A COMPARATIVE STUDY OF IMAGE COMPRESSION METHODS

Ashwin Swaminathan, Gaurav Agarwal

ashwins@umd.edu, gaaga@umd.edu

ENEE631: Digital Image Processing
Instructor : Dr. Min Wu

Course Project 1

1. INTRODUCTION

Transform coding has become a standard model for image coding. DCT based image codecs have been developed and their performance studied. Despite the phenomenal performance of the JPEG baseline algorithm that out performed the existing schemes, there were some short comings which became apparent later. Various algorithms have been proposed to improve the performance of the compression scheme. Recent works have mainly concentrated on the use of the discrete wavelet transform (or DWT) in image coding and compression applications. In this project, we study the commonly used algorithms for image compression and compare its performance.

JPEG's goal has been to develop a methods for continuous-tone image generation which meets the following requirements [1] .

1. Improved compression efficiency.
2. Lossy to lossless compression.
3. Multiple resolution representation.
4. Embedded bit-stream (progressive and SNR scalability).
5. Tiling.

6. Region-of-interest (ROI) coding.
7. Error resilience.
8. Random code stream access and
9. Improved performance to multiple compression/ decompression cycles.

2. IMAGE FORMATS

Various image formats exist. Each one of them have certain specific applications and are widely used in digital image storage and retrieval. We briefly discuss some of the basic image formats in this section. The JPEG (named after Joint Photographers Expert Group) format uses lossy compression to achieve high levels of compression on images with many colors. The compression works best with continuous-tone images, that is, images where the change between adjacent pixels is small but not zero.

The GIF (Graphics Interchange Format) uses lossless compression to achieve medium levels of compression on images with up to 256 colors. The compression works best on images with few colors or images in which one color is dominant. GIF compression acts best on identical, adjacent pixels (or rows of identical, adjacent pixels).

The PNG format is lossless like GIF but supports larger numbers of colors and more flexible transparency. The TIFF format can store up to 24-bit images with no loss, making them better than JPEG files for archiving images when space is not critical.

3. LITERARY SURVEY

3.1. Baseline JPEG

In this project, we mainly study the JPEG compression standards in detail. Figure 1 describes the JPEG compression process. JPEG divides up the image into 8 by 8 pixel blocks, and then calculates the discrete cosine transform (DCT) of each block. A quantizer rounds off the DCT coefficients according to the quantization matrix (can be image dependent). This step produces the "lossy" nature of JPEG, but allows for large compression ratios. JPEG's compression technique uses a variable length code such as run length codes or LZW on these coefficients. For decompression, JPEG recovers the quantized DCT coefficients from the compressed data stream, takes the inverse transform and displays the image. The

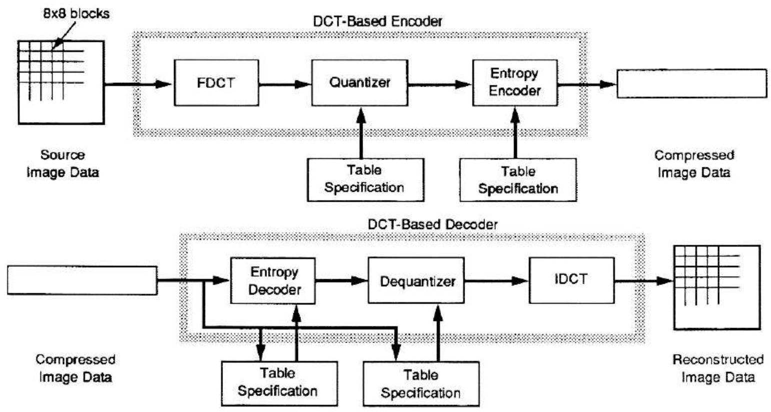

Fig. 1. The Baseline JPEG algorithm

basic idea behind this scheme is that the discrete cosine transform provides good energy compaction properties and hence we would require less number of bits to quantize and store these coefficients. Further, the quantized AC coefficients are mostly zeros and therefore can be run-length coded to give better compression ratios. For further details of the baseline JPEG algorithm, refer [2].

3.2. The Wavelet Transform

Wavelet based techniques for image compression have been increasingly used for image compression. The wavelet uses subband coding to selectively extract different subbands from the given image. These subbands can then be quantized with different quantizers to give better compression. The wavelet filters are specifically designed to satisfy certain constraints called the smoothness constraints. The wavelet filters are designed so that the coefficients in each subband are almost uncorrelated from the coefficients in other subbands [3]. The wavelet transform achieves better energy compaction than the DCT and hence can help in providing better compression for the same Peak Signal to Noise Ratio (PSNR). A lot of research has been done on the performance comparison of the DWT and DCT for image compression. A comparative study of DCT and wavelet based image coding can be found in [5].

The Embedded Zerotree Wavelet or popularly known as EZW is an efficient coding scheme developed by Shapiro [4]. The author in this landmark paper introduces an efficient coding scheme based on the multi-resolution nature of wavelet transforms. The resulting algorithm gave a better performance at low bit rates over the then existing schemes. The EZW marked the beginning of a new era of wavelet coding and ignited a lot of research work in this field. The two important features of the EZW coding are significance map coding and successive approximation quantization. This algorithm exploits the energy compaction properties and the self-similar and hierarchical nature of the wavelet transform. The hierarchical nature facilitates coding as it forms a tree structure. Inter band prediction is used to code the positions of the significant coefficients. The EZW algorithm does not code the location of significant coefficients but instead codes the location of zeros. For further details, the readers are referred to papers [3, 4].

