International Journal of Information Technology and Knowledge Management
January June 2009, Volume 2, No. 1, pp. 115-118

STILL IMAGE COMPRESSION USING EMBEDDED ZEROTREE WAVELET ENCODING

V. S. Shingate*, T. R. Sontakke & S. N. Talbar*****

This paper proposes a **Embedded Zerotree Wavelet Encoding (EZW)** for image compression.

Shapiro's *Embedded Zerotree Wavelet* encoder or *EZW* encoder for short [Sha93]. An EZW encoder is an encoder specially designed to use with *wavelet transforms*, which explains why it has the word wavelet in its name. The EZW encoder was originally designed to operate on images (2D-signals) but it can also be used on other dimensional signals. The EZW encoder 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, a property similar to JPEG encoded images. It is also similar to the representation of a number like Π. Every digit we add increases the accuracy of the number, but we can stop at any accuracy we like. Progressive encoding is also known as *embedded encoding.* Coding an image using the EZW scheme, together with some optimizations results in a remarkably effective image compressor with the property that the compressed data stream can have *any* bit rate desired. *Any* bit rate is only possible if there is information loss somewhere so that the compressor is *lossy*. However, lossless compression is also possible with an EZW encoder, but of course with less spectacular results.

Index Terms: Entropy coding, wavelet based image compression, embedded image coding, zero trees

1. Introduction

Image compression is becoming ubiquitous in many application areas as diverse as web browsing, multimedia database, digital still cameras, printers, and scanners. They are supported on many different hardware as well as software platforms such as PCs, Unix Workstations, embedded systems such as DSPs, and micro-controllers. All these different applications and appliances impose different constraints and requirements for the image coding algorithm. On the other hand, it is highly preferred that one single image coding algorithm can meet the requirements of the majority of these applications so that interoperability can be guaranteed. These conflicting requirements provide a big challenge for image coding research. In the last a few years, a lot of progresses have been made in image coding algorithms, especially with the introduction of wavelet based methods [1,2,3,4]. However, the majority of the attention has been paid to improving coding efficiency of the image compression algorithms. While coding efficiency is an very important factor in judging an image coding algorithm, even more important for practical applications, is to evaluate the balance between coding efficiency, implementation complexity, and features such as spatial and quality scalability, progressive transmission, random accesses, and error resilience. This paper is an attempt in that direction.

A major breakthrough was achieved by Shapiro — J.M. Shapiro, "**Embedded Image Coding using Zerotrees of Wavelet Coefficients**". IEEE Trans. on Signal Processing Dec 1993. He realized that there would be correlations between information about an image at different resolution levels.

"If the co-efficient of a wavelet at one scale in part of the image is not significant (i.e. close to zero), then the higher resolution wavelet coefficients in the same part of the image are also likely to be insignificant."

This in turn implies that we should output coefficients for large scales before coefficients for smaller scales. Significance is relative to a threshold value T. Shapiro devised a multi-pass scheme where T is successively halved, with additional information being output in each pass.

The scanning order in conventional transform codec like JPEG can be regarded as *vertical* scanning since it encodes each coefficient completely before proceeding to the next coefficient, in both lossy and lossless mode. The Embedded Zerotree Wavelet (EZW) [2] codes each bit-plane successively to give an embedding property by *horizontal* scanning every bit-plane of the block 's during scanning the coefficients in both bit-by-bit and coefficient-by-coefficient manners. Below discussion explains the zerotree structure, scanning procedure and Algorithm in detail.

* Senior Lecturer in Electronics Dept, K.B.P. College of Engineering, Satara. *E-Mail: vsshingate@yahoo.co.in*

** Principal, S.G.G.S. College of Engg. & Technology, Nanded.

*** Head of Electronics Dept, Babasaheb Ambedkar, Technical University, Lonere

2. The Zerotree

The EZW encoder is based on two important observations:

(1) Natural images in general have a low pass spectrum. When an image is wavelet transformed the energy in the subbands decreases as the scale decreases (low scale means high resolution), so the wavelet coefficients will, average, be smaller in the higher subbands than in the lower subbands. This shows that progressive encoding is a very natural choice for compressing wavelet transformed images, since the higher subbands only add detail.

(2) Large wavelet coefficients are more important than smaller wavelet coefficients.

These two observations are exploited by the EZW encoding scheme by coding the coefficients in decreasing order, in several passes. For every pass a threshold is chosen against which all the coefficients are measured. If a wavelet coefficient is larger than the threshold it is encoded and removed from the image, if it is smaller it is left for the next pass.

When all the wavelet coefficients have been visited the threshold is lowered and the image is scanned again to add more detail to the already encoded image. This process is repeated until all the wavelet coefficients have been encoded completely or another criterion has been satisfied (maximum bit rate for instance). The trick is now to use the dependency between the wavelet coefficients across different scales to efficiently encode large parts of the image, which are below the current threshold. It is here where the ***zerotree*** enters.

Wavelet transform transforms a signal from the time domain to the joint time-scale domain. This means that the wavelet coefficients are two-dimensional. If we want to compress the transformed signal we have to code not only the coefficient values, but also their position in time. When the signal is an image then the position in time is better expressed as the position in space. After wavelet transforming an image we can represent it using trees because of the subsampling that is performed in the transform. A coefficient in a low subband can be thought of as having four descendants in the next higher subband (see figure 1). The four descendants each also have four descendants in the next higher subband and we see a quad-tree emerge: every root has four leafs.

