A Review of Image Compression Techniques

ARWINDER SINGH AND G.ANAND
Faculty of Engineering and Technology,
INTI International University College,
Persiaran Perdana BBN, Putra Nilai
71800 Nilai, Negeri Sembilan, Malaysia

Abstract
In this modern era of communication, where time is of the essence, the development of better image compression and transmission techniques has been a topic of much interest.

In this paper, we will only investigate and explain the basic image-processing techniques, which are the fast Fourier transform, discrete cosine transform, and discrete wavelet transform. With the basic understanding the methodology used in JPEG and JPEG 2000 compression will be simulated and its results explained.

The aim of this paper is to provide the reader with a basic understanding of the various image compression schemes available before they are actually used with different modulation schemes to simulate performances in wireless transmission mediums.

Introduction
Images are signals with special characteristics. They are measures of parameters over space (distance), while most signals are measures of parameters over time. It is known that digital-imaging applications such as multimedia, internet publishing, and teleconferencing have grown significantly over the past decade. Many people now prefer to complement their messages with photos, images, and/or animations to better express themselves and to make it more attractive for other people to see. Some things simply cannot be described by words, but "A picture is worth a thousand words." These phenomena are found mostly in communication areas, such as instant-messaging, e-mails, and even cell phones. Also, in the corporate world, there are corporations that engage in online catalogs, which employ plentiful images and animations in their websites.

A digital image consists of a grid of dots, more commonly known as pixel (picture element). The more pixels an image has the better the resolution, hence the better its quality. A pixel consists of three primary color components--Red, Green and Blue (RGB)--which are associated with integer numbers that define the number of bits in a single pixel. Any other color can be represented by the mixture of these RGB colors. A typical color resolution for a solid black and white unrealistic image is one bit per pixel (bps); eight bps for grayscale images, unrealistic color images, and coarse realistic images; 24 bps for photographic quality realistic images; and 48 bps for ultrahigh quality images. The size of an image stored is calculated by the number of pixels (width x height) times the color bits per pixel. For example, an uncompressed 640 x 320 pixels image with eight bits will be equivalent to 640 x 320 x 8 bits = 1,638,400 bits or equal to 204,800 byte (note: 1 byte = 8 bits).

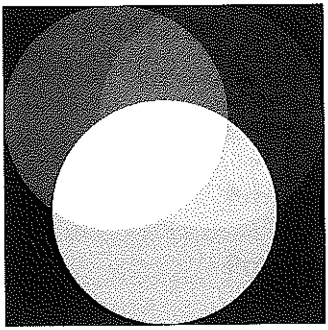


Figure 1: Primary colors of RGB (Red, Green and Blue) and their combinations

Images can generally be classified as "realistic" or "unrealistic". Realistic images include photographs and realistic artworks with many shades and few straight lines. Unrealistic images are those with block colors and lots of straight lines, for example graphics, block diagrams, graphic banners and cartoons.

Two types of image-compression methods available are lossless and lossy compression. As the names imply, the lossless compression method produces exactly the same image after decompression, while the lossy compression method brings about degraded image results after decompression, depending on the degree of compression--the more it is being compressed, the more degraded the image will be after decompression process. The lossless compression method is a necessity for image files in which high resolution imagery is required and larger file-size is not an issue, as with, for example, texts, engineering graphs and diagrams, or medical images. The lossy compression method is employed mostly for realistic images/artworks and photographic images where some of the picture elements are eliminated. For these kinds of images, the missing elements of an image will not be immediately obvious to the human eye.

Literature Review

To compress an image, the image is first divided into blocks. This method is called block encoding (BE).Once the image being broken down into blocks, vector quantization (VQ) can be done. This is just a quantization technique in which the input data is arranged to achieve maximum intra-vector and minimum inter-vector correlations, so that the compression ratio of VQ can be increased.

Another method of block encoding is called block truncation coding (BTC). This method divides the original images into small sub-images and then uses a quantizer, which adapts itself according to the image statistics to reduce the number of gray scale levels in the image. With this method the compression ratio is limited.

If the output of the block encoder is further transformed using the Fourier, cosine or wavelet transforms, then better compression ratios can be achieved. The transforms are described below.

Fast Fourier Transform

The fundamental principle in processing an image using Fourier [3] analysis is to manipulate the spectrum of an image by letting some of its elements equal zero, to get rid of unwanted frequencies (the frequencies that have zero or almost zero magnitude in the spectrum are neglected), and to then transform back the image using the inverse Fourier transform. Thus only a small fraction of the spectrum is needed to represent an image, with some tolerable losses.

Discrete Cosine Transform

The Discrete Cosine Transform (DCT) [6] is very similar to the discrete Fourier transform, as both transform an image from the spatial domain to the frequency domain. However, when using DCT, in most images, much of the signal energy lies at low frequencies, which appear in the upper left corner of the transform. Thus the higher frequencies can be neglected with little visible distortion.

Discrete Wavelet Transform

The Discrete Wavelet Transform (DWT) [9] decomposes an image into a set of wavelet basis functions that are localized, both in time and frequency. Therefore, signals can be reconstructed from a much smaller set of wavelet basic functions. This is a major advantage, as image is not seriously degraded.

Joint Photographic Expert Group (JPEG)

JPEG [8] is the first international standard for still-image compression where the encoders and decoders are DCT-based.

