BIOMEDICAL IMAGE COMPRESSION BY DCT

Dimpy Singh[1] , Modassir Anis[2] , Toshi Patel[3], Amrita Kaur[4]

[1,2,3,4]*Software Engineering, SRMSCET, Bareilly, (India)*

ABSTRACT

Large amount of medical images are assimilated, in biomedical field. (e.g. vertebra, lung digital X rays etc.). The storage and transmission is an important dilemma due to enormous size of medical image data. For example, each slice of CT abdomen images is 512 by 512 of 16 bits, and the data set consists of 200 to 400 images leading to 150 MB of data in average. An efficient compression of the medical data can solve the storage and transmission problem. It is very difficult to store such type of large images and transmit them. Thus, image compression is widely applied to biomedical images.Image compression is a method which reduces the size of the data to reduce the amount of the space required to store the data and decreases the transmission time. In this paper, we use a Discrete Cosine Transform (DCT) for compression of biomedical images in JPEG and PNG format and compare the JPEG and PNG images on the basis of results obtained by DCT in a MATLAB platform.

Keywords**: DCT (Discrete Cosine Transform), JPEG (Joint Photographic Experts Group), PNG (Portable Network Graphics), Biomedical image, Image compression.***

I. INTRODUCTION

Medical image compression plays a key role as hospitals move towards filmless imaging and go completely digital. Image compression will allow Picture Archiving and Communication Systems (PACS) to reduce the file sizes on their storage requirements while maintaining relevant diagnostic information. Even as the capacity of storage media continues to increase, it is expected that the volume of uncompressed data produced by hospitals will exceed capacity and drive up costs [2].

Medical imaging has a great impact on diagnosis of diseases and preparation to surgery. On the other hand, the storage and transmission is an important dilemma due to enormous size of medical image data. For example, each slice of CT abdomen images is 512 by 512 of 16 bits, and the data set consists of 200 to 400 images leading to 150 MB of data in average. An efficient compression of the medical data can solve the storage and transmission problem [7].

The purpose for image compression is to reduce the amount of data required for representing sampled digital images and therefore reduce the cost for storage and transmission. Image compression plays a key role in many important applications, Including biomedical images, image database, image communications, and remote sensing. The image(s) to be compressed are grey scale with pixel values between 0 to 255. There are different techniques for compressing images. They are broadly classified into two classes called lossless and lossy compression techniques.

As the name suggests in lossless compression techniques, no information regarding the image is lost. In other words, the reconstructed image from the compressed image is identical to the original image in every sense.

Whereas in lossy compression, some image information is lost, i.e. the reconstructed image from the compressed image is similar to the original image but not identical to it. In this work we will use a Discrete Cosine Transformation for biomedical image compression [3].

The main objective of this paper is to compare the compressed JPEG (Joint Photographic Experts Group**)** and PNG (Portable Network Graphics**)** biomedical images through MATLAB using DCT. The basic objective of image compression is to find an image representation in which pixels are less correlated.

The entire paper is organized in the following sequence. In section -2 compression techniques are stated. In section-3 DCT (Discrete Cosine Transformation) and Wavelet transformation are explained. Section-4 deals with Image file formats. Section 5 covers the implementation of DCT. section-6 Shows the Result. Lastly, in Section-7 ends with conclusion and future work .

II. DIFFERENT COMPRESSION TECHNIQUES

Compression can be divided into two categories, as Lossless and Lossy compression. In lossless compression, the reconstructed image after compression is numerically identical to the original image [3]. In lossy compression scheme, the reconstructed image contains degradation relative to the original. Lossy technique causes image quality degradation in each compression or decompression step. In general, lossy techniques provide for greater compression ratios than lossless techniques.

The following are the some of the lossless and lossy data Compression techniques:

2.1 Lossless Coding Techniques

As the name suggests in lossless compression techniques, no information regarding the image is lost. In other words, the reconstructed image from the compressed image is identical to the original image in every sense.

2.1.1. Run Length Encoding

This, one of the simplest of the techniques in use, can be used to compress any kind of data. However, the compression attained is dependent on the content type. It has become very popular as it is easy to implement and provides a quick method of compressing data [4].

It works by reducing repeating strings of characters into runs of, typically, two bytes (although the atomic RLE base can also be bit- or pixel-based). The first byte represents the number of characters in the run and is called the run count. The second byte is the value of the encoded character string and is called the run value [4].

2.1.2. Huffman Encoding

This algorithm produces variable-length codes according to a symbol’s probability within a stream. These codes can then replace the symbols in the compressed stream, thus producing compression. The two important things to note about this are:

- Shorter bit-codes represent the symbols most likely to occur. Thus a 1-bit code can be assigned to the most probable symbol in the stream and the largest amount of bits per symbol represents the least likely.
- The codes have a unique prefix attribute, which allows variable-length codes to be identified and decoded even though they are not uniform [4].

