art, or comics. Lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossy methods are especially suitable for natural images such as photographs in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate. The lossy compression that produces imperceptible differences may be called visually lossless. Lossy compression formats suffer from generation loss: repeatedly compressing and decompressing the file will cause it to progressively lose quality. This is in contrast with lossless data compression, where data will not be lost via the use of such a procedure. Huffman encoding is one of the most qualitative lossless compression technique existing. A digital image after sampling and quantization requires enormous storage. Assuming, a 24 bit color image 512x512 pixels will occupy 768 Kbyte storage on a disk [1]. It takes nearly 4 minutes to transmit an image over a 28.8 kbps modem. The purpose for image compression is to reduce the amount of data required for representing sample digital images and reducing the cost of image transmission.

The compression of images plays a key role in many important applications such as image database, image communications, remote sensing (weather sensing and other earth resource applications). The image compression is done in gray scale with pixel values between 0 and 255. There are different methods of image compression, which include lossy and lossless [2]. In lossless compression no information regarding the image is lost. In other words, the reconstructed image from compressed image is identical to the original image [3]. In lossy compression image information is lost, i.e., the reconstructed image from compressed image is not identical to original image. Huffman encoding generates minimum redundancy codes compared to other algorithms. It is being effectively used in text, image, video compression and conferencing system such as JPEG, MPEG-2, MPEG-4 and H.263 etc. It can calculate the probability value and sort the symbols based on the probability value by collecting the unique symbols from the source image. From the lowest probability value symbol to the highest probability value symbol, two symbols combined to form a binary tree. A synthetic performance of the compression is given by PSNR and Compression Ratio to validate the image compression. Image compression has two main categories [4]. Statistical schemes and dictionary schemes. Huffman coding comes under statistical scheme.

The major disadvantage of image compression and transmission during Huffman encoding is that some strings of symbols will be missing during transmission. The main objective of this paper is to obtain a new algorithm to find the missing strings during transmission time.

The entire paper consists of the following sections:

- i) Introduction
- ii) Principle behind compression
- iii) Error metrics
- iv) Performance Parameters

v) Huffman coding

vi) Algorithm to find missing strings

2. PRINCIPLE BEHIND COMPRESSION:

The common characteristic of most images is that the neighbouring pixels are correlated and therefore contain redundant information. Two fundamental components of compression are redundancy and irrelevancy reduction.

i) Removal of duplication from the signal source

ii) Irrelevancy reduction, omits parts of the signal that will not be noticed by Human Visual System.

In an image that consists of a sequence of images there are three types of redundancies to compress file size

i) Coding Redundancy: fewer bits to represent frequently occurring symbols.

ii) Interpixel redundancy: Neighbouring pixels have almost same value

iii) Psycho Visual redundancy: Human visual system cannot simultaneously distinguish all colours.

3. ERROR METRICS:

Mean Square Error is the cumulative squared error between compressed and original image.

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

4. PERFORMANCE PARAMETERS:

A) *Compression Ratios:*

It is defined as the number of bits to represent the size of original image to the number of bits to represent the size of compressed image [5]. $R = n_1/n_2$ where n_1 and n_2 represents the number of bits required by original and compressed image respectively.

B) *Peak Signal To Noise Ratio (Psnr):*

It's the measure of quality of reconstruction of image. The signal in this case is the original data and the noise is the error introduced by compression. Logically a higher value of PSNR is good because it means that the ratio of signal to noise is higher [6].

$$PSNR = 20 * \log_{10} (255 / \sqrt{MSE})$$

5. HUFFMAN CODING :

The two important observations of Huffman code procedure are:

a) More frequently occurred symbols will have shorter code words than symbols that occur less frequently.

b) The two symbols that occur least frequently will have the same length.

The Huffman code is an entropy encoding algorithm used for lossless data compression [7]. The term refers to the use of a variable length code table for encoding a source symbol where the variable length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It uses a specific method for choosing the representation for each symbol, resulting in a prefix code that expresses the most common source symbols using shorter strings of bits than are used for less common source symbols.

ADVANTAGES:

i) Algorithm is easy to implement [8].

ii) Produce a lossless compression of images.