The DCT-based encoder is simply a compression of a stream of 8x8 blocks of image samples. Each 8x8 block makes its way through each processing step and yields an output in a compressed form, onto the data stream. Each of the 64 DCT coefficients is then uniformly divided with a quantization weightage table.

After been divided by the weightage table, all the rounded coefficients are ordered into the "zig-zag" sequence. This ordering helps to facilitate entropy encoding by placing low-frequency non-zero coefficients before high-frequency coefficients. The DC coefficient, which contains a significant fraction of the total image energy, is differentially encoded.

Entropy Coding (EC) achieves additional compression losslessly by encoding the quantized DCT coefficients more compactly, based on their statistical characteristics. The JPEG proposal specifies both Huffman coding and arithmetic coding. But in this, the Matlab simulation will only be implemented using Huffman coding [7] and Run Length encoding[5] .A reverse process is then done to reconstruct the image.

JPEG2000

JPEG2000 [2] applies a form of transform coding to compress images using the wavelet transform. How JPEG 2000 works can be explained as follows. Initially, images have to be transformed from the RGB (red, green, blue) color space to the YUV color space (the Y component represents brightness of a pixel, and the U and V components together represent the hue and saturation)

After color transformation, the image is split into tiles which are rectangular regions of the images that are transformed and encoded separately. These tiles are then wavelet-transformed to an arbitrary depth. This results in a collection of sub-bands which represent several approximation scales. A sub-band is a set of coefficients which represent aspects of the image associated with a certain frequency range as well as a spatial area of the image. These coefficients are then quantized using scalar-quantization, giving rise to a set of integer numbers which need to be encoded bit-by-bit.

The quantized sub-bands are then further split into precincts, which are rectangular regions in the wavelet domain. These precincts are then further split into code-blocks. Code-blocks are located in a single sub-band and have equal sizes--except for those located at the edges of the image. The encoder has to encode the bits of all quantized coefficients of a code-block, starting with the most significant bits and progressing to less significant bits by a process called the Embedded Bitplane Coding by Truncation (EBCOT) scheme.

During the encoding process, each bit-plane of the codeblock gets encoded in three coding passes, which are called significance propagation pass; magnitude refinement pass; and cleanup pass [2].

The bits selected by these coding passes then get encoded by a binary MQ-coder. The context of a coefficient is formed by the state of its nine neighbors in the codeblock. The result is a bit-stream that is split into packets where a packet group selected passes through all codeblocks from a precinct into one indivisible unit.
Packets from all sub-bands are then collected in layers. These layers will then define the progression by image quality within its codestream.

Methodology
The study on image compression using various techniques has been split into two main parts.

In the first part, for both FFT and DCT, the basic binary image (Lena.mat 256x256) was split into the basic 8x8 blocks and then subjected to either an "fft2" or a "dct2" MATLAB command. The output of this is then multiplied with a respective mask In the masking process the low frequency energy component of an image can be kept by multiplying it with "one" and high frequency components removed by multiplying it with "zero" in which the number of "ones" to "zero" represents the compression ratio. The image is then reconstructed back using the appropriate "ifft2" or "idct2" respectively, as shown in the flow chart of Figure 3.1.

The compression ratio using masking can be calculated using the total number of zeros divided by 64, which represents the 8x8 block matrix.

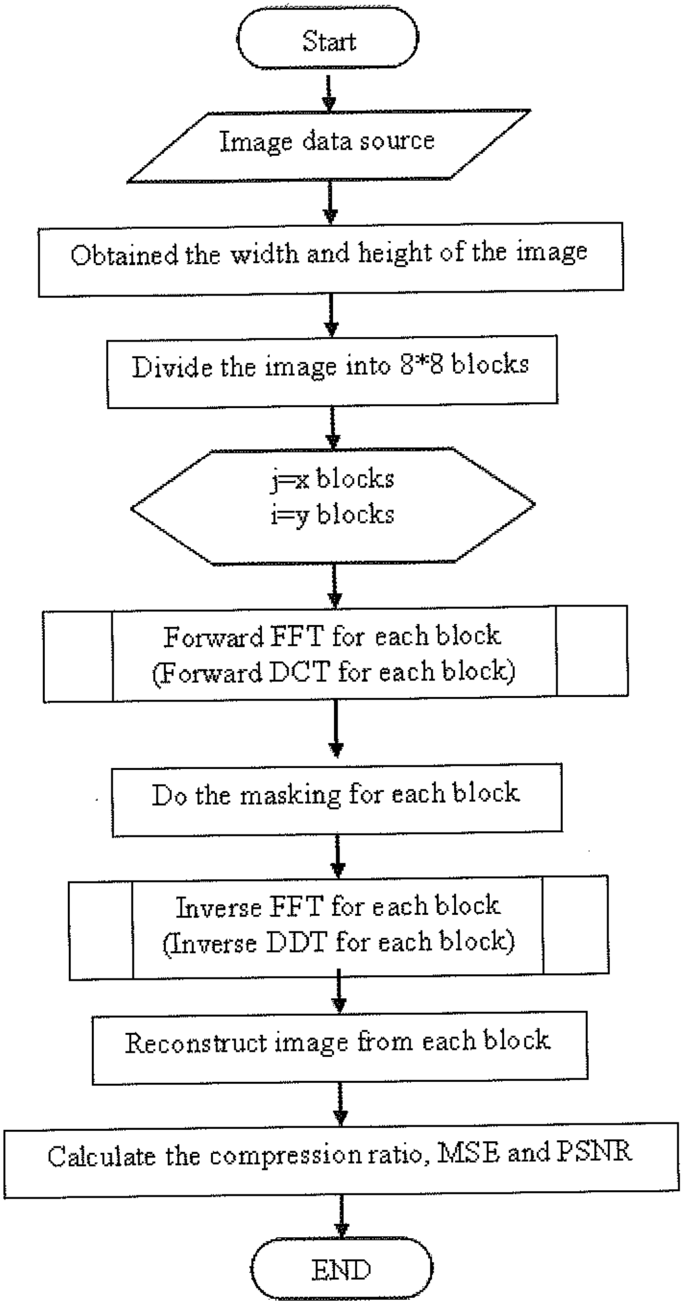
For example, for the block below

```
mask = [1 1 1 1 1 1 0 0
        1 1 1 1 1 0 0 0
        1 1 1 1 0 0 0 0
        1 1 1 0 0 0 0 0
        1 1 0 0 0 0 0 0
        1 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0 0];
```

