Image Compression Using Embedded Zero-tree Wavelet Coding (EZW)

1 author:

Balaji Vasan Srinivasan
Adobe Systems Inc

Image Compression Using Embedded Zero-tree Wavelet Coding (EZW)

Balaji Vasan Srinivasan

balajiv@umd.edu

ENEE 631: Digital Image Processing
Instructor: Prof. Min Wu

Course Project 1

1. Introduction

Transform coding forms an integral part of compression techniques. In transform coding a reversible linear transform is used to map the image into a set of coefficients which are then quantized and coded. The advantage of using a transform is that it packs the data into a lesser number of coefficients. The main purpose of using the transform is thus to achieve energy compaction. The type of transform used varies between different codecs.

Discrete Cosine Transform (DCT) is a commonly used in different compression techniques. DCT closely resembles the optimal transform in terms of performance. The DCT based compression techniques are efficient.

However, it has its own shortcomings. New schemes are developed to overcome the problems with the DCT based compression techniques. The recent compression techniques have mainly concentrated on the usage of Discrete Wavelet Transforms (DWT).

In this project, an Embedded Zero-tree Wavelet Encoder is developed and its performance is studied for different parameters and compared with DCT based encoder.

2. Different formats of Images

Digital images can be stored in different formats. JPEG is a format developed by the Joint Photographic Expert Group. Images in this format are compressed to a high degree. JPEG's purpose is to achieve high compression ratio with images containing large number of colors. Graphics Interchange Format (GIF) is an 8-bit per pixel format. It supports animations as well. GIF images are compressed using lossless LZW data compression technique. Portable Network Graphics (PNG) is another lossless data compression which is close to GIF but supports 24-bit color palette. It can be seen that JPEG can be used whenever the size is a criterion.

3. JPEG

In this project, we mainly study the performance of JPEG. The first format of JPEG standard was the JPEG Baseline. JPEG baseline compression standard divides the image into blocks of size 8*8 and performs DCT on each block. The transform coefficients are then quantized and encoded. The quantization is a lossy part but enables higher compression. The decoder recovers the DCT coefficients and performs inverse DCT to obtain the original image. The various processes in a JPEG Baseline encoder and decoder are shown in Figure 1.

