

ENEE631 Project 1

Wavelet-Based Image Compression

Naotoshi Seo, sonots@umd.edu

April 11, 2007

1 Introduction

Wavelet-based image coding has gone through significant advancement since the DCT-based codec became the first JPEG standard. In this project, we explore the wavelet-based image coding and examine major factors that affect the coding performance. The Embedded Zero-Tree Wavelet (EZW) codec for image compression is implemented. In addition, Block-based EZW coding is implemented and comparison between different block sizes are performed, furthermore, the Block-based EZW coding is compared with the JPEG-like code which was implemented in assignment 3. SPIHT based compression is also experimented.

2 Literature Survey

2.1 EZW

The embedded zerotree wavelet (EZW) encode is based on progressive encoding to compress an image into a bit stream with increasing accuracy. This means that when more bits are added to the stream, the decoded image will contain more detail. Coding an image with EZW scheme, an encoder can terminate the encoding at any point thereby allowing a target rate or target distortion metric to be met exactly. Also, given a bit stream, the decoder can cease decoding at any point in the bit stream and still produce exactly the same image that would have been encoded at the bit rate corresponding to the truncated bit stream.

The EZW algorithm is based on four key concepts: 1) a discrete wavelet transform or hierarchical subband decomposition, 2) prediction of the absence of significant information across scales by exploiting the self-similarity inherent in images, 3) entropy-coded successive approximation quantization, and 4) universal lossless data compression which is achieved via adaptive arithmetic coding [3].

2.2 The Wavelet Transform

Wavelet based techniques for image compression have been increasingly used for image compression. The main contribution of wavelet theory is that it provides an elegant framework in which both "anomalies," or signal behavior that is more localized in the time or space domain and tends to be wide band in the frequency domain, and "trends," or signal behavior that is more localized in frequency but persists over a large number of lags in the time domain. Wavelets provide a signal representation in which some of the coefficients represent long data lags corresponding to a narrow band, low frequency range, and some of the coefficients represent short data lags corresponding to a wide band, high frequency range. Using the concept of *scale*, data representing a continuous tradeoff between time (or space in the case of images) and frequency is available.

A comparative study of DCT and wavelet based image coding can be found in [2].

2.3 SPIHT

The Set Partitioning in Hierarchical Trees (SPIHT) algorithm is an extension of the EZW scheme, proposed by A. Said and W. Pearlman in [4]. SPIHT improves the basic zerotree coding algorithm of EZW by using slightly different tree structures and allowing the zerotree symbols to be generated in more cases (i.e., for combinations of parallel zerotrees). SPIHT achieves more efficient significance map coding because of the set partitioning algorithm (sophisticated method of stepping through and partitioning the sets of trees and coefficients shared by encoder and decoder).

2.4 EBCOT

The Embedded Block Coding with Optimized Truncation (EBCOT) is a scheme which JPEG2000 uses, introduced by David Taubman [6]. The EBCOT algorithm is related to much earlier work such as EZW, SPIHT. Like each of these, the EBCOT algorithm uses a Wavelet transform to generate the subband samples which to be quantized and coded. Like each of these, the EBCOT also generates scalable compressed bit-streams; however, whereas its predecessors typically exhibited one (SNR) or at most two (SNR and resolution) dimensions of scalability, the EBCOT algorithm produces bit-streams which are SNR and resolution scalable and which also have a random access attribute whereby independent portions of the bit-stream correspond to different spatial regions of the image.

A key advantage of scalable compression is that the target bit-rate or reconstruction resolution need not be known at the time of compression. Rather than focusing on generating a single scalable bit-stream to represent the entire image, EBCOT partitions each subband into relatively small blocks of samples and generates a separate highly scalable bit-stream to represent each so-called code-block [6].