thus the compression ratio is (43x100%)/64=67.1875%.

For wavelet transform, the matlab function "wavedec2" is used to return the wavelet decomposition of the basic binary image (Lena.mat [256x256]). Function "wdcbm2" returns level-dependent threshold THR and numbers of coefficients to be kept, for compression. Finally, by executing the function "wdencmp", we obtained a compressed version of the input image obtained by providing wavelet coefficients thresholding using global positive threshold THR.

Figure 3.1: Flowchart for image compression using FFT and DCT.



The above steps are carried out purposely, since frequencies (coefficients) that are most prominent in the original image appear as high amplitudes in that region of the wavelet decomposition image, which includes those particular frequencies. The frequencies (coefficients) that are not very prominent in the original image will have very low amplitudes, and that part of the wavelet decomposition image can be discarded using threshold without any major loss of information, allowing image compression.

Thus the wavelet image compression is achieved when the number of coefficients after thresholding to represent the image is fewer compared to the initial number of coefficients in the original image.

In the second part, both JPEG and "JPEG2000 like compression" have been simulated. In JPEG compression, the binary "Lena image" is split into blocks and a suitable JPEG quality factor was chosen (quality factor used in this paper is 50). Since luminance plays a very important part in image compression the image was subjected to a standard JPEG luminance weight age quantization table as recommended by the CCITT or International Telecommunication Union (ITU) in which each frequency component was divided by a separate quantization coefficient which determines the extent to which a coefficient is weighted. The image is next transformed using discrete cosine transformation, quantise and then zigzag to convert the matrix to a column. It was then subjected to Huffman coding and run length encoding and the number of bits for the compression was calculated. The image was then reconstructed back using de zigzag, dequantization and inverse discrete cosine transformation.

In "JPEG2000 like compression", the binary image is split into 8x8 tile size (same block size as JPEG so that a fair evaluation between JPEG and JPEG2000 compression ratios can be made).

A forward JPEG2000 transformation is carried out on all tiles. In this forward transformation discrete wavelet transform is performed, after which each block is quantized. The difference between the real JPEG2000 and our "JPEG2000 like compression" is that instead of using a complicated context based coder (MQ-coder), we use zero-run length encoder (this function returns zero and non-zero counts in blocks multiple of) and Huffman compression[1]. The tiles were then reconstructed using backward JPEG2000 in which dequantization and backward discrete wavelet transform was done. The compressed image was then reconstructed by rearranging the tiles.

ANALYSIS OF DATA

Image Compression

In this paper we have used a gray scale Lena (256x256).mat image which has an intensity range form 0 to 1.The intensity image for each pixel has five decimal point for example 0.52344. Eight bits have being allocated for each pixel, thus the total number of bits for our test image was 524288 (obtain from 256x256x8).

As explained in section 3, a Matlab program was written to study how to compress and to reform back the image using fast Fourier transform, discrete cosine transform and wavelet transform using compression ratios of 64%, 89% and 92%. The compression ratio was calculated using the formula:

$$\text{Compression\%} = 100 - 100 \times \frac{\text{total number of bits after compression (bitcount)}}{\text{Ttotal number of bits before compression}} \quad (4.1)$$

Figure 3.2: Flowchart for image compression using DWT.