The EZW algorithm was further extended by Amir et. al to give a new scheme called the Set Partitioning in Hierarchical Trees (SPIHT) [6]. SPIHT achieved better performance than the EZW without having to use the arithmetic encoder and so the algorithm was computationally more efficient. The SPIHT uses a more efficient subset partitioning scheme. Due to this, even binary encoded transmission achieves almost similar performance compared to EZW. The better performance of the SPIHT over EZW can be attributed to better wavelet filters (7/9 orthogonal wavelet filters instead of length 9 QMF filters), separation of the significance of the child nodes from that of the grand child nodes, and separation of the child nodes from the parent.

A comparative study of DCT and wavelet based methods was reported in [5]. In this work, the authors study the performance difference by comparing the entire coding scheme on the same footing. The authors indicate that the wavelet transform outperforms the DCT by around 1 dB PSNR. Some interesting results have been described by the authors in this work. Wavelet based JPEG like image coding has been shown to increase the PSNR by 1 dB over baseline JPEG. DCT-Based Embedded Image coding has been suggested in [8]. This results in a modified EZW algorithm based on the fact that the 8 x 8 DCT of the image can be viewed as a 64-subband decomposition and hence the tree structures introduced by shapiro can be used.

The drawback of EZW and SPIHT algorithms are that they are not ‘resolution scalable’. EBCOT (embedded block coding with optimized truncation) proposed by Taubman [17] is both ‘SNR scalable’ and ‘resolution scalable’. This algorithm is block based i.e. it encodes blocks independently (Embedded block based coding).Due to this, a particular region of interest (ROI) can be decoded separately without decoding the full image. Also the embedded bit stream includes the information about the number of bits to be decoded, to give the optimal reconstructed quality at a given bit rate. Apart from providing additional functionality, it give 0.25 dB (on an average) better performance than SPIHT. JPEG2000 [18] standard is based on EBCOT.

The other results of the literary survey have been simulated and the corresponding results obtained can be found in subsequent sections.

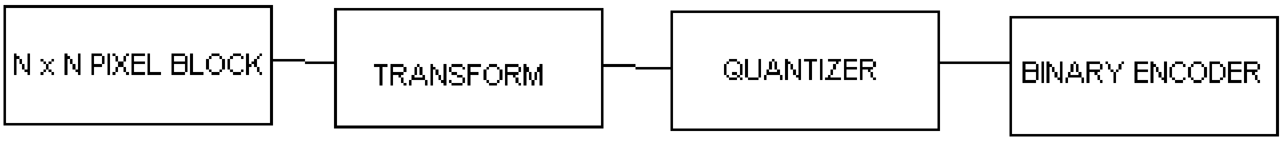

Fig. 2. The Basic Framework for compression algorithms

4. RESULTS AND DISCUSSIONS

We have used the basic framework as shown in figure 2. The image is first converted to a NxN block (N can also be the size of the image in which case the whole image is considered together). The transform of the image is taken and then coded. Many possible cases arise. Each one of these modules can be separately considered and its impact on the entire model be studied. In this work, we compare the performance of the DCT and the DWT transforms with different quantizers and encoders such as EZW, SPHIT, EBCOT etc. The results have been obtained for different block lengths such as N = 8, 16 and 32. The impact of run length coding, arithmetic coding and entropy coding techniques have been explored.

4.1. Baseline JPEG and its modifications

In this scheme, the image is initially converted to 8x8 blocks. DCT of each of the blocks is taken separately and then quantized and zig-zag coded. The DC components of the image are coded separately and the AC components are run-length coded to reduce the length of the sequence. Zigzag coding with wavelet transform scheme has been discussed in [5]. The wavelet transform is coupled with baseline JPEG quantizer with the DCT in JPEG replaced with DWT. A gain of 1 dB is obtained over the baseline JPEG scheme [5].

4.2. DCT Based Embedded Image coding

In this scheme the EZW scheme is used to quantize the DCT coefficients. The logic behind this scheme is that the 8x8 DCT image representation can be thought as a 64 subband decomposition and each of the coefficients can be considered as a separate coefficient. The DCT based scheme is computationally more efficient than the DWT based scheme and the loss in performance is only 0.7 dB. The algorithm was studied and implemented. Some of the obtained compressed images are shown in Fig. 3 and the compression ratio vs PSNR graph is shown in Fig. 4. The effect of block sizes was studied and the

Fig. 3. The results of Compression with DCT and Embedded Zerotree coding for compression ratios 4, 2 and 1.2.

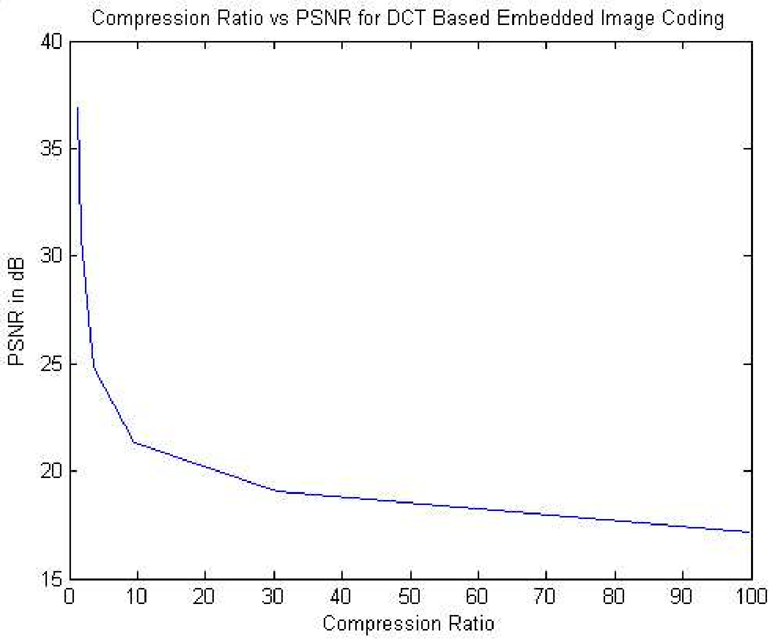

Fig. 4. Compression Ratio vs PSNR graph for DCT based Embedded Zerotree coding

results are shown in Fig. 5 and Fig. 6. As observed from the figures, increase in block sizes reduces the compression ratio.