3 Experiments and Results

3.1 EZW

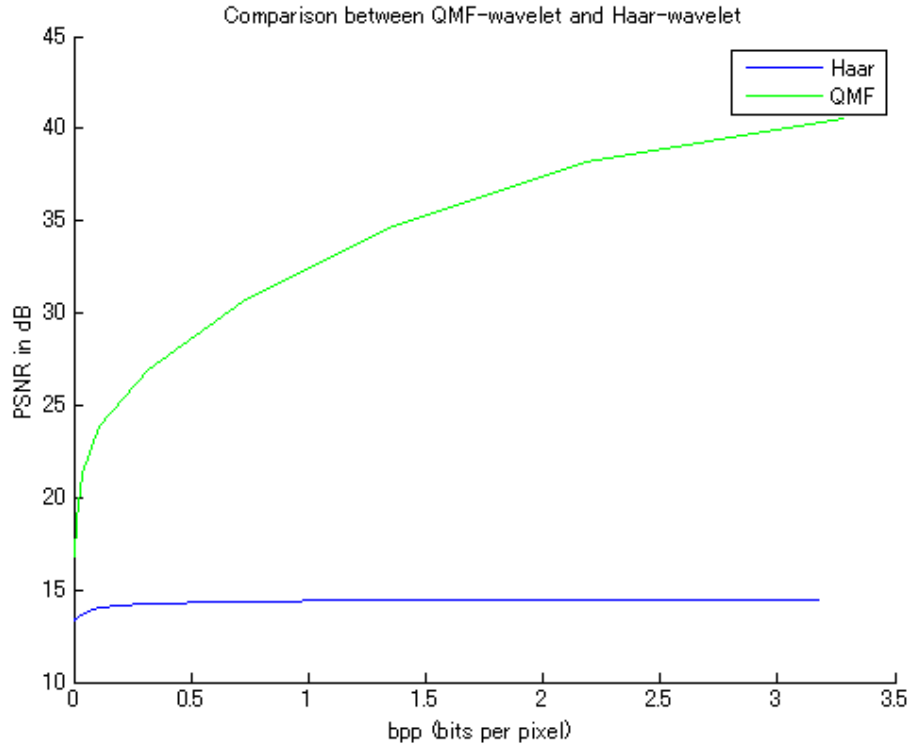
EZW-based Image Compression was implemented. The details of this scheme is available on Shapiro's paper [3]. Huffman encoding was used for entropy encoding. Comparison between a quadrature mirror filter (QMF)-wavelet and a haar-wavelet was performed. As the result, a QMF-wavelet was used.



Figure 1: (a) Decoded image at fourth pass, $\text{bpp} = 0.0387$, $\text{PSNR} = 21.2866$ dB . (b) Decoded image at sixth pass, $\text{bpp} = 0.3179$, $\text{PSNR} = 26.9341$ dB . (c) Decoded image at eighth pass, $\text{bpp} = 1.3424$, $\text{PSNR} = 34.6630$ dB .

Comparison between QMF-wavelet and haar-wavelet

EZW compression using a haar-wavelet and a quadrature mirror filter (QMF)-wavelet were compared. As shown in the next figure, a QMF-wavelet is completely better than a haar-wavelet which achieved only pretty low (bad) PSNR.



3.2 Block-EZW

Block based EZW coding was also implemented. As a way of implementation, a delimiter symbol was inserted between each block's symbol table generated at the dominant and the subordinate pass, and these symbol tables were connected and encoded at burst using huffman encoding. Below pictures show the compressed image at the 1st, 2nd, 3rd pass with block size $M = 8$.



Figure 2: Block-EZW with block size 32. (a) Decoded image at 1st pass, $\text{bpp} = 0.0658$, $\text{PSNR} = 21.0563$ dB . (b) Decoded image at 3rd pass, $\text{bpp} = 0.4858$, $\text{PSNR} = 27.1965$ dB . (c) Decoded image at 5th pass, $\text{bpp} = 1.7988$, $\text{PSNR} = 34.3224$ dB .

