International Journal of Engineering Research and General Science Volume 3, Issue 4, July-August, 2015
ISSN 2091-2730

A Image Comparative Study using DCT, Fast Fourier, Wavelet Transforms and Huffman Algorithm

Kuldeep Jain, Vimal KumarAgrawal

M.Tech Scholar, kuldeepjain1991@yahoo.com, Apex Institute of Engg. & Tech., ladiwalvimal@gmail.com

Abstract— Image compression is now very important for applications such as transmission and storing data bases. In this paper we review and discuss about the image compression, need of compression, its principles, and various algorithm of image compression. This paper attempts to give a recipe for selecting one of the popular image compression algorithms based on Wavelet, JPEG/DCT, fourier, and huffman approaches. We review and discuss the advantages and disadvantages of these algorithms for compressing true color images, given an experimental comparison on commonly used image of wpeppers.jpg.

Keywords — Image, Compression, Discrete Cosine Transform, Fourier Transform, wavelet Transform and Huffman Algorithm.

INTRODUCTION

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. The development of higher quality and less expensive image acquisition devices has produced steady increases in both image size and resolution, and a greater consequent for the design of efficient compression systems. Although storage capacity and transfer bandwidth has grown accordingly in recent years, many applications still require compression. The basic rule of compression is to reduce the numbers of bits needed to represent an image. In a computer an image is represented as an array of numbers, integers to be more specific, that is called a ― digital image. The image array is usually two dimensional (2D), If it is black and white (BW) and three dimensional (3D) if it is color image [1]. Digital image compression algorithms exploit the redundancy in an image so that it can be represented using a smaller number of bits while still maintaining acceptable visual quality. Factors related to the need for image compression include:

- The large storage requirements for multimedia data
- Low power devices such as handheld phones have small storage capacity
- Network bandwidths currently available for transmission
- The effect of computational complexity on practical implementation.

In the array each number represents an intensity value at a particular location in the image and is called as a picture element or pixel. Pixel values are usually positive integers and can range between 0 to 255. In other words, we say that the image has a grayscale resolution of 8 bits per pixel (bpp). On the other hand, a color image has a triplet of values for each pixel one each for the red, green and blue primary colors. Hence, it will need 3 bytes of storage space for each pixel. The captured images are rectangular in shape [2]. The ratio of width to height of an image is called the aspect ratio.

Data Compression Model

Compression is also known as encoding process and decompression is known as decoding process. A data compression system mainly consists of three major steps and that are removal or reduction in data redundancy, reduction in entropy, and entropy encoding. A typical data compression system can be labeled using the block diagram shown in Figure 1.2 It is performed in steps such as image transformation, quantization and entropy coding. JPEG is one of the most used image compression standard which uses discrete cosine transform (DCT) to transform the image from spatial to frequency domain [3]. An image contains low visual information in its high frequencies for which heavy quantization can be done in order to reduce the size in the transformed representation. Entropy coding follows to further reduce the redundancy in the transformed and quantized image data.
Digital data compression algorithms can be classified into two categories-
1. Lossless compression
2. Lossy compression
Lossless compression
In lossless image compression algorithm, the original data can be recovered exactly from the compressed data. It is used generally for discrete data such as text, computer generated data, and certain kinds of image and video information. Lossless compression can achieve only a modest amount of compression of the data and hence it is not useful for sufficiently high compression ratios. GIF, Zip file format, and Tiff image format are popular examples of a lossless compression [4]
Lossy compression techniques refer to the loss of information when data is compressed. As a result of this distortion, must higher compression ratios are possible as compared to the lossless compression in reconstruction of the image. 'Lossy' compression sacrifices

exact reproduction of data for better compression. It both removes redundancy and creates an approximation of the original. The JPEG standard is currently the most popular method of lossy compression.

Elucidation of Each Algorithm Used

Transform refers to changing the coordinate basis of the original signal, such that a new signal has the whole information in few transformed coefficients. The processing of the signals in the transform domain is more efficient as the transformed coefficients are not correlated [7].

The first step in the encoder is to apply a linear transform to remove redundancy in the data, followed by quantizing the transform coefficients, and finally entropy coding then we get the quantized outputs [8]. After the encoded input image is transmitted over the channel, the decoder reverse all the operations that are applied in the encoder side and tries to reconstruct a decoded image as close as to the original image [9].

Discrete Cosine Transform

The JPEG/DCT still image compression has become a standard recently. JPEG is designed for compressing full-color or grayscale images of natural, real-world scenes. To exploit this method, an image is first partitioned into non overlapped 8×8 blocks. A discrete Cosine transform (DCT) [1] is applied to each block to convert the gray levels of pixels in the spatial domain into coefficients in the frequency domain. The coefficients are normalized by different scales according to the quantization table provided by the JPEG standard conducted by some psycho visual evidence. The quantized coefficients are rearranged in a zigzag scan order to be further compressed by an efficient lossless coding strategy such as run length coding, arithmetic coding, or Huffman coding. The decoding is simply the inverse process of encoding. So, the JPEG compression takes about the same time for both encoding and decoding. The encoding/ decoding algorithms provided by an independent JPEG group [7] are available for testing real world images. The information loss occurs only in the process of coefficient quantization. The JPEG standard defines a standard 8×8 quantization table [8] for all images which may not be appropriate. To achieve a better decoding quality of various images with the same compression by using the DCT approach, an adaptive quantization table may be used instead of using the standard quantization table.

