COMPRESSION OF BIOMEDICAL IMAGES USING DCT

Mohd. Aurangzeb Atiq[1], Priyanka Pradhan[2], Vipul Rastogi[3], L.S Maurya[4]

[1,2,3] M. Tech (SE), [4]HOD CS/IT, SRMSCET, (India)

ABSTRACT

In the biomedical field, large amount of medical images are acquired (e.g. vertebra, lung digital X rays etc.). The storage and transmission is an important dilemma due to the 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 250 to 400 images leading to 150 MB of data on 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 the 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: Biomedical image, DCT (Discrete Cosine Transform), Image compression, JPEG (Joint Photographic Experts Group), PNG (Portable Network Graphics

I. INTRODUCTION

Medical image compression plays a key role as hospitals move towards filmless imaging and go digital. Image compression provides Picture Archiving and Communication Systems to reduce the sizes on their storage requirements while maintaining relevant diagnostic data. Even as the capacity of storage media continues to increase, it is expected that the size 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 data storage and transmission is an important dilemma due to the enormous size of medical image. In general, each slice of C.T. abdomen images is 512 X 512 of 16 bits, and the data set consists of 250 to 400 images leading to 150 MB of data in average. An efficient compression of the image can solve the storage and transmission problem [7].

The purpose of image compression is to decrease the amount of data required for representing digital images and therefore reduce the cost of storage space and transmission. Image compression is very important in many important applications, including biomedical images, image database, image communications, and remote sensing. The images to be compressed are gray scale with pixel values between 0 to 255. There are many different techniques for image compression. They are mainly classified into two classes called lossless and lossy compression techniques .Firstly lossless compression techniques, no information is lost regarding the image, the reconstructed image from the compressed image is identical to the original image. Secondly lossy compression, some information of the image is lost, i.e. the image reconstructed from the compressed image is visually similar to the original image but not identical to the original image. 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 research methodology is described section -3 compression techniques are stated, section-4 image formats are explained, section-4 DCT (Discrete Cosine Transformation) and Wavelet transformation are explained, In section-5 Image file formats are explained, section 6- the implementation of DCT, section-7 Result, section-8 Lastly, the paper ends with the conclusion, future work and References

II. RESEARCH METHODOLOGY

Research is a systematic and stepwise process that starts with identifying a problem or an area of research and then follows the steps of the our research are data collection, and hypothesis testing and results analysis.

2.1 Data Collection: Data the collected through the various websites like Google and MSN search engine.

2.2 Tool Selection: In this paper MATLAB is used for performing the operations on images.

2.3 Performing the Procedure: Through DCT algorithm we compress the images and plot the histogram of the images by using the MATLAB.

2.4 Result Analysis: Analysis the JPEG and PNG compress image through histogram.

III. COMPRESSION TECHNIQUES

Compression can be broadly divided into two classes, as Lossless technique and Lossy technique .Begins with lossless compression, in the 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 decreases the quality of images in each compression or decompression step. In general, lossy Techniques is more suitable because it provide greater compression ratios than lossless techniques. There are some of the lossless and lossy data Compression techniques are as follows:

3.1 Lossless Coding Techniques

As the name suggests in lossless compression techniques, there is no loss of information regarding the image. That means, the reconstructed image from the compressed image is identical to the original image in every sense.

3.1.1 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].

3.1.2 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:

- Likely to be small, using relative values allows smaller values to be stored. Next, the zero and non-zero values are treated 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].

3.1.3 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].

3.1.4 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 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].

3.2 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.

3.2.1 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 colour value for a pixel, denote $f'(n)$ based on the known pixel colour values of its neighbouring pixels. The residual error, which is the difference value between the actual colour 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 colour value of the current pixel $f'(n)$ based on the previously decoded colour 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 colour value of the current pixel [5].

3.2.2 Transform Coding (FT/DCT/Wavelets)

Transform coding compresses image data by representing the original signal with a small number of transform coefficients. Transform coding exploits the fact that for typical images a large amount of signal energy is concentrated in a small number of coefficients. Transform coding is vital 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. The transformation is typically lossy in which data must loss, resulting in a degrade quality copy of the original input.