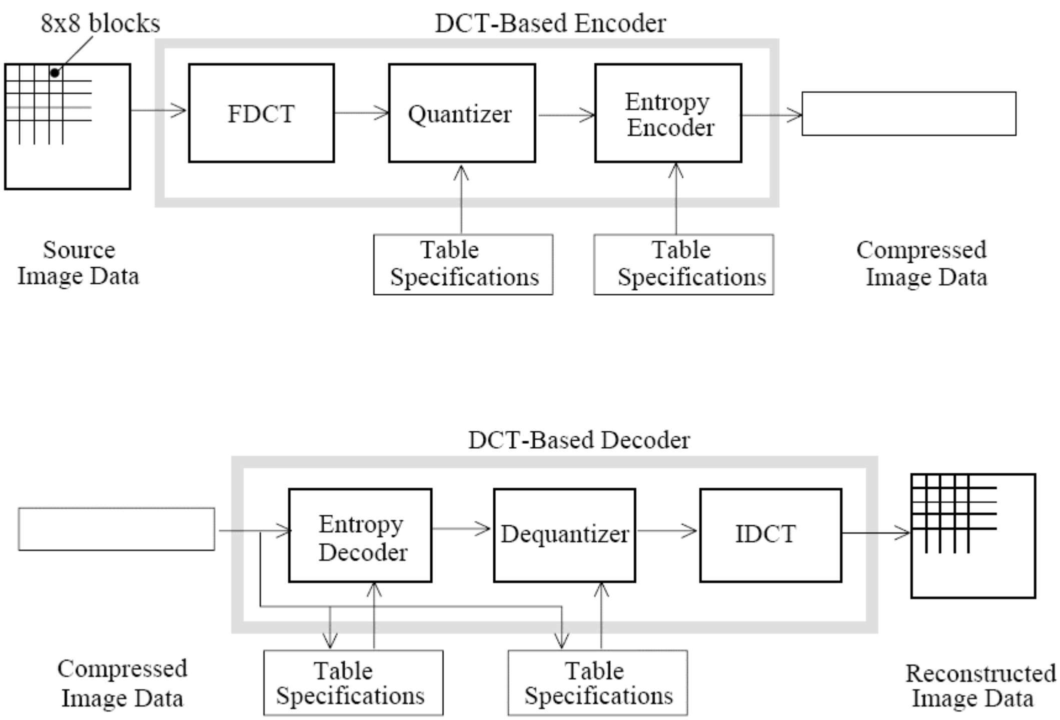


Figure 1

From The JPEG Still Picture Compression Standard by Gregory K. Wallace

The basic idea behind the JPEG baseline is that the DCT transform performs sufficient energy compaction and thus the higher order AC coefficients are approximately zero and thus using run-length coding a good amount of compression can be achieved.

While good compression is achieved by this method, it does not achieve high compression with lesser distortions. In order to achieve this Wavelet transforms are used in latest standards.

4. Discrete Wavelet Transform

Wavelet based image compression techniques are increasingly used for image compression. The wavelet based coding divides the image into different sub-bands. These sub-bands can then be successively quantized using different quantizers to give better compression. The wavelet filters are so designed that the coefficients in each sub-band are almost uncorrelated with each other.

In case of an image, the low frequency part of the image carries most information and the high frequency part adds intricate details to the image.

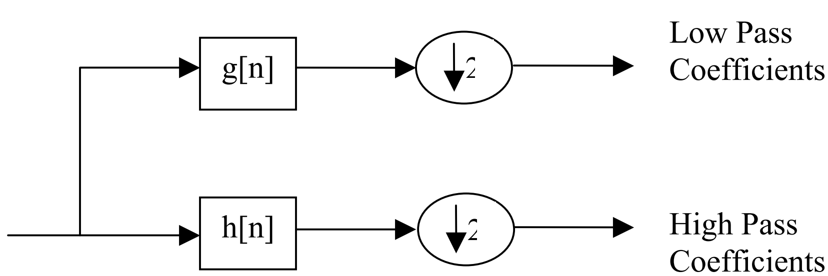


Figure 2

In case of an image the filter is applied along the rows and then applied along the columns, thus the operations results in four bands, low-low, low-high, high-low and high-high. The low-low frequency part can further be processed so that it is again subdivided into further four bands. Figure 3B and 3C show the sub-bands of image 3A. Image 3A is divided into 4 bands in 3B and the low-low is further divided into 4 bands in 3C.

A

B

C

Figure 3

5. Embedded Zero Tree algorithm

Embedded Zero tree algorithm is a simple yet powerful algorithm having the property that the bits in the stream are generated in the order of their importance. The first step in this algorithm is setting up an initial threshold. Any coefficient in the wavelet is said to be significant if its absolute value is greater than the threshold. In a hierarchical sub-band system, every coefficient is spatially related to a coefficient in the lower band. Such coefficients in the higher bands are called 'descendants'. This is shown in figure 4.

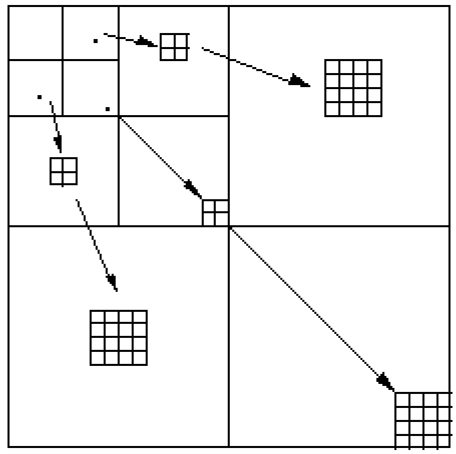


Figure 4

http://www.acm.org/crossroads/xrds6-3/sahaimgcoding.html

If a coefficient is significant and positive, then it is coded as ‘positive significant’ (ps). If a coefficient is significant and negative, then it is coded as ‘negative significant’ (ns). If a coefficient is insignificant and all its descendants are insignificant as well, then it is coded as ‘zero tree root’ (ztr). If a coefficient is insignificant and all its descendants are not insignificant, then it is coded as ‘insignificant zero’ (iz).