4.3. Wavelet transform based Embedded Zerotree coding

The details of this scheme can be found in [4]. This scheme efficiently exploits the parent-child relationship of the DWT coefficients to code and compress them efficiently. Symbols are used to represent the four different kinds of node types (ps, ns, ztr and iz) in the dominance table. The significance table contains the elements 0 and 1. From our analysis, it was found that arithmetic coding of the dominance table and the significance table together achieved better compression ratios and this was adopted in this project. In this case only 4 symbols were used to represent all the different kinds of symbols in the dominance table and the significance table. The presence of the arithmetic encoder part however makes

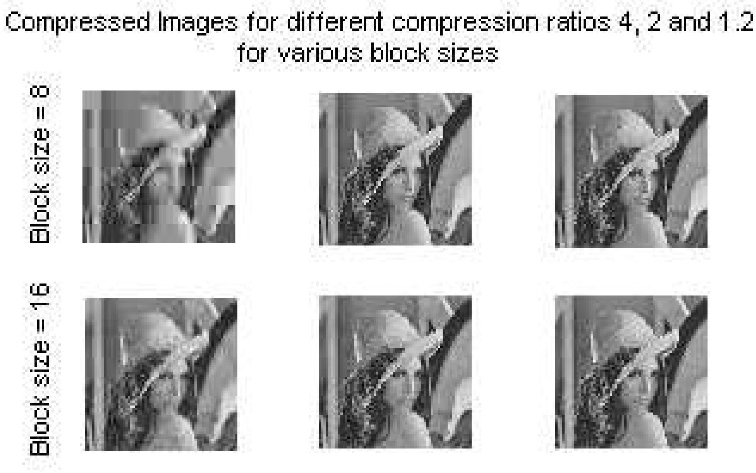

Fig. 5. Some Compressed images for block based DCT and Embedded Zerotree coding for compression ratios 4, 2 and 1.2.

the scheme computationally inefficient. This also forms one of the advantages of the SPIHT scheme over the EZW. This scheme was first implemented for different minimum thresholds and then the effects of different block sizes on the performance was studied. Some of the compressed images for various compression ratios are shown in Fig. 7. The Compression Ratio vs PSNR graph is shown in Fig. 8.

The effects of the block sizes was also studied. The compressed images for N = 8, 16 and 32 are shown in Fig. 9. The Compression Ratio vs PSNR for different block sizes are shown in Fig. 10. As expected increasing block size reduces the PSNR for the same compression ratio. As shown in figure, increase in block size implies better compression ratio for the same PSNR and so the performance improves. This is expected because by increasing the block size, we are using lesser coefficients per block and so we end up storing a fewer number of coefficients and so we achieve a better compression. The complete analysis of this is done in the next section.

4.4. Wavelet transform based SPIHT coder

The details of the scheme can be found in [6]. This scheme was first run for different bit lengths and then the effects of different block sizes on the performance was studied. Some of the compressed images for various compression ratios are shown in Fig. 11. The Compression Ratio vs PSNR graph is shown in Fig. 12. The effects of the block sizes was studied. The compressed images for N = 8, 16 and 32 are shown in Fig. 13. The Compression Ratio vs PSNR for different block sizes are shown in Fig. 14. As shown in figure, increase in block size does not improve the compression ratio a lot compared to the

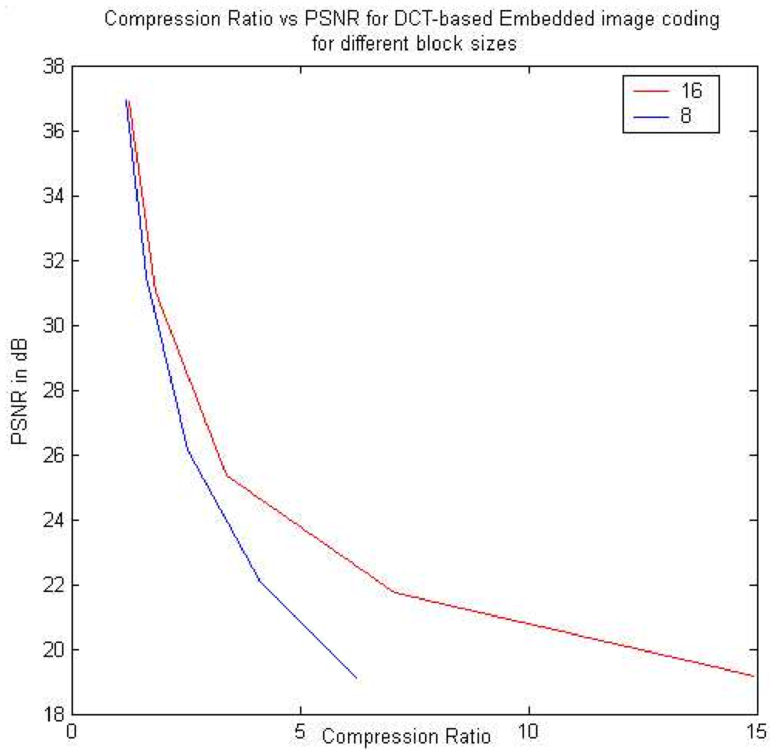

Fig. 6. Compression Ratio vs PSNR graph for DCT based Embedded Zerotree coding for different block sizes N = 8 and 16

previous case of the DWT with EZW. This can be attributed to the properties of the SPIHT algorithm.