In transform coding, knowledge of the application is used to choose irrelevant information and then discard that information, thereby decreasing its bandwidth. The important information can then be compressed via different methods. When the output is decoded, the result may not be identical to the original input, but is expected (visually) 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 individually. Block processing aids the coder to adapt to local image statistics, exploit the correlation present among neighbouring image pixels, and decrease the computational and storage space requirements. The baseline JPEG algorithm uses non overlapping blocks of dimensions 8x8.

IV. DCT AND WAVELET TRANSFORMATION

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

4.1 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].

4.2 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 different frequency and limited time period Wavelet Transform uses multiple resolutions where different frequencies are analysed with varies of 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 as compare to original. Generally, the Fourier transformation fits in 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 section. 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, comparison between waveform and 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].

V. IMAGE FILE FORMATS

Forty eight-bit colour, RGB, CMYK, LAB, or Indexed colour. Most of the "special" file types (for example, camera RAW files, fax files, or multipage documents) are based on The worldwide common used image file formats, the most important for cameras, scanning, printing, and internet use, are JPG, TIF, PNG, and GIF.

5.1 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 task, which is a concretely downside. A smaller file, yes, there is nothing like JPEG for small, but this is at the cost of quality of image. This degree is selectable (with an option setting named JPEG Quality), to be quality of smaller files are lower, or to be quality of larger files are higher. In general today, JPEG is rather unique in this way, allowing very small files of lower quality by using lossy compression, whereas almost or every 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 situations, if we don't do the compression over and over. 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 should need to know 2 things: i) JPEG should always choose a larger file and higher Quality and ii) do not keep editing and saving your JPEG images at regular interval or repeatedly, because quality of image is lost every time when you save it as JPEG (in the form of added JPEG artifacts. pixels become colours they ought not to be - lossy) [6].

5.2 TIF Files

TIF is lossless (including LZW compression option), which is considered the highest format of quality for commercial work. The format of TIF is not mandatorily any "higher quality" and most formats other than JPG are also lossless. This simply means there are no such additional losses or JPG artifacts to degrade and detract from the original. And TIF is the most diverse and versatile, excluding that web pages don't show TIF files. For other purposes generally, TIF does most of thing that you might want, from one-bit to TIF format, but with unique proprietary data tags - making these incompatible unless expected by their special software [6].

5.3 GIF Files

GIF was designed by CompuServe in the early days of computer video of 8-bit, before JPG, for display of video at dial up modem speeds. Lossless LZW compression is always used by GIF, but it is always an indexed colour file (8-bits, 256 colours maximum), which is relatively poor quality for 24-bit colour image. Don't use indexed colour for colour images, the colour is limited. PNG and TIF files can also optionally handle the same indexed colour mode that GIF uses, but they are more diverse and versatile with other alternatives. But GIF is still very good for web (i.e., with a small number of colours). For graphics of only a limited colours, GIF can be relatively smaller than JPG, with more clear view of colours than JPG) [6].

5.4 PNG Files

Digital PNG can replace GIF today, and PNG also offers many options of TIF (indexed or RGB, one to forty eight bits, etc.). PNG was discovered more recently than the others formats, designed to alternate possible LZW compression patent regarding issues with GIF format, and since it was much modern, it offers other way too (RGB colour , 16 bits, etc). One important feature of PNG is transparency for 24 bit RGB images. Generally

PNG files are little smaller than LZW compression in TIF (all of these use lossless compression), but PNG may a little slower to read or write. That patent issue has overcame now, but PNG remains excellent. PNG is much less used than TIF or JPG, but PNG is next better way for lossless quality work [6].