The algorithm involves two passes – Dominant pass and Subordinate pass. In the dominant pass, the initial threshold is set to one half of the maximum pixel value. Subsequent passes have threshold values one half of the previous threshold. The coefficients are then coded as ps, ns, iz or ztr according to their values. The important part is that if a coefficient is a zero-tree root, then the descendants need not be encoded. Thus only the significant values are encoded

In the subordinate pass, those coefficients which were found significant in the dominant pass are quantized based on the pixel value. In the first pass, the threshold is half of the maximum magnitude, so the interval is divided into two and the subordinate pass codes a 1 if the coefficient is in the upper half of the interval and codes a 0 if the coefficient is in the lower half of the interval. Thus if the number of passes is increased, the precision of the coefficients is increased.

5. Implementation

The embedded zero-tree wavelet algorithm was implemented using Matlab. The function *EZWenco.m* encodes the given wavelet using EZW algorithm into a vector of significant coefficients (output of Dominant pass) and their significance (output of the Subordinate pass). *EZWenco.m* uses *dominant.m* for the dominant pass and *subordinate.m* for the subordinate pass. The function *checkchildren.m* is used to check if the children are significant or insignificant. The function *setzerotree.m* sets the current coefficient as a zero-tree root and it instructs the encoder not to code the children of the particular coefficients. In the decoder part, after the dominant pass, the approximate values of the coefficients are obtained, further refinement of the coefficients are done by decoding the significance and by subsequent passes. This part is done by *EZWdeco.m*. The initial wavelets are obtained by the functions *band4afilter.m* which divides an image into 4 sub-bands by successively applying the filter along the rows and columns. The *bandsfilter.m* combines the 4 bands into one image. The *subband_anal.m* applies band4afilter.m to an image successively on the low-low band. The *subband_synt.m* retrieves the image from the sub-bands split by the *subband_anal.m*.

6. Results and Comparisons

The EZW scheme implemented divides an image into blocks of standard size and performs coding on each of the blocks. The decompressed images for a bit rate of 3bpp for different block sizes are shown in Figure 5a – 5f.

It can be seen that for very small block sizes and very large block sizes the visual distortions are more pronounced. In case of smaller images the distortions may be due to the size of the header details needed to be added for each block, which would be significant compared to the information of the block itself. In case of large block size the initial threshold is high and hence needs a lot of passes to achieve significant amount of visual quality.