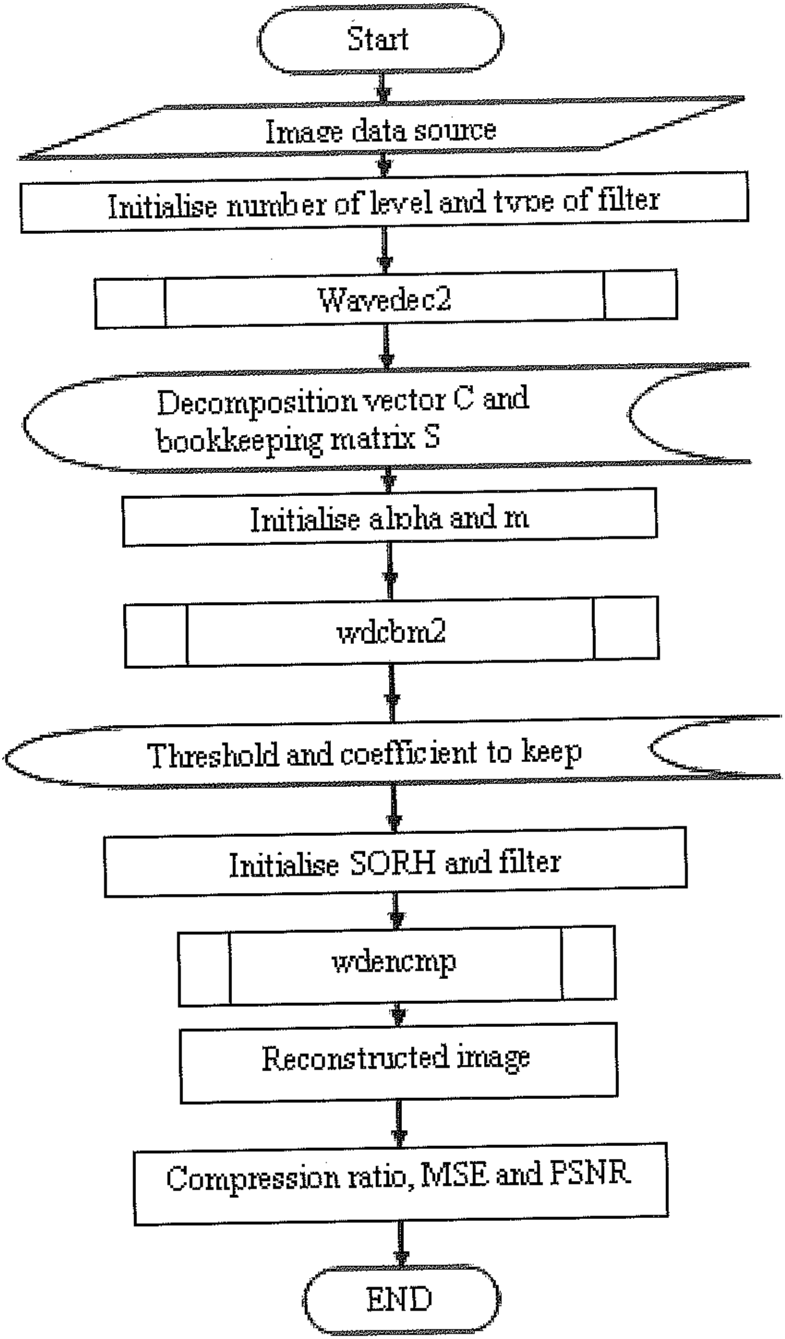


Figure 3.3: Flowchart for the main program of image compression using JPEG compression.

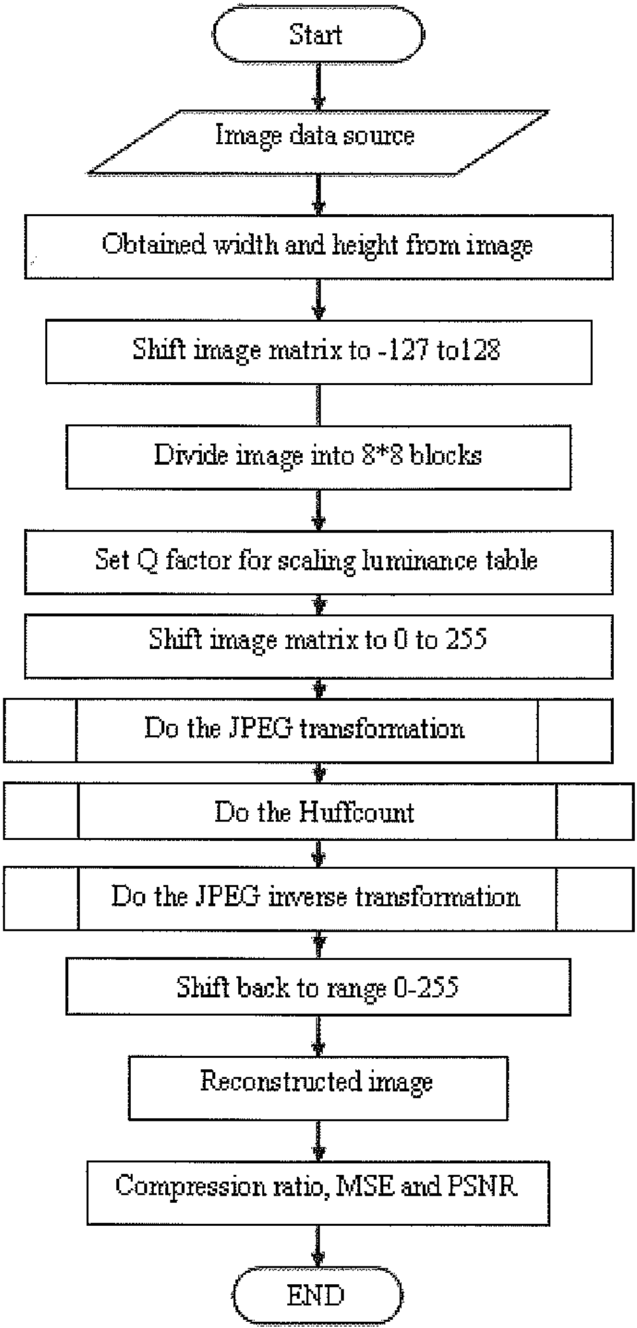


Figure 3.4: Flowchart for main program of image compression using "JPEG2000 like compression".