4.5. Block DWT Based JPEG2000 (Annex C and D) Compression with our implementation

The details of the scheme could be found in [18]. This scheme is based on EBCOT [17]. The salient features of the algorithm is the resolution scalability and the embedded optimal truncation for achieving the best quality at a given bit rate. This is achieved by making the image blocks independent of each other and by recursively optimizing the truncation points respectively. The bitstream is encoded using a arithmetic coder(MQ_coder) and the redundancy within neighboring bits is used to generate the probability model. The algorithm was studied and Annex D of [18] was implemented to develop a probability model. This is given input to the MQ_coder to compress the bitstream. On the decoder side the same context model is used to decompress the bitstream. Each plane is encoded in three passes: Significant pass, Refinement pass and Cleanup Pass. The algorithm was tested for various compression ratio and block size. The results are shown in Fig.17 and Fig.18. All the block sizes above 16 gave similar results. Fig.16 shows the plot of PSNR vs compression ratio (which is varied by taking different number of reconstruction planes). This scheme is more computationally efficient that the EBCOT.

Fig. 7. The results of Compression with DWT and Embedded Zerotree coding for compression ratios 50, 5 and 2.5.

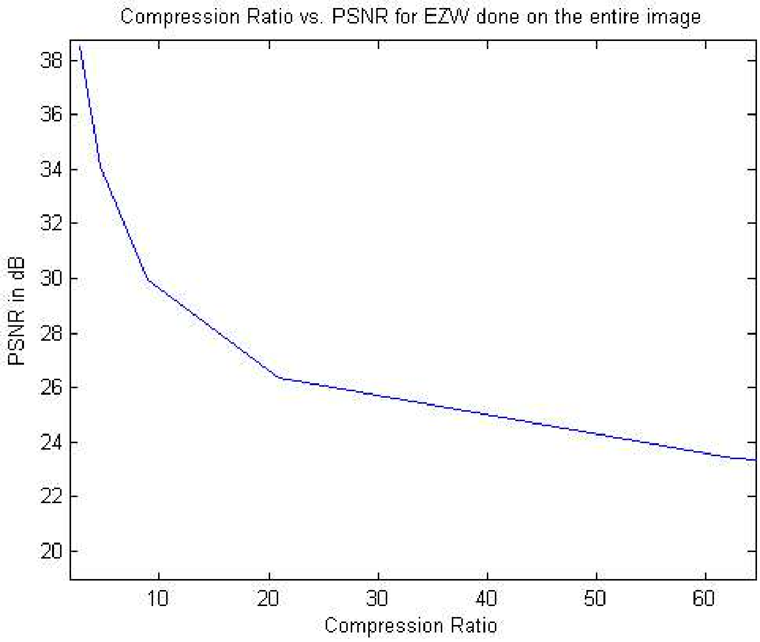

Fig. 8. Compression Ratio vs PSNR graph for Wavelet transform based Embedded Zerotree coding

Bit allocation is an important issue in the implementation of EBCOT. The EBCOT uses iterative methods that are computationally very inefficient to implement bit allocation ideas. In the current implementation, we use different number of reconstruction bit planes to obtain different compression ratios. Alternatively, we could also use a different scheme based on the wavelet transform. We know that a change in the low frequency components affect our final output and a corresponding change in the high frequency components do not. We also try to preserve the most significant bits and hence we must extract more bits from the MSB bit plane and less bits from a LSB bit plane. Fig. 15 shows a typical bit plane. The polygon drawn in the bit plane figure would hence be able to capture more information in the low frequency components and would also give more importance to the most significant bit. This is similar to zigzag coding in 3 dimensions. This scheme also has resolution scalability and increasing the area of

Fig. 9. The results of Compression with DWT and Embedded Zerotree coding for different N = 8, 16 and 32 for compression ratio 4.

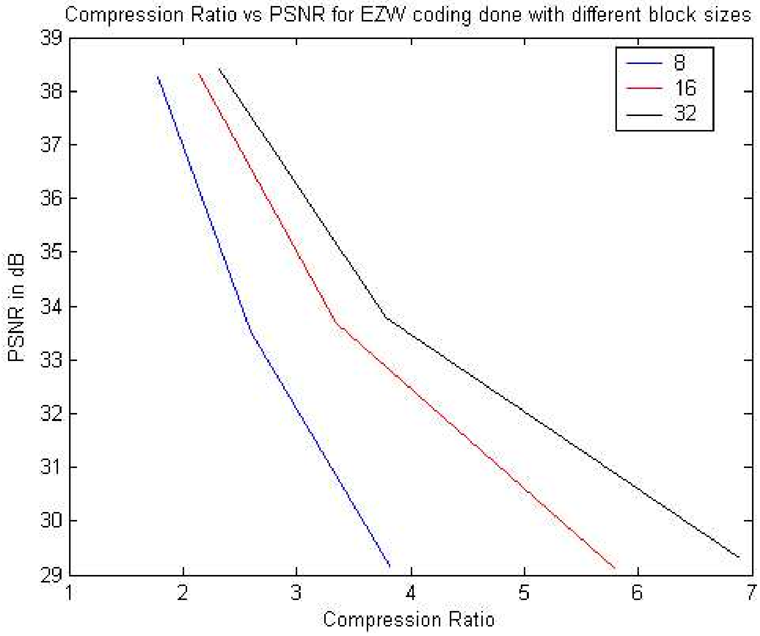

Fig. 10. Compression Ratio vs PSNR graph for Wavelet transform based Embedded Zerotree coding for different block lengths

the polygon would give more resolution. On implementation, this can give better results.

4.6. Block DCT Based JPEG2000 (Annex C and D) Compression

The same concept as described in the last section is used. The transform used is DCT. Block size of 8x8 is used. Fig.19 and Fig. 20 show the degradation in quality as the compression ratio is increased.

Fig. 11. The results of Compression with DWT and SPIHT coding for compression ratios 7, 4 and 2.