2.1.3. Arithmetic Encoding

Arithmetic Coding has the capability of optimal storage for any string through the use of arithmetic. More specifically, a string is represented as a floating point number between 0 and 1. This number can be uniquely decoded to give the stream of symbols that it is constructed from.

To do this we must first calculate the probability of each of the symbols contained in the stream. With these values we need to assign a range within 0-1 for each of the symbols according to their likelihood [4].

2.1.4. Entropy Encoding

Three sub-stages complete the JPEG encoding process. Firstly, the coefficients in the top left of each block are converted from absolute to relative values. As the differences from block to block are likely to be small, using relative values allows smaller values to be stored. Next, the zero and non-zero values are treated separately. The zeroed values can be converted to run-length encoding pairs reducing the storage requirements drastically. Due to the nature of the frequency distribution (most important, or low, in the top left corner) the run-length encoding does not take the usual path across and down an image. Instead, a zigzag sequence is used which exploits the features of the frequency coefficient distribution [4].

2.2 Lossy Coding Techniques

In lossy compression, some image information is lost, i.e. the reconstructed image from the compressed image is similar to the original image but not identical to it.

2.2.1 Predictive Coding

The main component of the predictive coding method is the "Predictor" which exists in both encoder and decoder. The encoder computes the predicted color value for a pixel, denote $f'(n)$ based on the known pixel color values of its neighbouring pixels. The residual error, which is the difference value between the actual color value of the current pixel $f(n)$ and the predicted one, i.e. $e(n) = f(n)-f'(n)$ is computed for all pixels. The residual errors are then encoded, usually by an encoding scheme like Huffman encoding, to generate a compressed data stream. The decoder also computes the predicted color value of the current pixel $f'(n)$ based on the previously decoded color values of neighbouring pixels using the same method as the encoder. The decoder decodes the residual error $e(n)$ for the current pixel and performs the inverse operation $f(n) = e(n)+ f'(n)$ to restore the color value of the current pixel [5].

2.2.2 Transform Coding (FT/DCT/Wavelets)

Transform coding compresses image data by representing the original signal with a small number of transform coefficients. It exploits the fact that for typical images a large amount of signal energy is concentrated in a small number of coefficients. Transform coding is an integral part of the Joint Photographic Experts Group (JPEG) standard for lossy image compression.

Transform coding is a type of data compression for "natural" data like audio signals or photographic images. The transformation is typically lossy, resulting in a lower quality copy of the original input.

In transform coding, knowledge of the application is used to choose information to discard, thereby lowering its bandwidth. The remaining information can then be compressed via a variety of methods. When the output is decoded, the result may not be identical to the original input, but is expected to be close enough for the purpose of the application.

Block transform coding divides an image into blocks of equal size and processes each block independently. Block processing allows the coder to adapt to local image statistics, exploit the correlation present among neighbouring image pixels, and reduce computational and storage requirements. The baseline JPEG algorithm uses non overlapping blocks of dimensions 8x8.

III. TRANSFORMATION

DCT (Discrete Cosine Transform) and Wavelet Transformation are lossy coding techniques for image compression.

3.1. DCT (Discrete Cosine Transform)

Discrete Cosine Transform (DCT) is a technique for converting a signal into elementary frequency components. Like other transforms, Discrete Cosine Transform (DCT) attempts to de correlate the image data. After de correlation each transform coefficient can been coded independently without losing compression efficiency [1].

3.2. Wavelet Transformation

The Wavelet Transform (WT) is a way to represent a signal in time-frequency form. Wavelet transform are based on small waves, called wavelets, of varying frequency and limited duration Wavelet Transform uses multiple resolutions where different frequencies are analysed with different resolutions. This provides a more detailed picture of the signal being analysed. [1]

A transform can be thought of as a remapping of a signal that provides more information than the original. The Fourier transform fits this definition quite well because the frequency information it provides often leads to new insights about the original signal. However, the inability of the Fourier transform to describe both time and frequency characteristics of the waveform led to a number of different approaches described in the last chapter.

None of these approaches was able to completely solve the time–frequency problem. The wavelet transform can be used as yet another way to describe the properties of a waveform that changes over time, but in this case the waveform is divided not into sections of time, but segments of scale.

In the Fourier transform, the waveform was compared to a sine function in fact, a whole family of sine functions at harmonically related frequencies.

This comparison was carried out by multiplying the waveform with the sinusoidal functions, then averaging (using either integration in the continuous domain, or summation in the discrete domain)[1].