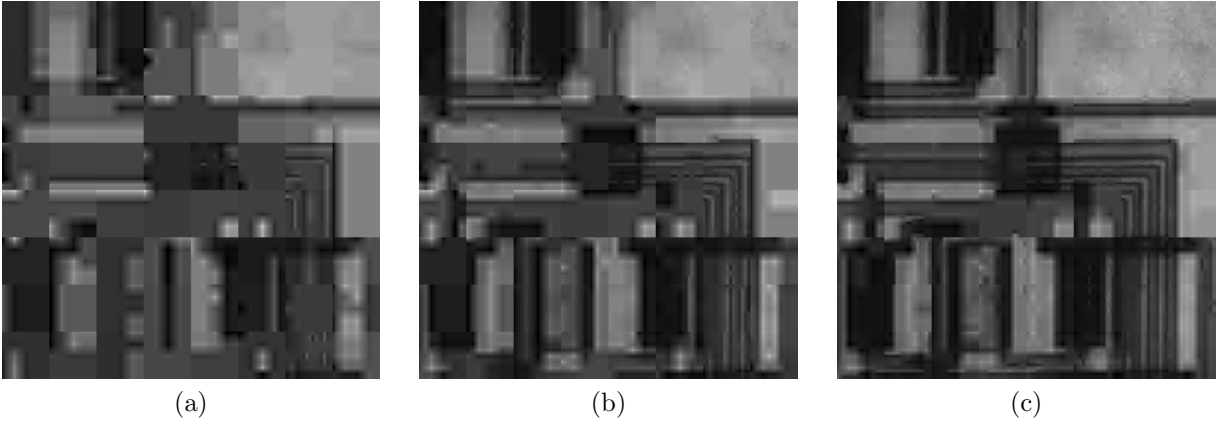
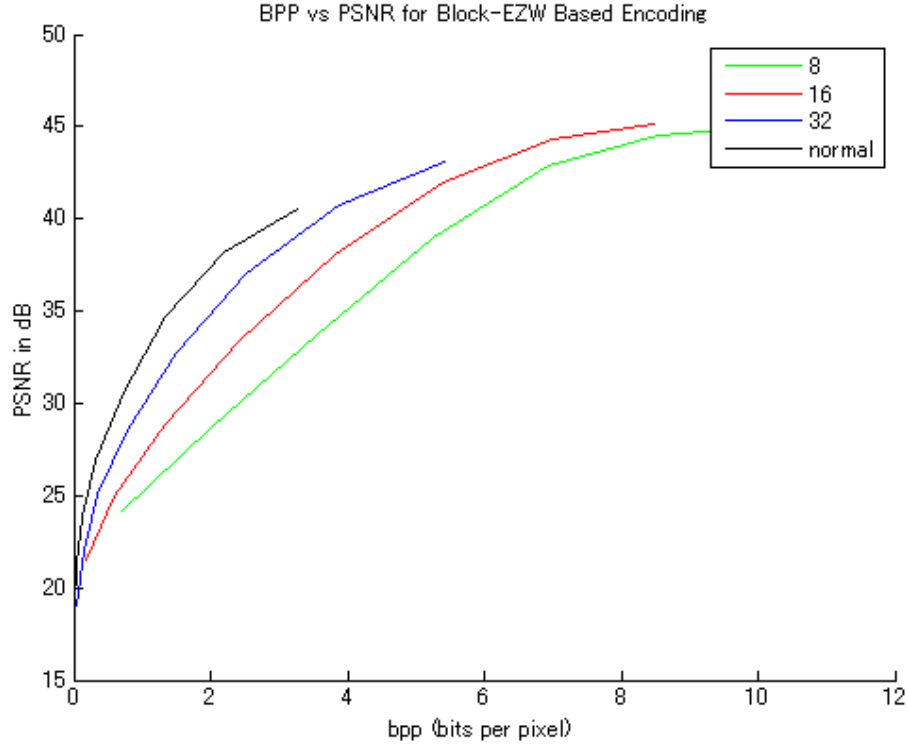


Figure 3: Block-EZW with block size 32. (a) Decoded image at 1st pass, $\text{bpp} = 0.7136$, $\text{PSNR} = 24.1599$ dB . (b) Decoded image at 2nd pass, $\text{bpp} = 1.9872$, $\text{PSNR} = 28.6582$ dB . (c) Decoded image at 3rd pass, $\text{bpp} = 3.5583$, $\text{PSNR} = 33.6806$ dB .