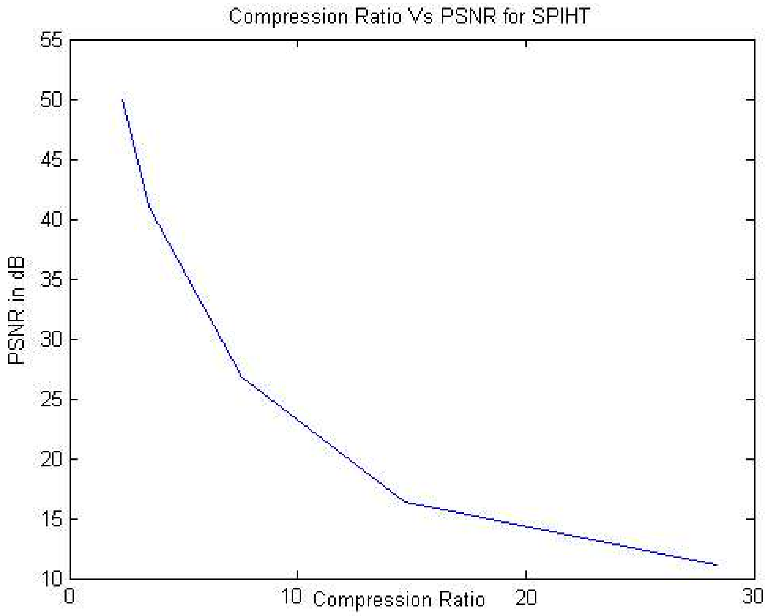

Fig. 12. Compression Ratio vs PSNR graph for Wavelet transform with SPIHT coding

4.7. Full JPEG 2000

The full JPEG2000 image coding standard program (JASPER [22]) was used to generate the compressed images and the compression ratio vs PSNR graph was obtained from the results. These are indicated in Fig. 21 and Fig. 22.

5. RESULTS AND PERFORMANCE COMPARISON

The table 1 gives a comparison the various schemes studied.

Fig. 13. The results of Compression with DWT and SPIHT coding for different N = 8, 16 and 32 for compression ratio 4.

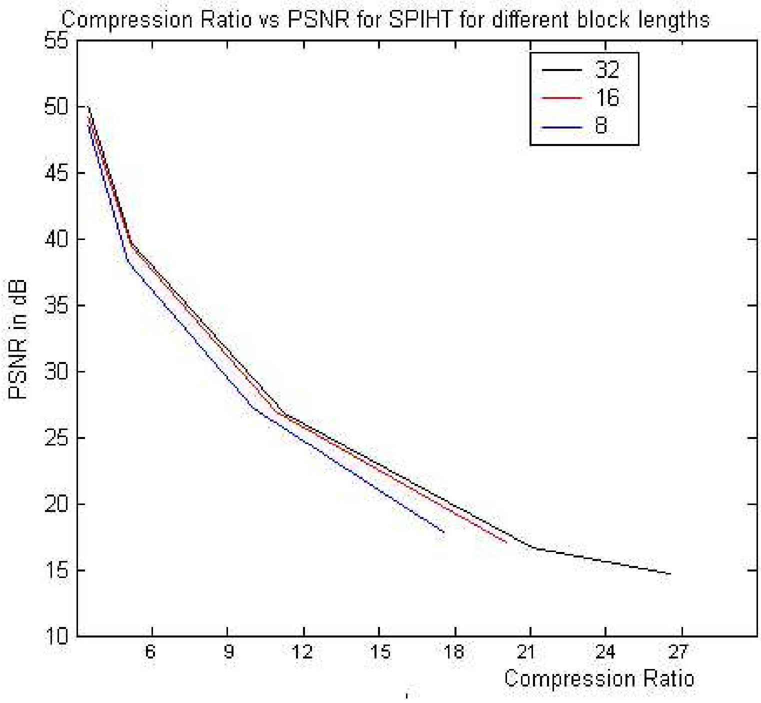

Fig. 14. Compression Ratio vs PSNR graph for Wavelet transform based SPIHT coding for different block lengths

5.1. The Effect of Transforms used

It is observed that with the limitations of our implementation that the Baseline JPEG performs around 3-4 dB better than the DCT-Based Embedded Image coding. The Wavelet based EZW scheme performs around 3 dB (for the same compression ratio) better than the corresponding DCT based version. A performance improvement of around 1-1.5 dB due to wavelet based techniques over DCT based schemes has been shown in various works [5, 8]. This improvement can be attributed to the better energy compaction properties of the wavelet transform compared to the discrete cosine transform. The effect of wavelet filters was also studied. For testing purposes, the EZW algorithm was implemented for various kinds of wavelet filters namely haar, db4 and db7 and the results are tabulated in table 2. The table shows

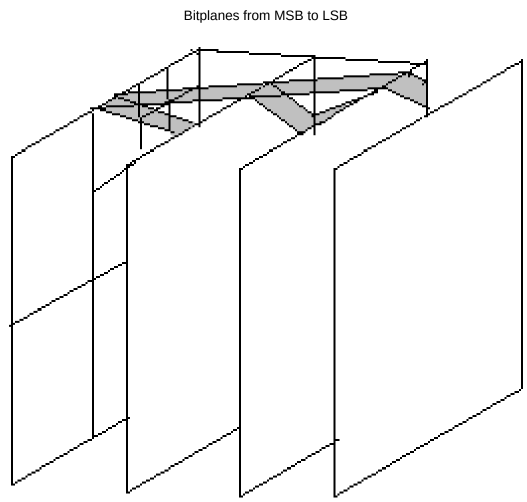

Fig. 15. Bit plane representation of images. The left plane stands for the Most significant bit and the right end stands for the least significant bit plane

Fig. 16. Images compressed with block based JPEG2000 Annex C and D coder for various number of reconstructed bit planes. Compression Ratios 1.43, 1.5 and 4.4

that the changing the filters does not enhance the performance a lot. Our implementation of JPEG2000 (Annex D) was tested for various filters (haar, db1, bi-orthogonal and reverse bi-orthogonal). The results indicate that the PSNR obtained in each of the cases is almost the same for a given compression ratio.