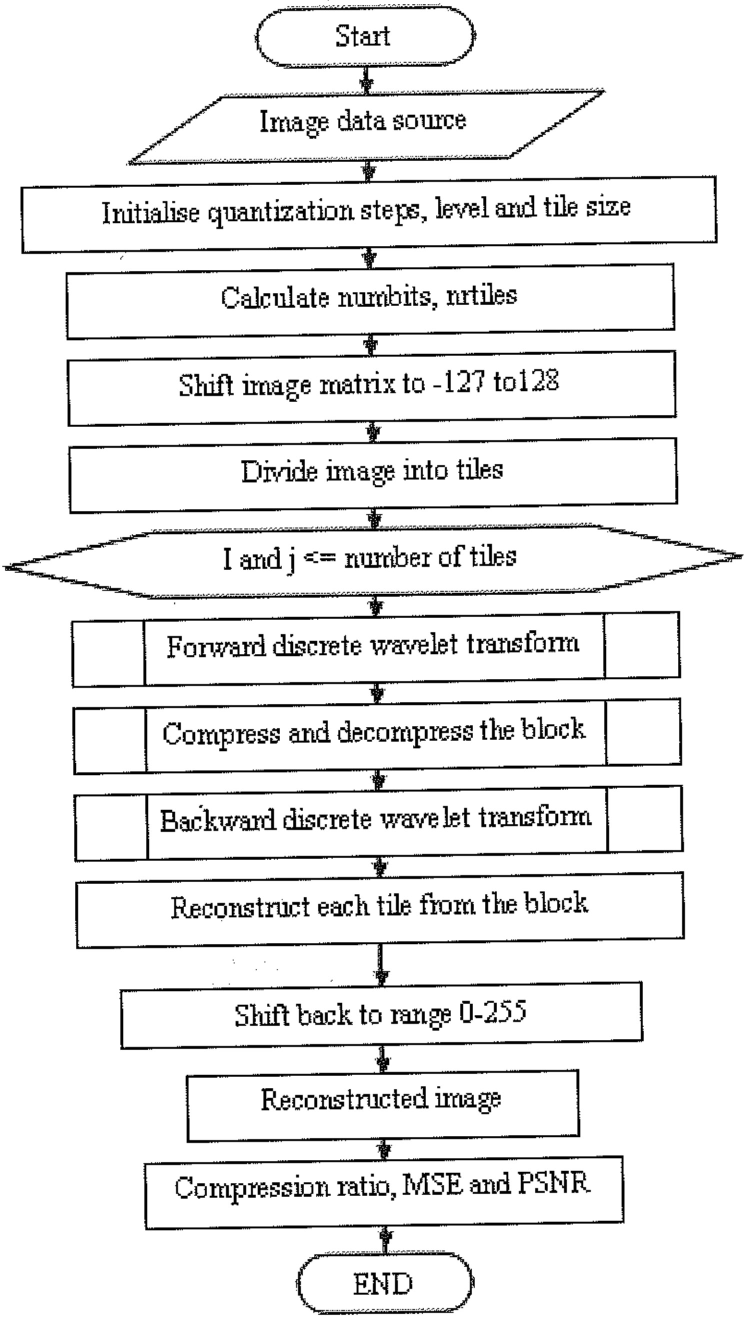


Image-compression can be defined as a process that alters the way image data are encoded in order to reduce the average size of an image file. Compression ratio used in this project for FFT and DCT can be calculated from the number of zeros divided by 64 which, represent the image block in terms of percentage, whereas for DWT is simply the ratio of coefficients to be kept.

Mean squared error (MSE) of an image can be defined as the ratio of the square of the difference between the image and the reconstructed image to the original image. In terms of equation

$$\mathrm{MSE} = \mathrm{sum}\left(\frac{\mathrm{sum}(x - \mathrm{img})^2}{(\mathrm{width} \times \mathrm{height})}\right) \tag{4.2}$$

where x is the original image,
img is the reconstructed image,
width and height are the dimension of the original image

The peak signal to noise ratio (PSNR) is the measurement of the peak error. The mathematical formula can be written as

$$\mathrm{PSNR} = 20 \log 10 \left(\frac{256}{\sqrt{\mathrm{MSE}}}\right) \tag{4.3}$$

A lower value for mean square error means lesser the error, and as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of signal to noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. Thus a compression scheme having a lower MSE (and a high PSNR) is a better one than a compression scheme with a higher MSE.

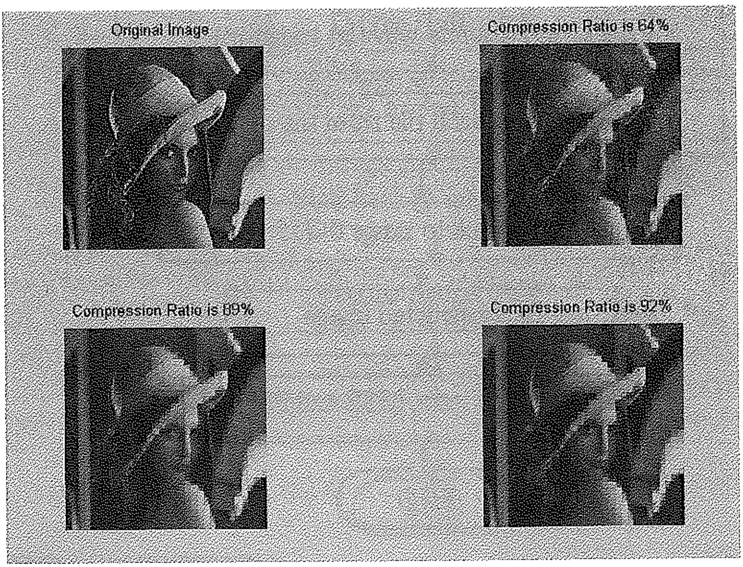


Figure 4.1: Compression using FFT with various compression ratios

From the results of both the visual image (Figure 4.1) and the PSNR (Table 4.1), it can be clearly shown that none of the above compressions was able to reproduce a good quality image.