3.2 Discrete Wavelet Transform

Wavelets are functions defined over a finite interval and having an average value of zero. The basic idea of the wavelet transform is to represent any arbitrary function (t) as a superposition of a set of such wavelets or basis functions. These basis functions or baby wavelets are obtained from a single prototype wavelet called the mother wavelet, by dilations or contractions (scaling) and translations (shifts). The Discrete Wavelet Transform of a finite length signal x(n) having N components, for example, is expressed by an N x N matrix. Wavelet-based schemes (also referred as sub band coding) outperform other coding schemes like the one based on DCT. Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compression avoid blocking artifacts. Wavelet-based coding [2] is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. In addition, they are better matched to the HVS characteristics [9].

3.3 Fast Fourier Transform

The Fourier transform is a representation of an image as a sum of complex exponentials of varying magnitudes, frequencies, and phases [10]. If $f(m,n)$ is a function of two discrete spatial variables m and n, then the *two-dimensional Fourier transform* of $f(m,n)$ is defined by the relationship

$$F(\omega_1, \omega_2) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) e^{-j(\omega_1 m + \omega_2 n)}$$

The variables ω_1 and ω_2 are frequency variables; their units are radians per sample. $F(\omega_1,\omega_2)$ is often called the *frequency-domain* representation of $f(m,n)$. $F(\omega_1,\omega_2)$ is a complex-valued function that is periodic both in ω_1 and ω_2, with period 2π. Because of the periodicity, usually only the range is displayed. Note that $F(0,0)$ is the sum of all the values of $f(m,n)$. For this reason, $F(0,0)$ is often called the *constant component* or *DC component* of the Fourier transform. (DC stands for direct current; it is an electrical engineering term that refers to a constant-voltage power source, as opposed to a power source whose voltage varies sinusoidally.

The inverse of a transform is an operation that when performed on a transformed image produces the original image. The inverse two-dimensional Fourier transform is given by

$$f(m,n)=\frac{1}{4\pi}\int_{\omega 1=-\pi}^{\pi}\int_{\omega 2=-\pi}^{\pi} F(\omega_1,\omega_2) e^{j(w_1 m+\omega_2 n)} d\omega_1 d\omega_1$$

Roughly speaking, this equation means that $f(m,n)$ can be represented as a sum of an infinite number of complex exponentials (sinusoids) with different frequencies. The magnitude and phase of the contribution at the frequencies (ω_1,ω_2) are given by $F(\omega_1,\omega_2)$.

3.4 Huffman Algorithm

The basic idea in Huffman coding is to assign short codeword to those input blocks with high probabilities and long code words to those with low probability. A code tree is thus generated and the Huffman code is obtained from the labeling of the code tree [11]. An example of how this is done is shown in Table 1.

Table 1: Huffman Source Reductions

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

At the far left, a hypothetical set of the source symbols and their probabilities are ordered from top to bottom in terms of decreasing probability values. To form the first source reductions, the bottom two probabilities, 0.06 and 0.04 are combined to form a "compound symbol" with probability 0.1. This compound symbol and its associated probability are placed in the first source reduction column so that the probabilities of the reduced source are also ordered from the most to the least probable. This process is than repeated until a reduced source with two symbols (at the far right) is reached. The second step if Huffman's procedure is to code each reduced source, starting with the smallest source and working back to its original source. The minimal length binary code for a two-symbol source, of course, is the symbols 0 and 1. As shown in table 2, these symbols are assigned to the two symbols on the right (the assignment is arbitrary; reversing the order of the 0 and would work just and well). As the reduced source symbol with probabilities 0.6 was generated by combining two symbols in the reduced source to its left, the 0 used to code it is now assigned to both of these symbols, and a 0 and 1 are arbitrary appended to each to distinguish them from each other. This operation is then repeated for each reduced source until the original course is reached. The final code appears at the far-left in table 2. The average length of the code is given by the average of the product of probability of the symbol and number of bits used to encode it. This is calculated below: Lavg = (0.4)(1) + (0.3)(2) + (0.1)(3) + (0.1)(4) + (0.06)(5) + (0.04)(5) = 2.2 bits/ symbol and the entropy of the source is 2.14 bits/symbol, the resulting Huffman code efficiency is 2.14/2.2 = 0.973.

Table 2: Huffman Code Assignment Procedure

Original source			Source reduction							
Sym.	Prob.	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

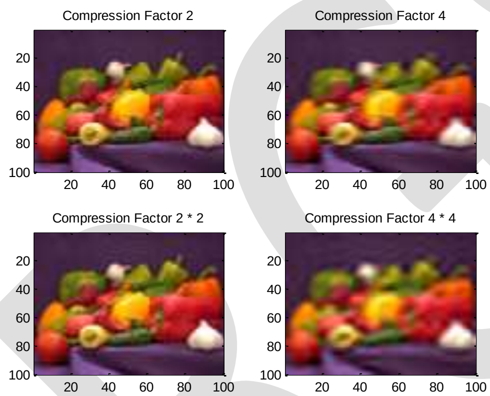
For the binary code of Table 2, a left-to-right scan of the encoded string 010100111100 reveals that the first valid code word is 01010, which is the code for symbol a_3. The next valid code is 011, which corresponds to symbol a1. Continuing in this manner reveals the completely decoded
message to be $a_3a_1a_2a_2a_6$.