Comparison of different block sizes

Below figure shows us that larger block size achieved better (less) bits per pixel for the same PSNR. This is because EZW encoding is more efficient for larger block (one zerotree symbol represents many coefficients). Notice: smaller block size version also requires more computation for one pass, but requires fewer number of passes to achieve the same PSNR, totally computation time could become almost same. In conclusion, the non-Block based default EZW encoding works better.



Comparison between block size 8, 16, 32 and non-block version of EZW.

Comparison between Block-EZW and JPEG-like codec

The block size of the JPEG-like codec assignment 3 is fixed 8, thus we compare them with the block size 8. Block-EZW requires pretty much bpp even in the first pass when the block size is 8. Because of this reason, Block-EZW with block size 8 was pretty worse than the JPEG-like codec.

The JPEG-like codec uses DCT transform. In the case of DCT, the block size compensates tradeoffs between the spatial resolution and the frequency resolution. However, in the case of Block-EZW, it resulted in only disadvantages. This is because EZW uses wavelet transform which originally provides an elegant framework in which both anomalies and trends. [3].

3.3 SPIHT

SPIHT Matlab code [8] library was used. SPIHT did not have fine adjustments of target bpp, the least bpp was 0.4377 and the most bpp was 7.5886. Even if $\text{bpp} = 0.01$ is specified, the resulted bpp was 0.4377, and even if $\text{bpp} = 15$ is specified, the resulted bpp was 7.5886.