VI. IMPLEMENTATION OF DCT

6.1 PROPOSED 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

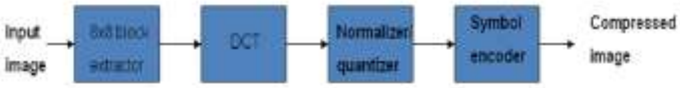

Fig. 1 Image Compression Using DCT

VII. RESULT

a. Comparison of compressed JPEG and PNG biomedical images

7.1 Jpeg

7.1.1 Original Image

Fig. 1 Original Jpeg Image

7.1.2 Compressed Image

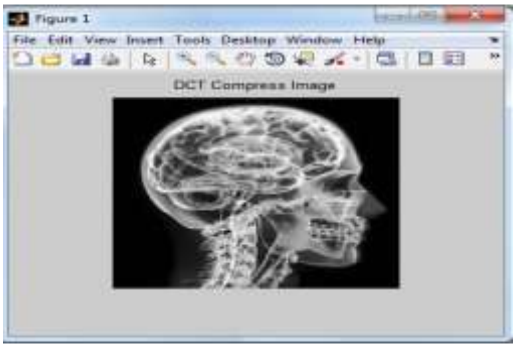

Fig. 2 Compressed Jpeg Image

7.1.3 Histogram of Original Image

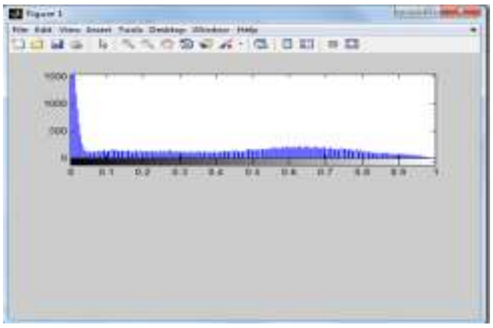

Fig. 3. Histogram of Original Jpeg Image

7.1.4 Histogram of Compressed Image

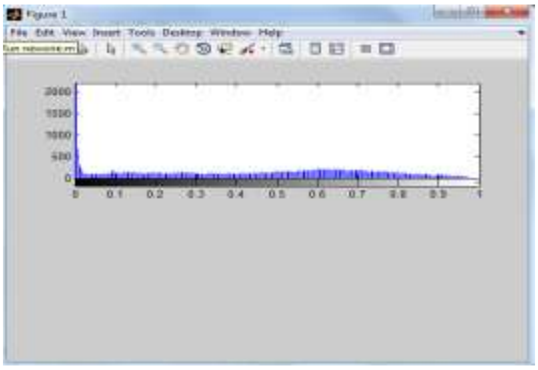

Fig. 4 Histogram of Compressed JPEG Image

7.2 Png

7.2.1 Original Image

Fig. 5 Original PNG Image

7.2.2 Compressed Image

Fig. 6 Compressed PNG Image

7.2.3 Histogram of Original PNG Image

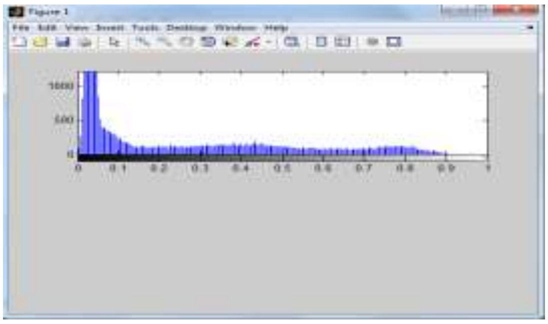

Fig. 7 Histogram of Original PNG Image

7.2.4 Histogram of Compressed Image

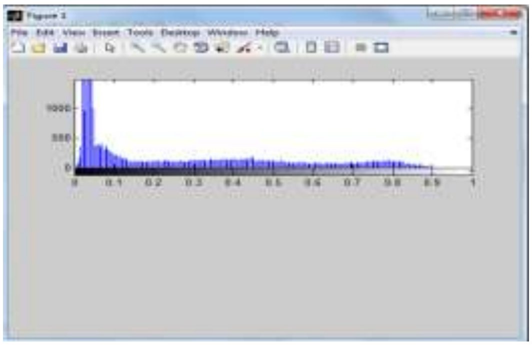

Fig. 8 Histogram of Compressed PNG Image

VIII. 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 colours.

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

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