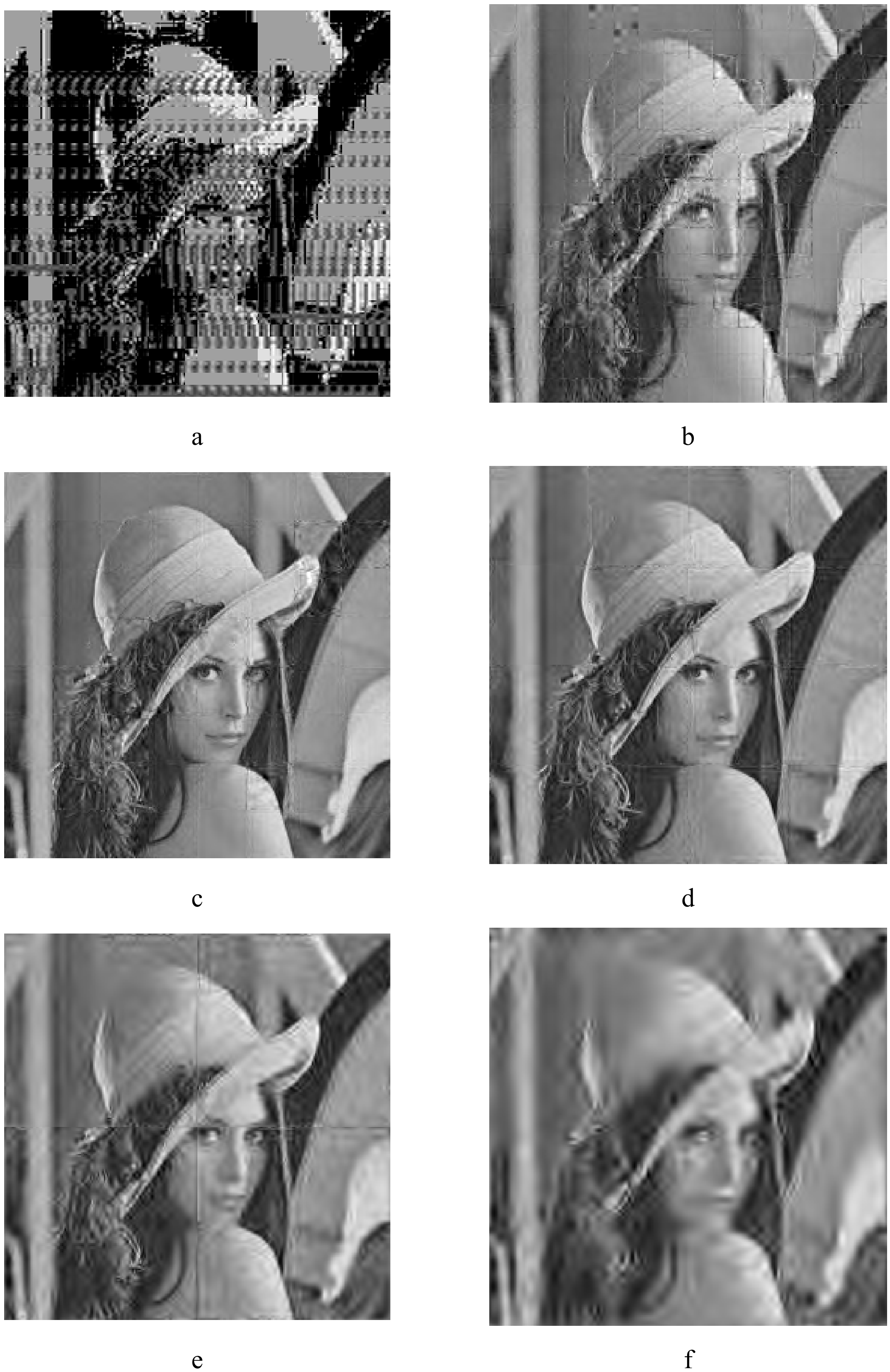


Figure 5

Figure 6 shows the plot of PSNR vs the block size. The X-axis is in log2 scale. Clearly, the PSNR is high for a block size of 32 and 64. In the implementation the block size used is 64.