IV. IMAGE FILE FORMATS

The most common image file formats, the most important for cameras, printing, scanning, and internet use, are JPG, TIF, PNG, and GIF.

4.1 JPEG files

Digital cameras and web pages normally use JPEG files because JPEG heroically compresses the data to be very much smaller in the file. However JPEG uses lossy compression to accomplish this feat, which is a strong downside. A smaller file, yes, there is nothing like JPEG for small, but this is at the cost of image quality. This degree is selectable (with an option setting named JPEG Quality), to be lower quality smaller files, or to be higher quality larger files. In general today, JPEG is rather unique in this regard, using lossy compression allowing very small files of lower quality, whereas almost any other file type is lossless (and larger). Frankly, JPEG is used when small file size is more important than maximum image quality (web pages, email, memory cards, etc.). But JPEG is good enough in many cases, if we don't overdo the compression. Perhaps good enough for some uses even if we do overdo it (web pages, etc.). But if you are concerned with maximum quality for archiving your important images, then you do need to know two things: 1) JPEG should always choose higher Quality and a larger file, and 2) do not keep editing and saving your JPEG images repeatedly, because more quality is lost every time you save it as JPEG (in the form of added JPEG artifacts. pixels become colors they ought not to be - lossy) [6].

4.2 TIF files

TIF is lossless (including LZW compression option), which is considered the highest quality format for commercial work. The TIF format is not necessarily any "higher quality" per se (the image pixels are what they are), and most formats other than JPG are lossless too. This simply means there are no additional losses or JPG artifacts to degrade and detract from the original. And TIF is the most versatile, except that web pages don't show TIF files. For other purposes however, TIF does most of anything you might want, from 1-bit to 48-bit color, RGB, CMYK, LAB, or Indexed color. Most any of the "special" file types (for example, camera RAW files, fax files, or multipage documents) are based on TIF format, but with unique proprietary data tags - making these incompatible unless expected by their special software [6].

4.3 GIF files

GIF was designed by CompuServe in the early days of computer 8-bit video, before JPG, for video display at dial up modem speeds. GIF always uses lossless LZW compression, but it is always an indexed color file (8-bits, 256 colors maximum), which is poor for 24-bit color photos. Don't use indexed color for color photos today, the color is too limited. PNG and TIF files can also optionally handle the same indexed color mode that GIF uses, but they are more versatile with other choices too. But GIF is still very good for web graphics (i.e., with a limited number of colors). For graphics of only a few colors, GIF can be much smaller than JPG, with more clear pure colors than JPG). Indexed Color is described at Color Palettes (second page of GIF link below) [6].

4.4 PNG files

Digital PNG can replace GIF today (web browsers show both), and PNG also offers many options of TIF too (indexed or RGB, 1 to 48-bits, etc.). PNG was invented more recently than the others, designed to bypass possible LZW compression patent issues with GIF, and since it was more modern, it offers other options too (RGB color modes, 16 bits, etc). One additional feature of PNG is transparency for 24 bit RGB images. Normally PNG files are a little smaller than LZW compression in TIF or GIF (all of these use lossless compression, of different types), but PNG is perhaps slightly slower to read or write. That patent situation has gone away now, but PNG remains excellent. Less used than TIF or JPG, but PNG is another good choice for lossless quality work [6].

V. DCT ALGORITHM USED

5.1 Used DCT Algorithm

a. The following is the general overview of JPEG and PNG format.

b. The image is broken into 8×8 blocks of pixels.

c. Working from left to right, top to bottom, the DCT is applied to each block.

d. Each block is compressed through quantization.

e. The array of compressed blocks that constitute the image is stored in a drastically reduced amount of space [1].

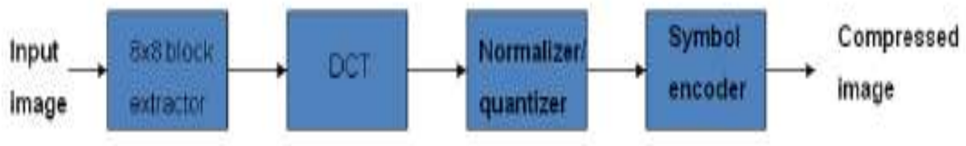

Fig. 1 Image Compression using DCT

VI. RESULT

Comparison of compressed JPEG and PNG biomedical images

6.1 JPEG

6.1.1 Original Image

Fig. 2 Original JPEG image

6.1.2 Compressed Image

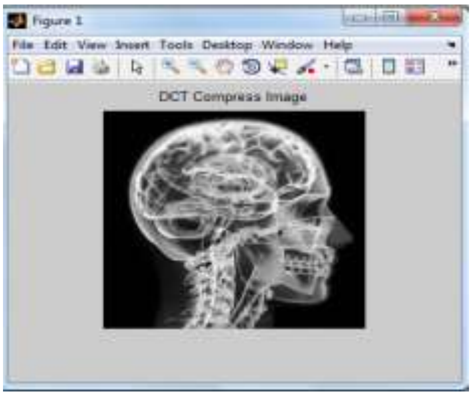

Fig. 1 Compressed JPEG image.