5.2. The Effect of the encoding scheme

For comparison purposes, the results of the SPIHT algorithm were obtained. It was observed that the SPIHT algorithm gave around 2 dB better performance than the our corresponding implementation of the Wavelet based Embedded image coder. This can be attributed to the better compression methods used in SPIHT. The Embedded Block coding algorithm as described in JPEG2000 [18] (Annex D) was implemented. The algorithm was tested for varying compression ratio and block sizes. The algorithm gives comparable results with SPIHT and full-fledged JPEG2000 for compression of 1.4 (PSNR $>$ 50,with

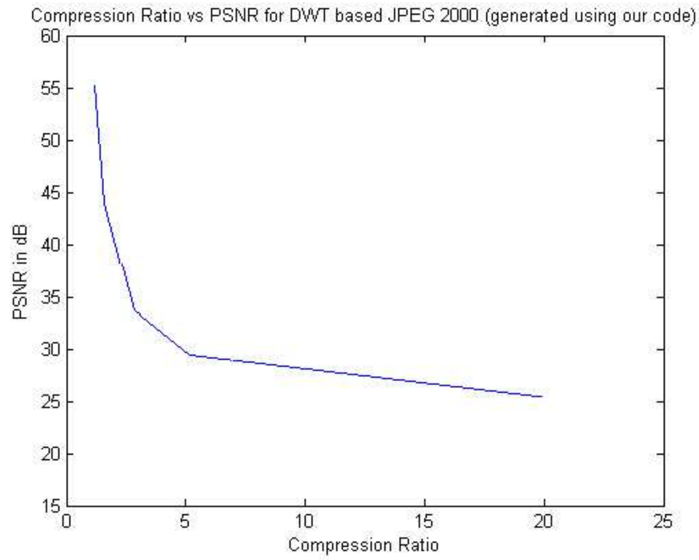

Fig. 17. Compression Ratio vs PSNR graph for JPEG2000 Annex C and D with block size 64

CR	PSNR						
	Basline JPEG	DCT with EZW	DWT with EZW	DWT with SPIHT	DCT with JPEG2000 (imp)	DWT with JPEG2000 (imp)	JPEG2000
2	34	29	35	39	33	37	42
4	31	27	32	35	31	34	35
8	29	23	31	30	27	29	31

Table 1. A Comparison of the PSNR vs Compression Ratio for various schemes

no constraint on bit rate). This confirms the correctness of the implementation. But as the compression ratio is increased, the quality of reconstructed image using our implementation degrades as compared to that of full-fledged JPEG2000. This is due to the fact that in our implementation, we remove bit planes (starting from LSB) to reduce the bit rate, which is not the optimal way. Instead the optimal contribution from each block for each plane should be considered to find which bits should be used for decoding. The high quality (PSNR $>$ 50, compression 1.4) for DCT based implementation too suggests the generic nature of the concept. The JPEG2000 scheme that we implemented does better than the EZW scheme but not quite as good as the SPIHT scheme. This could mainly be because, we have not implemented the full functionality of the scheme.

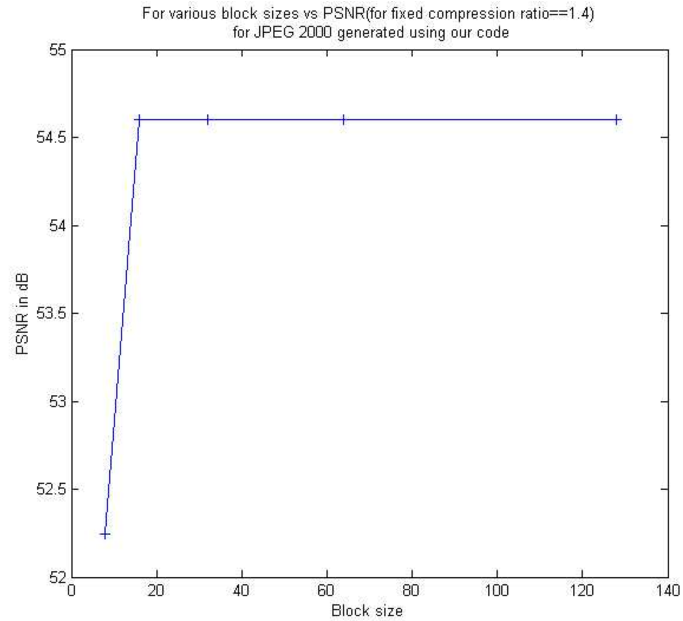

Fig. 18. PSNR vs Block size for JPEG2000 Annex C and D with DWT

Fig. 19. Images compressed with block based JPEG2000 Annex C and D with DCT for various number of reconstructed bit planes. Compression ratios 1.3, 1.5 and 3.4

5.3. Effect of block size

As observed in the previous results section, increasing the block size enhances the performance and one can attain a higher PSNR for the same compression ratio. This effect of block size can be explained by the following argument. When the transform is applied to each NxN block separately instead of the entire image, we would end up storing more number of coefficients. This is because, we would at least end up storing the most significant coefficients in each block. Let us consider an illustration to elaborate this point. Consider an HxW image and lets assume that we are doing a NxN block based implementation on the image. Let L be the number of significant coefficients in every block. We would hence require L(W x H)/(N x N) number of coefficients for the entire image. Here, L would also depend on the block