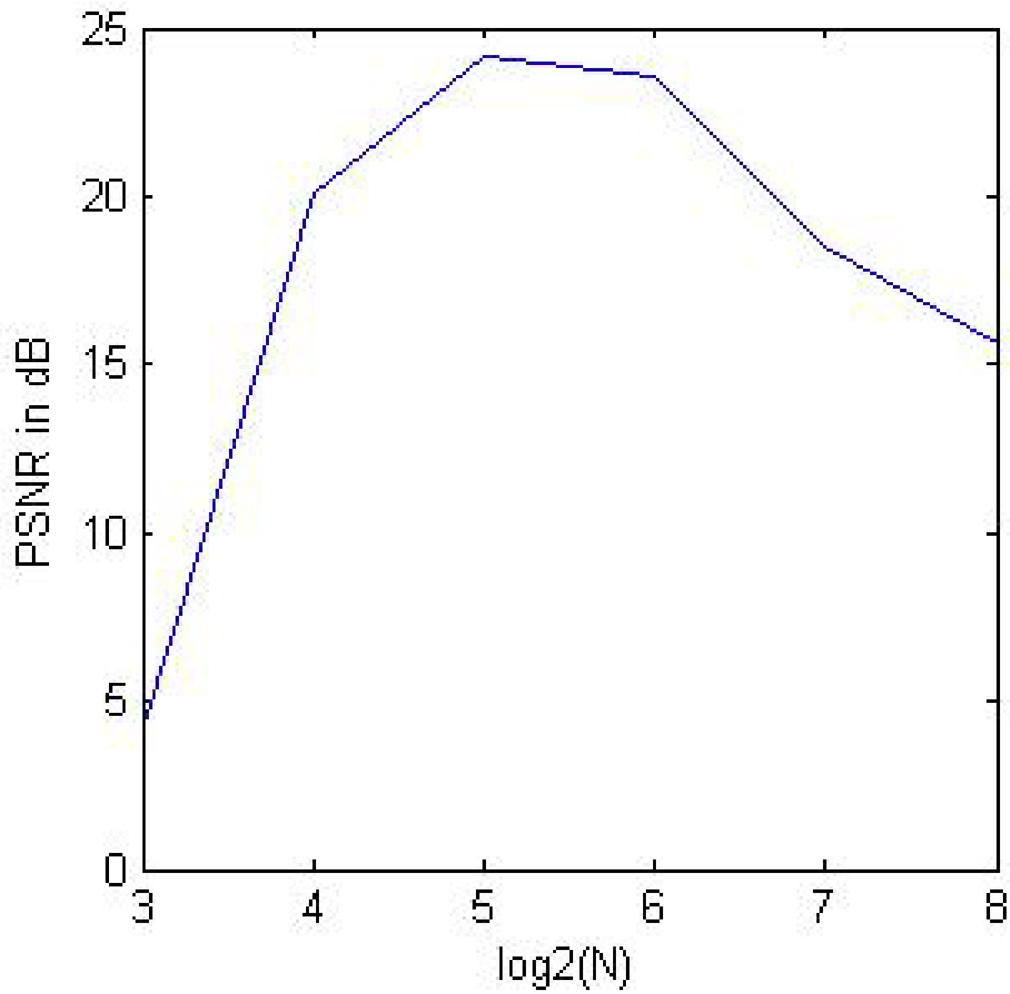


Figure 6

The EZW scheme was applied to different bit rates and different block size. The image results for different block sizes and bitrates are shown in figure 7,8 and 9. The compression ratio vs PSNR plot is shown in figure 10.