6.1.3 Histogram of ORIGINAL IMAGE

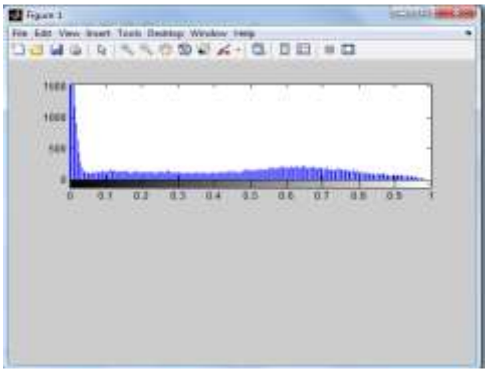

Fig. 4 Histogram of original JPEG image

6.1.4 Histogram of Compressed Image

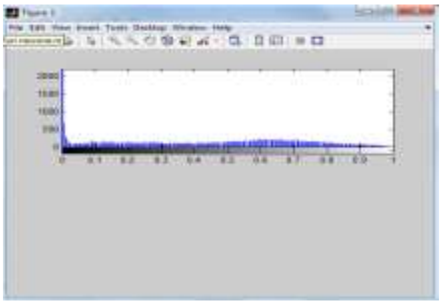

Fig. 5 Histogram of Compressed JPEG image.

6.2 PNG

6.2.1 Original Image

Fig. 6 Original PNG image

6.2.2 Compressed Image

Fig. 7 Compressed PNG image

6.2.3 Histogram of Original PNG Image

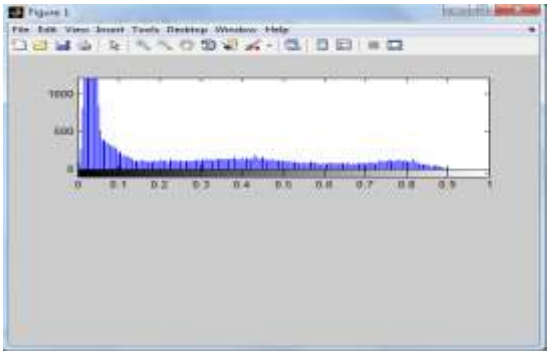

Fig. 8 Histogram of Original PNG image

6.2.4 Histogram of Compressed Image

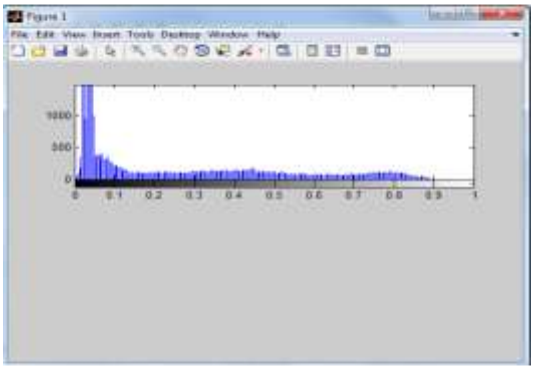

Fig. 9 Histogram of compressed PNG image

VI. CONCLUSION AND FUTURE WORK

On the basis of Compressed JPEG and PNG images and histogram of original and compressed JPEG and PNG images as shown above we conclude that JPEG image format is better than PNG image format for compression of biomedical images. As the future work for compression of biomedical images we can use wavelet transformation compression technique.

REFERENCES

[1] Prabhakar Telagarapu, V.Jagan Naveen, A. Lakshmi. Prasanthi, et al "Image Compression using DCT and Wavelet transformation", vol. 4, no. 3, September 2011

[2] [2]. Matthew J. Zukoski, Terrance Bolt, Tunc Lyriboz, "A novel approach to medical image compression", vol. 2, no. 1, 2006.

[3] Jagdish H. Pujar, Lohit M. Kadlaskar, "A new Lossless method of Image Compression and Decompression using Huffman coding techniques",2010 .

[4] M.G. Ross, project report on "measuring image Quality", april 1996.

[5] Fuangfar Pensiri, Surapong Auwatonamongkol, "A lossless image compression algorithm using predictive coding based on quantized colors.

[6] http ://www.scantips.com/basics09.html#summary

[7] robotics.stanford.edu/~gokturkb/Compression/Finalreport.htm