Table 4.1: MSE and PSNR for FFT using different compression ratio.

Compression ratio	MSE	PSNR
64%	0.001030	78.03 dB
89%	0.002364	74.43 dB
92%	0.002809	73.68 dB

The second program executed was for the compression done using DCT.

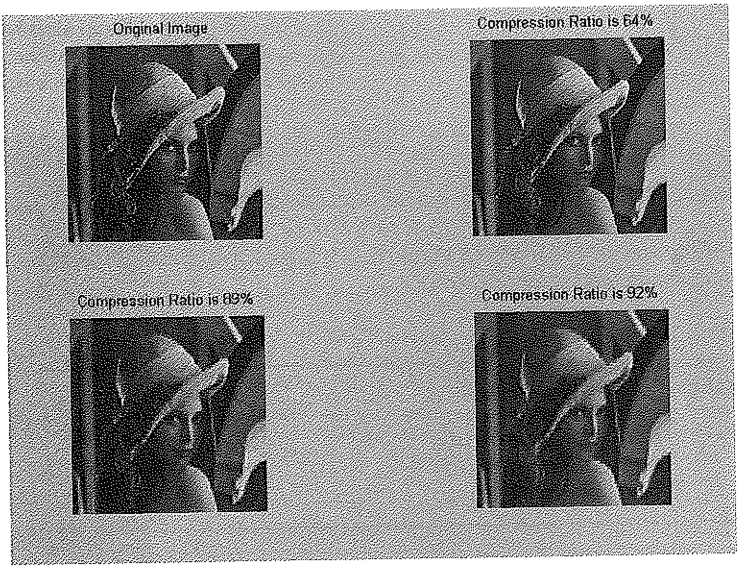


Figure 4.2: Compression using DCT with various compression ratios.

Table 4.2: MSE and PSNR for DCT using different compression ratio.

Compression ratio	MSE	PSNR
64%	0.000667	79.92 dB
89%	0.002106	74.93 dB
92%	0.002644	73.94 dB

From the visual image (Figure 4.2) and PSNR (Table 4.2), it can be clearly shown that only with at least a 64% compression ratio will DCT be able to reproduce a good quality image. Thus these satisfy the theory found in any reference text or reputable Internet sites which state that DCT is superior in image-processing to FFT.

The third program executed was for the compression using DWT.

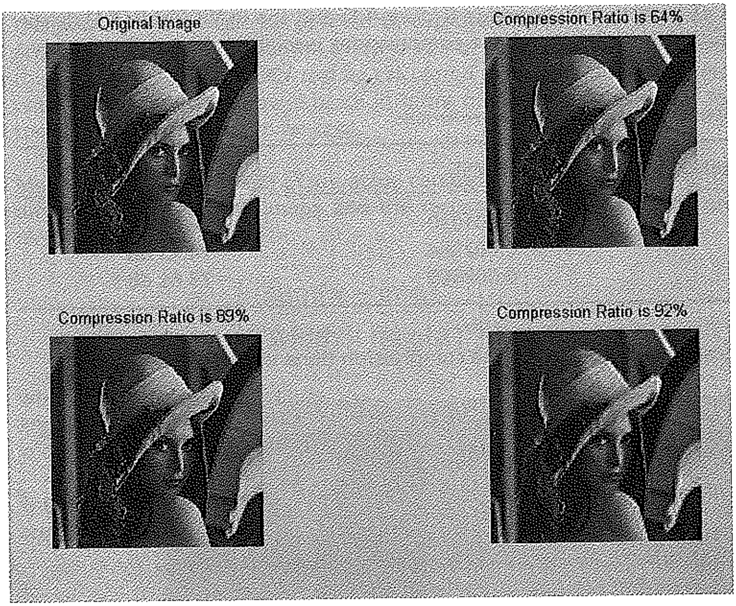


Figure 4.3: Compression using DWT with various compression ratios.

Form the visual results (Figure 4.3) and PSNR (Table 4.3), it is shown that the compression rates of DWT can be made much higher than DCT.

Table 4.3: MSE and PSNR for DWT using different compression ratio.

Compression ratio	MSE	PSNR
64%	0.000100	88.16 dB
89%	0.000633	80.15 dB
92%	0.001920	75.33 dB

From the above three figures, it is clear why FFT is not a popular means of image compression. This justifies why JPEG and JPEG2000 uses DCT and DWT respectively, instead of FFT.