A – 2bpp

B – 3bpp

C – 4bpp

D – 6bpp

Figure 7 – Block size 16

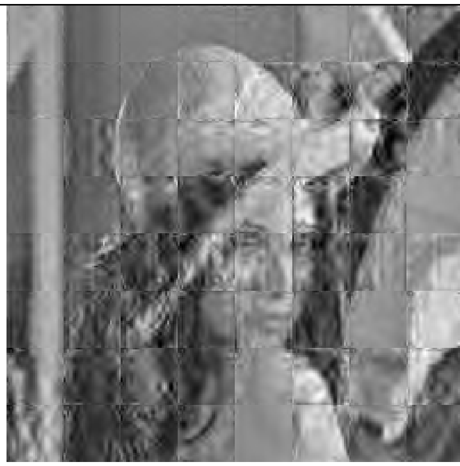
	
A – 1 bpp	B – 2bpp
	
C – 3bpp	D – 4bpp
	
E – 6bpp	F – 8bpp

Figure 8 : N=32

	
A – 1bpp	B – 2bpp
	
C – 3bpp	D – 4bpp
	
E – 6bpp	F – 8bpp

Figure 9 – N=64

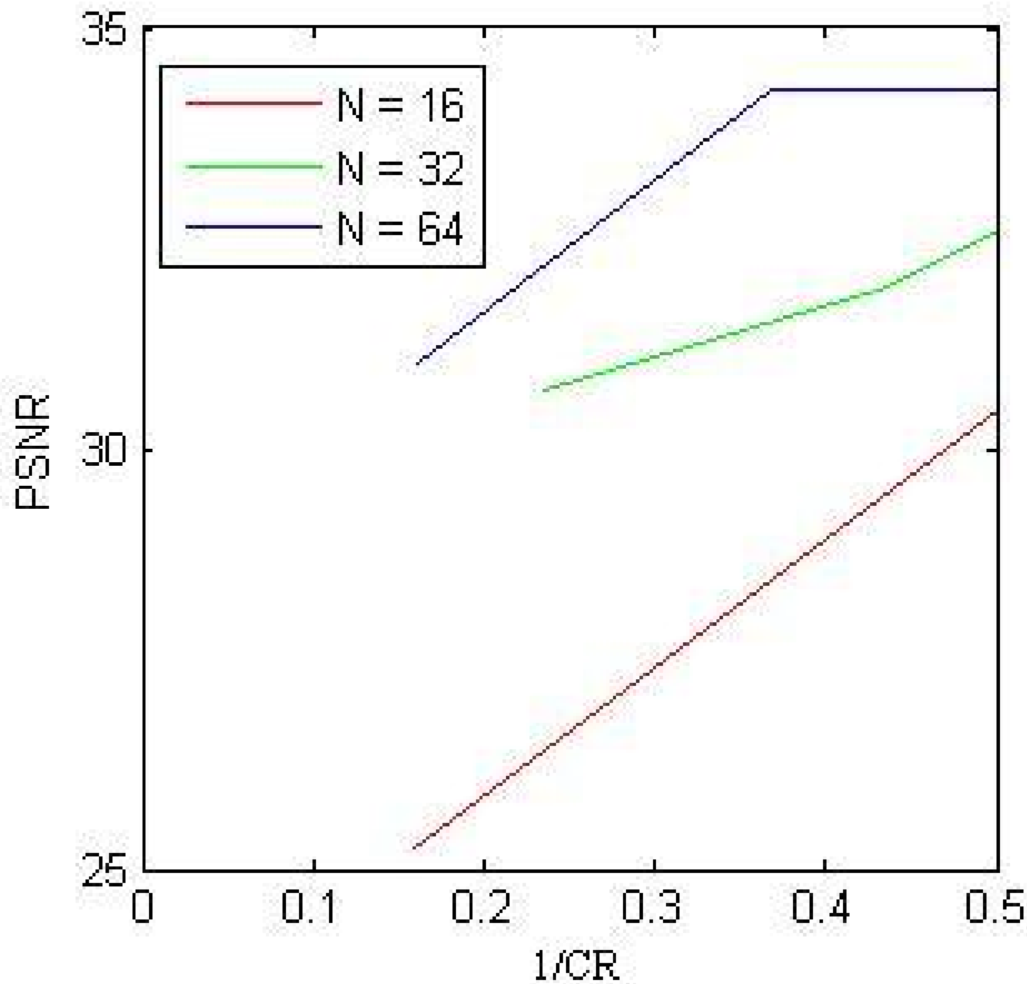


Figure 10 (1/CR in X-axis)