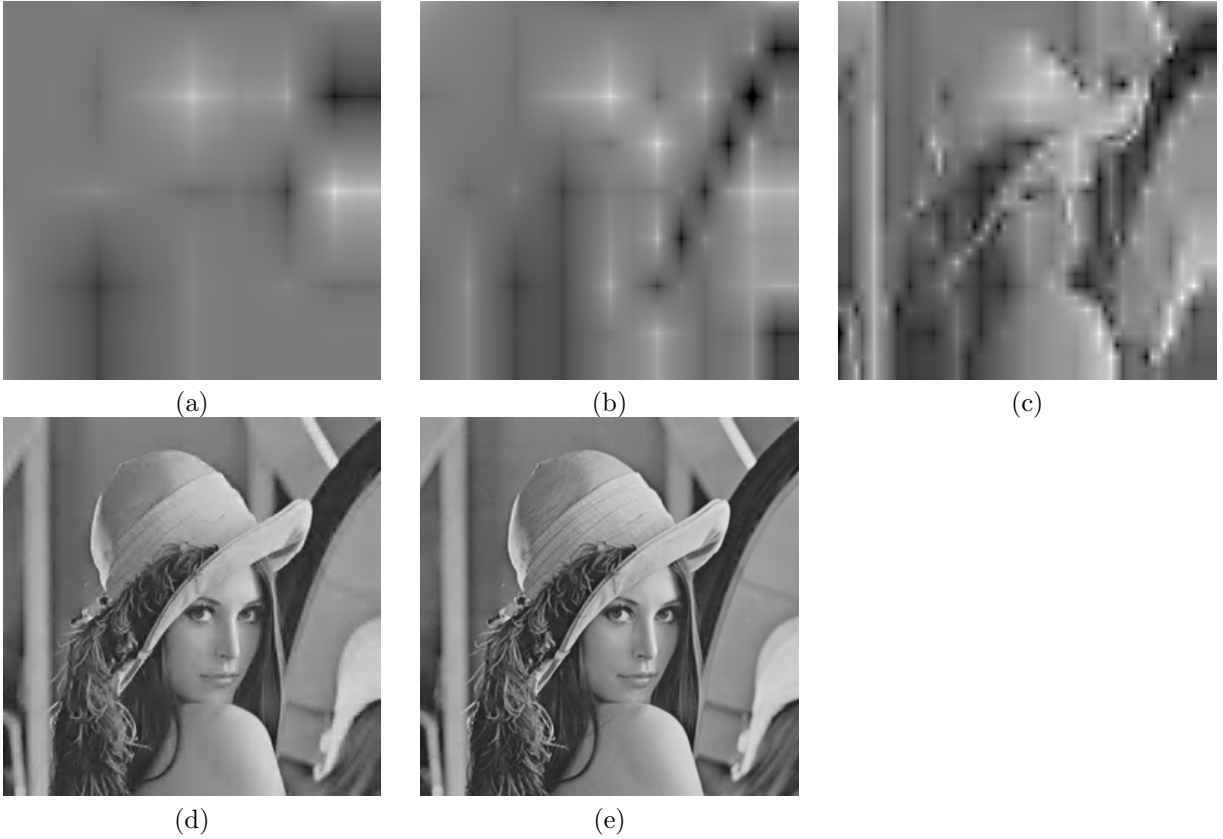


Figure 4: SPIHT. (a) Decoded image at 1st pass, $\text{bpp} = 0.4377$, $\text{PSNR} = 15.3562$ dB . (b) Decoded image at 2nd pass, $\text{bpp} = 0.8754$, $\text{PSNR} = 16.4917$ dB . (c) Decoded image at 3rd pass, $\text{bpp} = 1.7541$, $\text{PSNR} = 20.7105$ dB . (d) Decoded image at 4th pass, $\text{bpp} = 3.7656$, $\text{PSNR} = 33.3704$ dB . (e) Decoded image at 5th pass, $\text{bpp} = 7.5886$, $\text{PSNR} = 45.3426$ dB .

4 Lists of codes

4.1 EZW

- `ezw.m` - EZW encoding
- `ezw_dominantpass.m` - performs dominant pass of ezw encoding, used by `ezw.m`
- `ezw_subordiantepass.m` - performs subordinate pass of ezw encoding, used by `ezw.m`
- `ezw_mortonorder.m` - generate morton scanning order matrix, used by ezw family.
- `ezw_childtree.m` - generate ezw child-tree mask for given coefficient position, used by ezw family.
- `iezw.m` - EZW decoding
- `iezw_dominantpass.m` - Inverse dominant pass
- `iezw_subordinatepass.m` - Inverse subordiante pass
- `demo_ezw.m` - demo

4.2 Huffman Encoding

- huffman.m - huffman encoding
- ihuffman.m - huffman decoding
- demo_huffman.m - demo

4.3 Wavelet

matlabPyrTools [7] library was used.

- wave_transform.m - haar wavelet transform
- wave_transform_qmf.m - qmf-wavelet transform
- invwave_transform.m - inversed haar-wavelet transform
- invwave_transform_qmf.m - inverse qmf-wavelet transform
- demo_wavelet.m - demo

4.4 SPIHT

SPIHT Matlab code [8] library was used.

4.5 Compress - Decompress

- compress_ezw.m - EZW based compression (with qmf-wavelet and huffman encoding)
- decompress_ezw.m - EZW based decompression
- compress_block_ezw.m - Block-EZW based compression
- decompress_block_ezw.m - Block-EZW based decompression
- compress_spiht.m - SPIHT based compression
- decompress_spiht.m - SPIHT based decompression

4.6 Small Utilities

- imread.m - read an image as gray image
- psnr.m - compute peak signal to noise ratio
- bytes.m - obtain the number of bytes of an object

Bibliography

- [1] G. K. Wallace, "The JPEG still picture compression standard", *IEEE Trans. on Consumer Electronics*, vol. 38, no. 1, pp18-34, Feb 1992

- [2] Zixiang Xiong, Kannan Ramchandran, Michael T. Orchard, and Ya-Qin Zhang, "A Comparative Study of DCT- and Wavelet-Based Image Coding", *IEEE Trans. on Circuits and Systems for Video technology*, vol. 9, no. 5, pp 692-695, August 1999.
- [3] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp 3445-3462, 1993.
- [4] Amir Said and William A. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, pp. 243-250, June 1996.
- [5] Bryan E. Usevitch, "A Tutorial on Modern Lossy wavelet Image Compression: Foundations of JPEG 2000", *IEEE Signal Processing Magazine*, pp.22-35, Sep 2001.
- [6] D. S. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. on Image Processing*, pp. 1158-1170, 2000.
- [7] Eero Simoncelli, "matlabPyrTools - Image and Multi-scale Pyramid Tools," <http://www.cns.nyu.edu/~eero/software.html>
- [8] Paul Heideman, "SPIHT Matlab code," <http://www.cs.ru.ac.za/research/students/g01h2306/>.