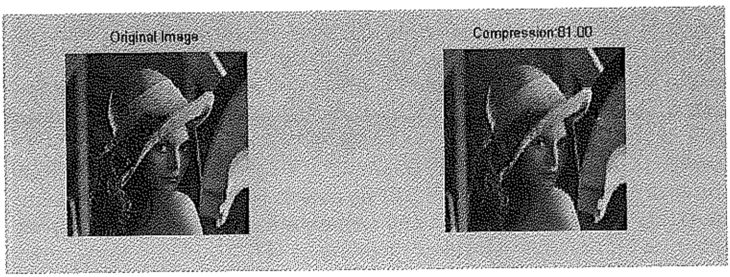


Figure 4.4: JPEG image compression, which will need 99567 bits to be transmitted across the wireless channel.

The fourth program executed was for a JPEG compression technique (Figure 4.4) in which the quality factor taken was 50 (as recommended by the JPEG community). This resulted in a compression of 81%, thus it will only need 99569 bits to be transmitted across wireless channel using Huffman and run-length encoding, with a mean square error of 0.001837 (PSNR of 75.52 dB).

In our final image compression technique (Figure 4.5), we have executed a "JPEG2000 like compression" in which we used an 8x8 tile size with a quantization step size of 1, in which we obtained a compression of 99.82%. Thus if we wanted to transmit the image across the wireless channel, we only needed 949 bits to be transmitted, and this produced a mean square error of 0.000003 (PSNR of 103.39 dB).

It should be noted that the bit-count was done using Huffman and run length encoding technique, instead of the complicated contact based coder (MQ-coder), which is used in the actual JPEG 2000. This high compression ratio is very useful when large amounts of data need to be transmitted from one point to another, such as during Internet tele-conferencing, satellite communication, or in High-Definition Television (HDTV) where time is of the essence.

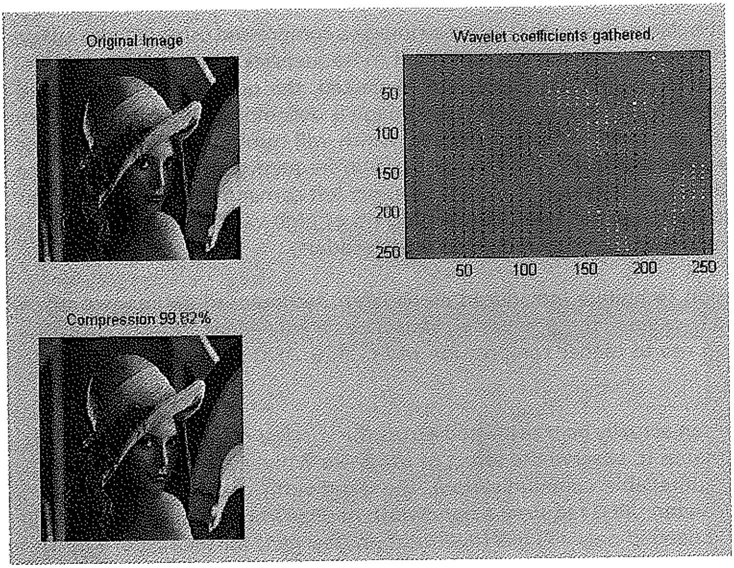


Figure 4.5: "JPEG200 like compression" needing 949 bits to be transmitted across the wireless channel.

Discussion and Conclusion

In this section we discuss the results obtained in the paper and the conclusions that can be drawn from it.

From section 4, the visual picture perspective quality and the PSNR for FFT, DCT and DWT can be described in a table form as shown in Table 5.1 and 5.2.

Table 5.1: Comparison between FFT, DCT and DWT in terms of picture after compression.

Compression in percentage	FFT	DCT	DWT
64%	Fair	good	Very good
89%	Bad	fair	Good
92%	Bad	bad	Fair

Table 5.2: Comparison between FFT, DCT and DWT in terms of PSNR after compression.

Compression in percentage	PSNR when using FFT method	PSNR when using DCT method	PSNR when using DWT method
64%	78.03 dB	79.92 dB	88.16 dB
89%	74.43 dB	74.93 dB	80.15 dB
92%	73.68 dB	73.94 dB	75.33 dB

From the above, it is clearly shown that when an image is compressed using FFT, the image is decomposed in time domain and this will result in frequencies with small magnitude, and when these small magnitude signals are neglected, the quality of reconstructed image is poor. But with DCT manipulation, the image is decomposed into its frequency domain. This gives rise to all real matrix values, which will result in a shorter time for manipulation, as compared with FFT. The lower value frequency in which most of the image information is kept is always found at the upper left corner of the 8x8 matrix. This makes it easier to compress the image, by getting rid of high-value frequencies. This is the main reason why JPEG uses DCT encoders and decoders.

DWT, a compression technique introduced much later, decomposed an image both in terms of time and frequency, thus less wavelet functions were needed to reconstruct the image, thus giving rise to higher compression rates. JPEG 2000 uses DWT compression technique.

Before the image can actually be transmitted across the wireless channel using different modulating techniques to observe the effect of (Additive White Gaussian Noise) AWGN and slow fading (unreliable communication), the bandwidth of JPEG and JPEG2000 needed to be calculated. JPEG required a bandwidth of 99567 bits (quality factor of 50) whereas JPEG2000 needed only 949 bits (quantisation step of 1) for tile size of 8x8. This results in a bandwidth of JPEG2000 to be only 1% of JPEG the same tile size (or JPEG 8x8 block size). Theoretically, the saving in bandwidth will result in quicker transfer of an image in a time-critical environment, such as telemedicine or authenticating image that uses digital signature.