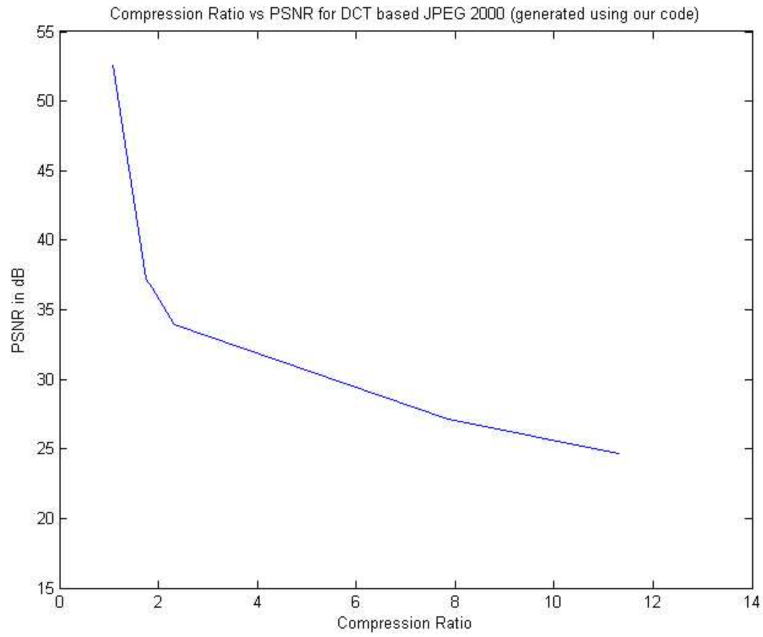

Fig. 20. Compression Ratio vs PSNR for DCT Based JPEG 2000 Annex C and D encoding

Fig. 21. The results of Compression with full fledged JPEG2000 for compression ratios 1, 2 and 11.

size. Increasing the block size would increase the value of L but this increase is less compared to the dramatic decrease when number of blocks are reduced. As far as computational complexity goes, the block based is more computationally efficient compared to the non-block based. For a block based scheme, we would require at least $O(N^2)$ multiplications for each pixel and hence $O(N^4)$ for the every block. Therefore, we require $O(N^4\ WH/\ N^2)$ for the entire image. This reduces to $O(N^2\ WH)$ which is higher when the block size is increased. Hence as far of transform is concerned, we would require the more number of computations for the block based than the corresponding non-block based schemes. Also, as far as coding is concerned, the block based schemes are more computationally efficient than the non-block based schemes. This reduces computational complexity is the price you pay for reduced PSNR of around 3-4 dB for the same compression ratio. The graphs given in the previous section illustrate this

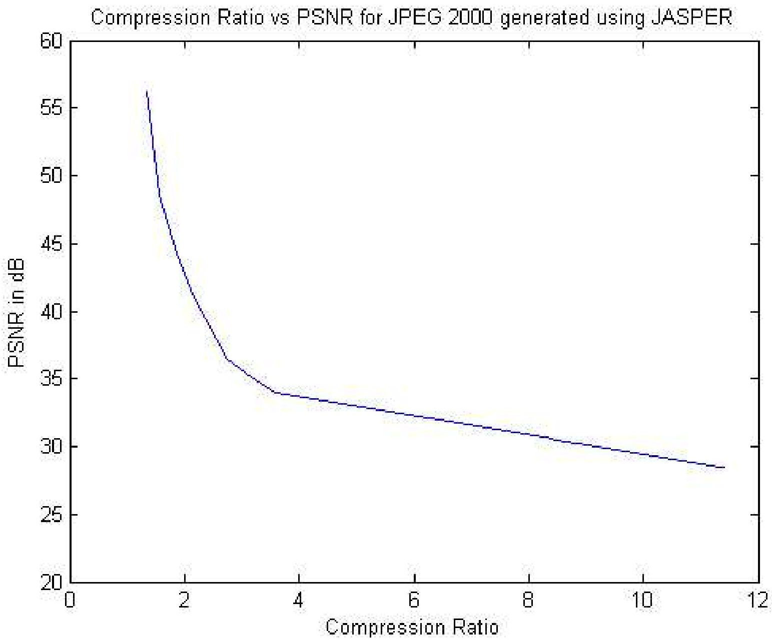

Fig. 22. Compression Ratio vs PSNR graph for full fledged JPEG2000 obtained using the Jasper program

CR	Haar	db4	db7
2	36	35	35.5
4	32	32	32.3
8	30	31	31.4

Table 2. The Effect of Wavelet Filters on DWT implemented with Embedded Zero tree. The table shows the PSNR for the 3 different cases

point. With our implementation of the JPEG2000 Wavelet based scheme, if the block size is greater than 16, there is no improvement in maximum PSNR. The standard also suggests a block size of 32X32 or 64X64.

5.4. Other Comments and Observations

PSNR has traditionally been a very popular metric to test the performance of image compression algorithms. However, Euclidean distance to find the mean square error is not an efficient method to compare images. Measures need to be developed to estimate the visual content of images so that the performance of the algorithms can be better studied. Recently, Wang et. al [19] have proposed a new scheme for visual comparison of images. They have introduced Structural SIMilarity (SSIM) index based on statistics (mean, variance and covariance) of image blocks where the image can be considered to be approximately stationary. This measure was used to compare the image and the results indicate that the resultant compressed images are almost similar with a SSIM index very close to 1.

6. CONCLUSIONS

In this project, we studied the various common schemes used in JPEG. We considered the modular design of the scheme and considered various possible cases. The effect of block based vs non-block based schemes was studied. While the non-block schemes gave better performance they were less computationally efficient. The performance of algorithm with various transforms used was considered. It was observed that the wavelet transform gave a 1-2 dB performance improvement over the DCT due to its better energy compaction properties. Various methods of encoding such as Embedded Zero tree, SPIHT and our implementation of JPEG2000 were considered. A comparative study based on rate scalability, block size, transform filters, computational complexity and rate-distortion tradeoff is also presented.