DISADVANTAGES :

i) Algorithm varies with different formats [8].

ii) Huffman encoding is usually done in two passes. During the first pass a statistical model is built, and then in the second pass the image data is encoded based on the generated model. From here we can see that Huffman encoding is relatively a slow process as time is required to build the statistical model in order to achieve efficient compression rate.

iii) Compression of image files that contains long runs of identical pixels by Huffman is not as efficient when compared to RLE.

iv) One of the major disadvantages is that, the encoded data is corrupted with missing bits, then whatever that is decoded will be wrong values and the final image displayed will not be fully correct. This paper aims at finding the missing bits during transmission of the compressed image using Huffman coding.

6) ALGORITHM TO FIND MISSING STRINGS:

Step 1 : Read the image on the work space of Matlab

Step 2 : Convert the given colour image into gray scale image .

Step 3 : Double mapping is done by DCT (Discrete cosine transform)

Step 4 : Obtain quantization matrix

Step 5 : Assign DCT matrix [8].

Step 6: Include the block processing functions

Step 7: Get the channel for encoding

Step 8: Complete the scale forward DCT

Step 9: If channel=1, compute quantization for luminescence, else compute quantization for colors.

Step 10: Compute the entropy code for the whole channel and get the entropy code without zeroes on the end.

Step 11: Perform Huffman encoding and add to output.

Step 12: Perform Huffman decoding and check if it is correct.

Step 13: Execute entropy decoding and dequantisation steps.

Step 14: By performing the inverse DCT, scale back the original image and set it as the output.

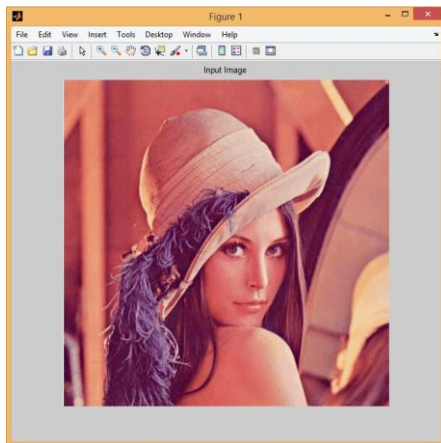
Step 15: To find out the missing strings, remove first symbol and calculate the ratio.

Step 16: Calculate the CR and PSNR [10].

Step 17: Show the results.

RESULTS:

INPUT IMAGE :



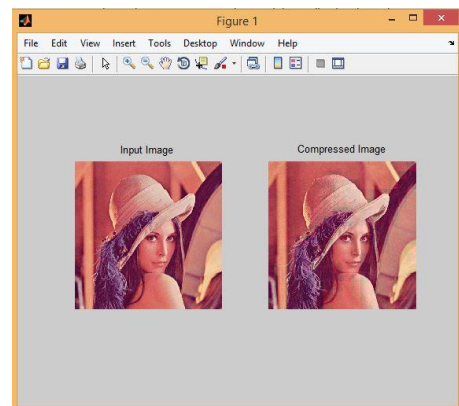
ENTROPY VALUES:

Command Window										
New to MATLAB? Watch this video, see Demos, or read Getting Started.										
0.4796	0.4935	0.4913	0.5000	0.9619	0.2126	0.3928	0.1379	0.5000		
0.4796	0.3928	0.4935	0.4619	0.4533	0.2126	0.4935	0.1379	0.1313		
0.4410	0.0747	0.5449	0.3536	0.1422	0.1379	0.7482	0.3753	0.3536		
0.1379	0.0747	0.2706	0.1513	0.2706	0.8979	0.7482	0.5133	0.4619		
0.2126	0.3928	0.0381	0.5000	0.3087	0.9061	0.4935	0.5879	0.5000		
0.2126	0.4935	0.2706	0.1313	0.2706	0.9061	0.3928	0.5879	0.4619		
0.1379	0.7482	0.5449	0.3536	0.1422	0.1379	0.0747	0.5133	0.3536		
0.5879	0.7482	0.4533	0.4619	0.4533	0.1379	0.0747	0.3753	0.1313		
0.9061	0.4935	0.4619	0.5000	0.9619	0.2126	0.3928	0.1379	0.5000		
0.9061	0.3928	0.4533	0.4619	0.4533	0.2126	0.4935	0.1379	0.1313		
0.5879	0.0747	0.5449	0.3536	0.1422	0.1379	0.7482	0.3753	0.3536		
0.1379	0.0747	0.2706	0.1513	0.2706	0.8979	0.7482	0.5133	0.4619		
0.2126	0.3928	0.0381	0.5000	0.3087	0.9061	0.4935	0.5879	0.5000		
0.2126	0.4935	0.2706	0.1313	0.2706	0.9061	0.3928	0.5879	0.4619		
0.1379	0.7482	0.5449	0.3536	0.1422	0.1379	0.0747	0.5133	0.3536		
0.5879	0.7482	0.4533	0.4619	0.4533	0.1379	0.0747	0.3753	0.1313		
0.9061	0.4935	0.4619	0.5000	0.9619	0.2126	0.3928	0.1379	0.5000		
0.9061	0.3928	0.4533	0.4619	0.4533	0.2126	0.4935	0.1379	0.1313		
0.5879	0.0747	0.5449	0.3536	0.1422	0.1379	0.7482	0.3753	0.3536		
0.1379	0.0747	0.2706	0.1513	0.2706	0.8979	0.7482	0.5133	0.4619		
0.2126	0.3928	0.0381	0.5000	0.3087	0.9061	0.4935	0.5879	0.5000		
0.2126	0.4935	0.2706	0.1313	0.2706	0.9061	0.3928	0.5879	0.4619		
0.1379	0.7482	0.5449	0.3536	0.1422	0.1379	0.0747	0.5133	0.3536		

PERFORMANCE METRICS:

Command Window
New to MATLAB? Watch this video, see Demos, or read Getting Started.
ratio =
0.5000
ms =
17.7933
PSNR =
71.2565
compressed_ratio =
18.7317
fx >>

INPUT-OUTPUT IMAGE:



MISSING BITS:



```

Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
F =
0.5000
F >>

```

CONCLUSION:

The above presented shows a new algorithm to find the missing strings of symbols during compressed image transmission. The technique used for compression is lossy Huffman encoding. The input image is given on the MATLAB platform of size 512x512 pixels in .bmp format. Depending upon the frequency of occurrence of symbols (in the image), they are encoded, quantized and mapped. Further dequantization and decoding is done to obtain back the original image. In this process, few string of bits can be missing while transmission. This paper presents the algorithm to find those missing bits. Also performance parameters such as mean square ratio (MSR) and compression ratio (CR) are calculated to validate the new algorithm. It is seen that whenever the CR increases, error will be minimum. It means that compressed image is almost same as original image.

REFERENCES:

1. A new lossless method of image compression and decompression using Huffman coding techniques- Jagadesh H. Pujaara, Lohit M Kadlaskar, Dept of EEE, BVB college of Engg and Tech, Hubly, India.
2. Lossless image compression- B C Vemuri, S Sahni, F Chen, C Kapoor, C Leonard and J Fitzsimmons
3. A Survey: Various techniques of image compression- Gaurav Vijayvargiya, Dr Sanjay Silakari, Dr Rajeev Pandey- Dept of CSE, UIT-RGPV, Bhopal, India
4. Latent palmprint matching- Anil K Jain, Fellow, IEEE and Jianjing Feng
5. Image compression using Huffman coding based on Histogram information and image segmentation. Dr Monisha Sharma (professor), Mr Chandrashekhar K. (Asst Professor), Lalak Chauhan- Dept of ECE, Sree Sankaracharya College of Engg and Tech, Bhilai, India.
6. A new, fast and efficient image coder based on set partitioning in hierarchical trees- IEEE- A Said, W Pearlman.
7. Literature review of image compression algorithm- Padmaja V K, Dr B Chandrasekhar, Jawaharlal Technological University, Anantapur, India
8. A low complexity, context based lossless image compression algorithm- M J Weinberger, G Seroussi, G Sapiro
9. Experiments with lossless and virtually lossless image compression algorithms- G G Langdon and C AHaidinyak
10. DSalomon: Data compression- The complete reference, Springer Verlag