For the modulation process, for data transmission, we had a quick review of both QPSK technique and GMSK [7], on which modulation scheme is superior and should be used. Each of them seems to have strong features that are desirable in the wireless network environment. First, most mobile phones are designed with Class C power amplifiers, which offer the highest power efficiency, yet because they are nonlinear, require the amplified signal to have a constant envelope thus making GMSK modulation more suitable in this situation. However, QPSK effectively utilizes bandwidth more efficiently, compared to GMSK, which requires more bandwidth to effectively recover its carrier.

Due to its linear amplification feature, QPSK is able to maintain low spectral side lobes, thus providing good adjacent channel performance. This is an important contribution to wireless systems because it enables a higher channel reuse factor. Furthermore, QPSK's importance in CDMA is evident with its efficient bandwidth use, enabling more users within a limited channel bandwidth.

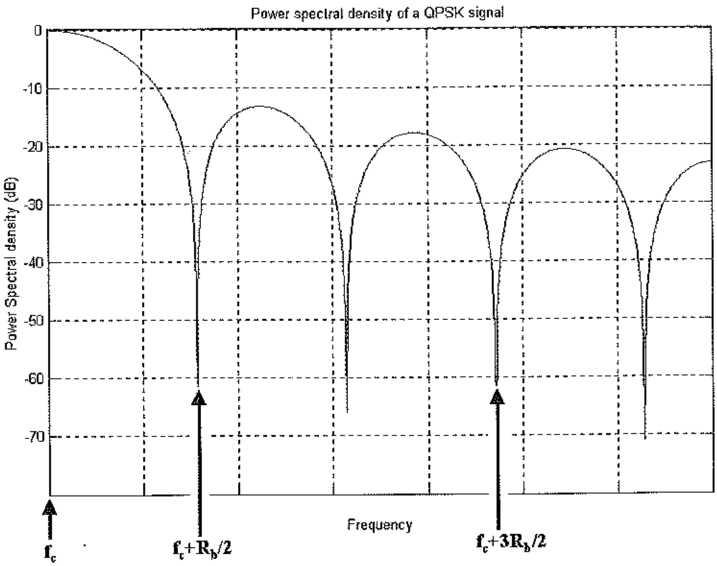


Figure 5.1: Power spectral density of a QPSK signal [7].

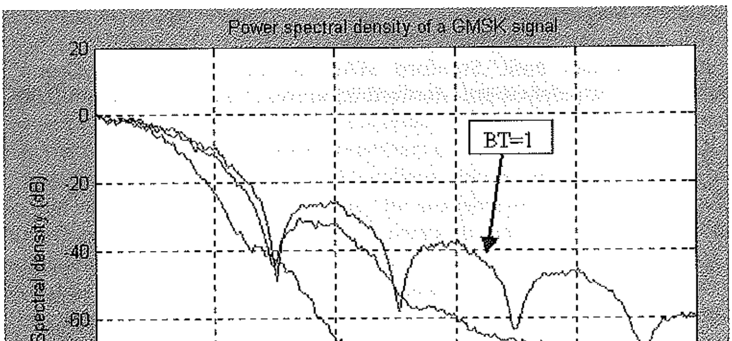


Figure 5.2: Power spectral density of a GMSK signal [7].

GMSK, meanwhile, makes its contribution to cellular systems in communications from the mobile to the base station. In the case of uplink, power is drained significantly from the mobile, necessitating a power-efficient amplifier. GMSK fulfills this need with its good power and spectral efficiency. Furthermore, due to its frequency modulating characteristic, GMSK shows a greater immunity to signal fluctuations.

Thus our central conclusion is that both QPSK and GMSK provide beneficial features, and although neither dominates the other, both can contribute to the advancement of wireless image-transmission. Indeed, a detailed study will be carried out in our next paper to actually see images reconstructed using both of these modulation techniques.

REFERENCES

Arno, S.(2003). An Introduction to JPEG 2000 like compression using Matlab, (Online). Available: http://www.math.ruu.nl/people/swart/wavelets.html. (2005,December 20)

Couch, L.W. (2001), Digital and Analog Communication Systems, (6th ed). USA: Prentice Hall.

Eric, W. W. (No date) Fast Fourier Transform., (Online). Available: http://mathworld.wolfram.com/FastFourierTransform.html (2005,June7)

Hagey, W. (1998). Huffman Coding, Queens University computer science lecture notes. (Online) Available: http://educ.queensu.ca/~compsci/units/encoding/huffman.html (2005, December 25)

Kinner G. S. (1999). The Compression Technology in Multimedia .(Online). Avaliable: http://www.scit.wlv.ac.uk/~c9581158/main.html#Run-Length Encoding (RLE) Method (2005, December 27)

Marshall, D. (2003), The Discrete Cosine Transform, (Online). Available: http://www.cs.cf.ac.uk/Dave/Multimedia/ node231.html(2006, January 20)

Rappaport, T. S., (2002).Wireless Communication-Principles and Practice,(2nd ed). USA. Prentice Hall.

Saha Subhasis (2001)."Image Compression-from DCT to Wavelet: A Review" ACM crossroad Student Magazine. (Online). Available:Website: http://www.acm.org/crossroads/xrds6-3/ sahaimgcoding.html(2006, January 20)

Twicken Joe,(1998). "Wavelet Basic",Stanford STARLab Wavelet Research Group". (Online). Available: http://www-star.stanford.edu/projects/sswrg/basics.html(2006, January 20)