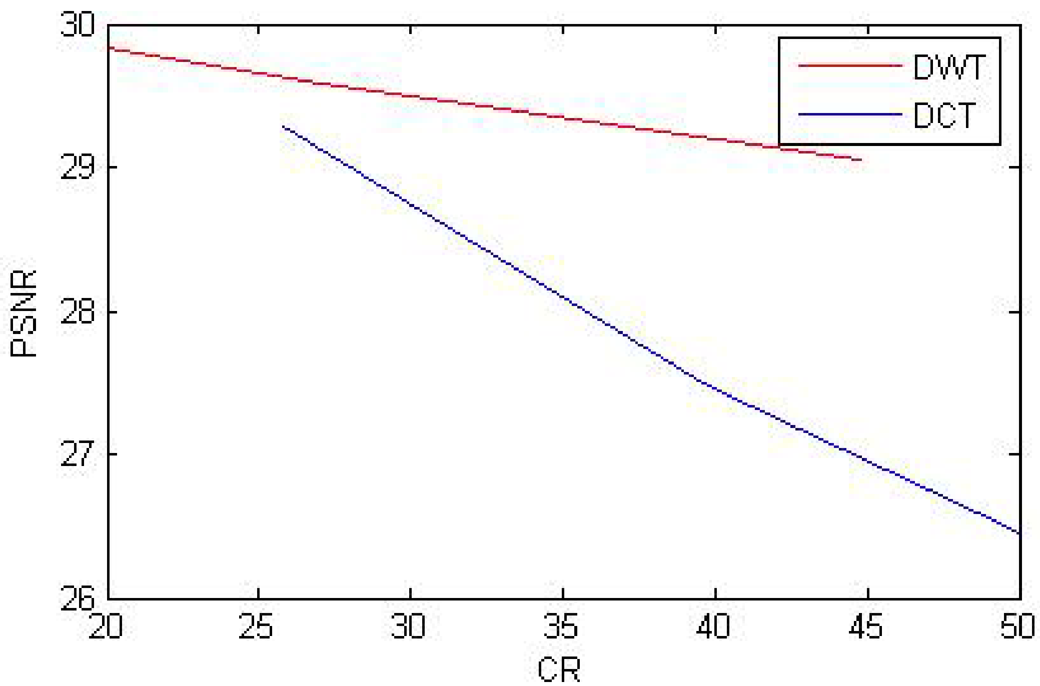
The initial wavelet used in all these images is obtained by db8 transform. The scheme was studied with different transforms such as Haar, db4, db7 and db8. The performance is shown in figure 11 for a bit rate of 3bpp and block size of 64.

	
A - Haar	B - Db4
PSNR = 34.2324	PSNR = 34.2014
	
C - Db7	D - Db8
PSNR = 34.2399	PSNR = 34.0097

Figure 11

The PSNR value indicates that there is no appreciable change with the changes in the transform used. The PSNR almost remains the same for the wavelet transforms Haar, db4, db7 and db8.

The plot of the PSNR vs CR for the EZW scheme and the JPEG like encoder is shown in Figure 12. It can be seen that EZW has a marginally better performance than the JPEG like encoder



7. Conclusion

In this project the Embedded Zero-tree Wavelet coding scheme was implemented. The effect of using different Wavelet transforms, using different block sizes and different bit rates were studied. Non-block based coding was also studied and it was observed that it takes a lot of computation. Finally, the EZW scheme was compared with the DCT based JPEG Baseline scheme. It was observed that the EZW scheme is marginally better than the DCT scheme.

References

[1] Jerome Shapiro “Embedded Image Coding Using Zerotrees of Wavelet Coefficients,” IEEE Transactions in Signal Processing, 1993

[2] Majid Rabbani, Rajan Joshi, “An overview of the JPEG2000 still image compression standard,” Signal Processing: Image Communication 17, 2002.

[3] Gregory Wallace, “The JPEG Still Picture Compression Standard,” IEEE Transactions on Consumer Electronics, 1991.

[4] Bryan Usevitch, “A Tutorial on Modern LossyWavelet Image Compression : Foundations of JPEG 2000,” IEEE Signal Processing Magazine, 2001.

[5] Jerome Shapiro “Embedded Image Coding Using Zerotrees of Wavelet Coefficients,” IEEE Transactions in Signal Processing, 1993

[6] AK. Jain, “Fundamentals of Digital Image Processing,” Pentice Hall , 2000.

[7] R. C. Gonzalez, R. E. Woods, “Digital Image Processing,” Pearson Education

[8] http://www.owlnet.rice.edu/~elec539/Projects99/BACH/proj1/report/node11.html

[9] http://perso.orange.fr/polyvalens/clemens/ezw/ezw.html

[10] http://faculty.washington.edu/edhong/523/slides/EZW_ShuFang.ppt