Results:

The program to compare all four techniques was designed in MATLAB 7.12. The results obtained are as follows:

1. For Discrete Cosine Transform

Oringinal Image

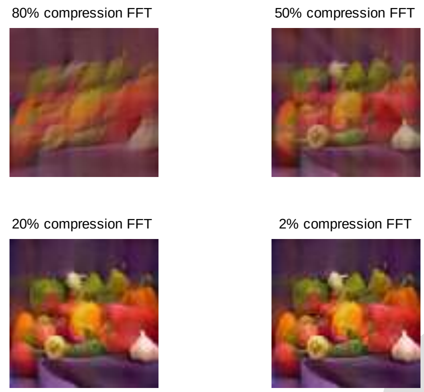


entropy2 = 7.3933
entropy4 = 7.3922
entropy2f = 7.3911
entropy4f = 7.3869
Time required = 1.006370e+001

2. For fast fourier transform

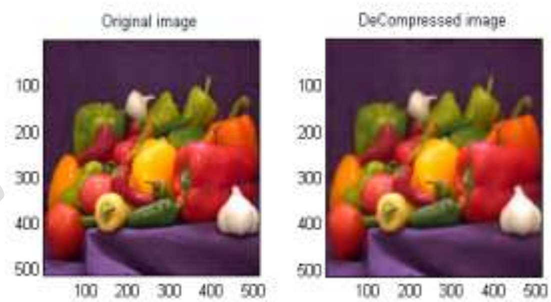
Original Image

International Journal of Engineering Research and General Science Volume 3, Issue 4, July-August, 2015
ISSN 2091-2730



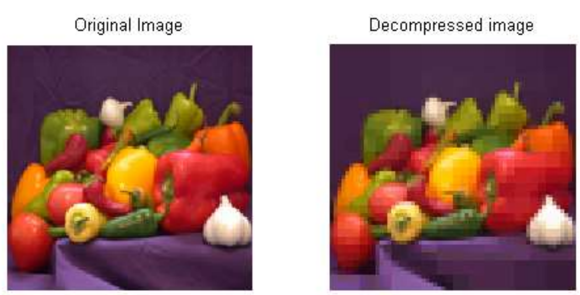
entropy_80_per = 6.9404
entropy_50_per = 7.1618
entropy_20_per = 7.4307
entropy_2_per = 7.4468
Time required = 1.244284e+000

3. For Discrete Wavelet Transform



entropy = 7.3865
Time Required = 3.892191e+000

4. For Huffman Algorithm



entropy = 7.2860

Time required = 1.141801e+002

CONCLUSION

For practical applications, we conclude that (1) Wavelet based compression algorithms are strongly recommended, (2) DCT based approach, (3) Huffman algorithm approach is not appropriate for a low bit rate compression although it is simple, (4) Fourier Transform approach should be utilize for a low bit rate compression.

REFERENCES:

[1] Chan, Y. T., “Wavelet Basics”, Kluwer Academic Publishers, Norwell, MA, 1995

[2] Buccigrossi, R., Simoncelli, E. P., “EPWIC: Embedded Predictive Wavelet Image Coder”.

[3] N.M. Nasrabadi, R.A. King, “Image coding using vector quantization: a review”, IEEE Trans. On Communications,vol. 36, 957-571, 1988.

[4] Ahmed, N., Natarajan, T., Rao, K. R., “Discrete Cosine Transform”, IEEE Trans. Computers, vol. C-23, Jan. 1974, pp. 90-93.

[5] Rao, K. R., Yip, P., “Discrete Cosine Transforms - Algorithms, Advantages, Applications”, Academic Press, 1990

[6] J.M. Shapiro, “Embedded image coding using zero tree of wavelet coefficients”, IEEE Trans. on Signal Processing, vol. 41, 3445-3462, 1993.

[7] Gersho, A., Gray, R.M., “Vector Quantization and Signal Compression”, Kluwer Academic Publishers, 1991.

[8] Malavar, H. S., “Signal Processing with Lapped Transforms”, Norwood, MA, Artech House, 1992.

[9] Mallat, S.G., “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation”, IEEE Trans. PAMI, vol. 11, no. 7, July 1989, pp. 674-693.

[10] Online]Available:ftp.uu.net:/graphics/jpeg/ jpegsrc.v6a.tar.gz

[11] . Said, W.A. Pearlman, “A new, fast, and efficient image codec based on set partitioning in hierarchical trees”, IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, 243-250, 1996

[12] Y. Linde, A. Buzo, R. M Gray, “An algorithm for vector quantizer design”, IEEE Trans. on Communications, vol. 36, 84-95, 1980.