7. PROGRAMMING

7.1. Baseline JPEG implementation

- JPEG_entropy_decode - Written by Guan-Ming Su. Used to perform entropy decoding (obtained from course web page).
- JPEG_entropy_encode - Written by Guan-Ming Su. Used to perform entropy encoding (obtained from course web page).

7.2. Embedded Zero tree Wavelet coding

- arithenco - inbuilt matlab functions to do arithmetic encoding.
- arithdeco - performs the reverse operation to arithenco.m
- calcPSNR - Used to check the performance for color images.
- entropyEncoder - performs entropy encoding using the arithmetic encoder.
- entropyDecoder - performs entropy decoding using the arithmetic decoder.

- ezwDecoder - implements the algorithm and performs EZW decoding (to be used with ezwEncoder).
- ezwEncoder - implements the encoder. Can be decoded with ezwDecoder.
- ezwScan - to perform scanning.
- findNodeType - finds the type of the node.
- IsZeroTreeRoot - finds if the given element is a zero tree root or not.
- genDWTMatrix - generates the DWT matrix to be given as the input to EZW.
- genImage - performs inverse of genDWTMatrix.
- genIndex - used in coding and decoding.

7.3. SPIHT coding

- arithenco - inbuilt matlab functions to do arithmetic encoding.
- arithdeco - performs the reverse operation to arithenco.m
- calcPSNR - Used to check the performance for color images.
- cSPIHT - obtained from http://www.cs.ru.ac.za/research/students/g01h2306/ to implement SPIHT for comparative studies. Encoder for SPIHT algorithm.
- dSPIHT - obtained from http://www.cs.ru.ac.za/research/students/g01h2306/ to implement SPIHT for comparative studies. Decoder of the SPIHT algorithm.

7.4. DWT and DCT based JPEG 2000 (Annex D) coding

- jpeg2000_dwt - This function calls the encoder and decoder for block based compression with DWT.
- jpeg2000_dct - This function calls the encoder and decoder for block based compression with DCT.
- encode_subband - this function separates the bands into separate blocks and codes them
- encode_code_block - This function calls the various passes(cleanup, significant, refinement) for encoding of bit planes.

- decode_code_block - does inverse of encode_code_block.
- decode_subband - does inverse of encode_subband.
- clean_scan - implements the clean up pass.
- refinement_scan - implements the refinement pass.
- significant_scan - implements the significant pass.
- mq_coder, mq_decoder, fsm, initmq_coder, mq - code obtained online. This code implements MQ coder according to annex C of jpeg2000.

8. REFERENCES

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

[2] Gregory Wallace, "The JPEG Still Picture Compression Standard," *IEEE Transactions on Consumer Electronics*, 1991.

[3] Bryan Usevitch, "A Tutorial on Modern Lossy Wavelet Image Compression : Foundations of JPEG 2000," *IEEE Signal Processing Magazine*, 2001.

[4] Jerome Shapiro "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions in Signal Processing*, 1993.

[5] Zixiang Xiong, Kannan Ramchandran, Michael T. Orchard, and Ya-Qin Zhang, "A Comparative Study of DCT- and Wavelet-Based Image Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 9, No. 5, 1999.

[6] Amir Said, Willian Pearlman, " A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchial Trees," *IEEE Transactions on Circuits and Systems for Video Technology*, 1996.

[7] David Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Transactions on Image Proc.* , 2000.

[8] Zixiang Xiong, Onur Guleryuz, Michael Orchard, " A DCT-Based Embedded Image coder," *IEEE Signal Processing Letters*, 1996.

[9] Marc Antonini, Michel Barlaud, “Image Coding Using Wavelet Transform,” *IEEE Transactions on Image Processing*, 1992.

[10] David Taubman, Michael Marcellin, “JPEG2000: Standard for Interactive Imaging,” *IEEE Proc.*, Vol 90, No. 8, 2002.

[11] Andy Hung, Teresa Meng, “Optimal Quantizer Step Sizes for transform Coders,” *IEEE Proc.*,1991.

[12] David Taubmana, Erik Ordentlichb, Marcelo Weinberger, Gadiel Seroussi, “Embedded block coding in JPEG 2000,” *Signal Processing: Image Communication*, 2002.

[13] Michael W. Marcellina, Margaret A. Lepleyb, Ali Bilgina, Thomas J. Flohrc, Troy T. Chinend, James H. Kasnere “An overview of quantization in JPEG 2000,” *Signal Processing: Image Communication*, 2002.

[14] Diego Santa-Cruz, Rapha.el Grosbois, Touradj Ebrahimi, “JPEG 2000 performance evaluation and assessment,” *Signal Processing: Image Communication*, 2002.

[15] Diego Santa-Cruz and Touradj Ebrahimi “A Study of JPEG 2000 still Image Coding Versus Other standards,” *European Signal Processing Conference*, Finland, 2000.

[16] D. Santa-Cruza, T. Ebrahimia, J. Askelfb, M. Larssonb and C. A. Christopoulosb, “JPEG 2000 still image coding versus other standards,” *http://ltswww.epfl.ch/ dsanta/ for the final reference*.

[17] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Transactions on Image Processing*, Vol. 9, 2000.

[18] “JPEG 2000 Part I Final Committee Draft Version 1.0,” *ISO/IEC JTC 1/SC 29/WG 1 N1646R, April 2000*.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, Vol. 13, no. 4, 2004.

[20] A. K. Jain, “Fundamentals of Digital Image Processing,” *Pentice Hall* , 2000.

[21] R. C. Gonzalez, R. E. Woods, “Digital Image Processing,” *Pearson Education*, II Edition, 2003.

[22] http://www.ece.uvic.ca/ mdadams/jasper/