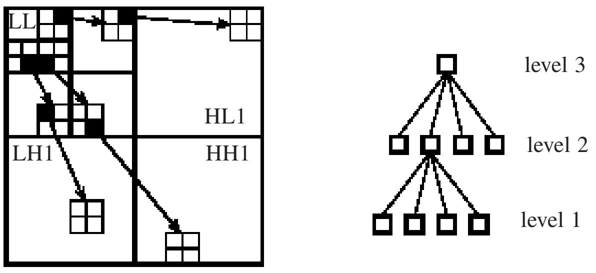


Figure 1: The Relation between Wavelet Coefficients in Subbands as Quad Tree

We can now give a definition of the ***zerotree***. A zerotree is a quad-tree of which all nodes are equal to or smaller than the root. The tree is coded with a single symbol and reconstructed by the decoder as a quad-tree filled with zeroes. To clutter this definition we have to add that the root has to be smaller than the threshold against which the wavelet coefficients are currently being measured.

The EZW encoder exploits the zerotree based on the observation that wavelet coefficients decrease with scale. It assumes that there will be a very high probability that all the coefficients in a quad tree will be smaller than a certain threshold if the root is smaller than this threshold. If this is the case then the whole tree can be coded with a single zerotree symbol. Now if the image is scanned in a predefined order, going from high scale to low, implicitly many positions are coded through the use of zerotree symbols. Of course the zerotree rule will be violated often, but as it turns out in practice, the probability is still very high in general. The price to pay is the addition of the zerotree symbol to our code alphabet.

3. Working

Now that we have all the terms defined we can start compressing. Lets begin with the encoding of the coefficients in decreasing order.

A very direct approach is to simply transmit the values of the coefficients in decreasing order, but this is not very efficient. This way a lot of bits are spend on the coefficient values and we do not use the fact that we know that the coefficients are in decreasing order.

A better approach is to use a threshold and only signal to the decoder if the values are larger or smaller than the threshold. If we also transmit the threshold to the decoder, it can reconstruct already quite a lot. To arrive at a perfect reconstruction we repeat the process after lowering the threshold, until the threshold has become smaller than the smallest coefficient we wanted to transmit. We can make this process much more efficient by subtracting the threshold from the values that were larger than the threshold. This results in a bit stream with increasing accuracy and which can be perfectly reconstructed by the decoder.

If we use a predetermined sequence of thresholds then we do not have to transmit them to the decoder and thus save us some bandwidth. If the predetermined sequence is a sequence of powers of two it is called bitplane coding since the thresholds in this case correspond to the bits in the binary representation of the coefficients. EZW encoding as described in [Sha93] uses this type of coefficient value encoding.

One important thing is however still missing: the transmission of the coefficient positions. Indeed, without this information the decoder will not be able to reconstruct the encoded signal (although it can perfectly reconstruct the transmitted bit stream). It is in the encoding of the positions where the efficient encoders are separated from the inefficient ones. As mentioned before, EZW encoding uses a predefined scan order to encode the position of the wavelet coefficients (see figure 2). Through the use of zerotrees many positions are encoded implicitly. Several scan orders are possible (see figure 3), as long as the lower subbands are completely scanned before going on to the higher subbands. In [Sha93] a raster scan order is used, while in [Alg95] some other scan orders are mentioned. The scan order seems to be of some influence of the final compression result.

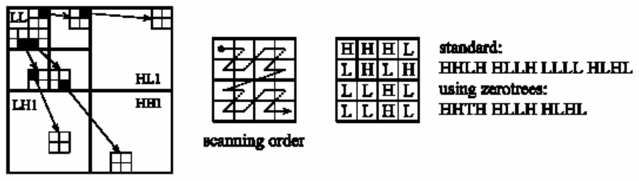
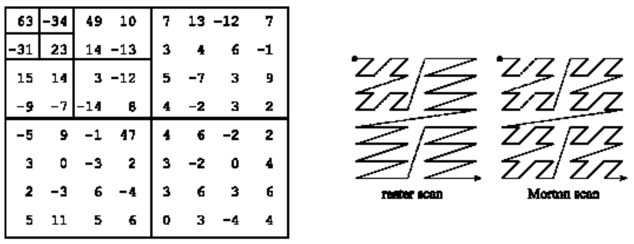


Figure 2: The Relations between Wavelet Coefficients in Different Subbands (Left), How to Scan them (Upper Right) and the Result of using Zerotree (Lower Right) Symbols (T) in the Coding Process. An H Means that the Coefficient is Higher than the Threshold and an L Means that it is Below the Threshold. the Zerotree Symbol (T) Replaces the Four L's in the Lowerleft Part and the L in the Upper Left Part



63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Figure 3: Scanning Order Using Morton Scan

There are two types of scanning procedures namely

(1) Raster scan

(2) Mortan scan

The difference in approach between the two procedures is illustrated by the following diagram.

In our implemented method we have used Mortan scan, which is more accurate and produces standard results.

4. Algorithm

(1) Image is read from the file and converted to gray scale.

(2) Wavelet transform is applied to the gray scaled image according to the size of image using

Decomposition order = log2 (size (original image)) (1)

(3) Generate wavelet coefficients

(4) Convert wavelet coefficients to matrix format

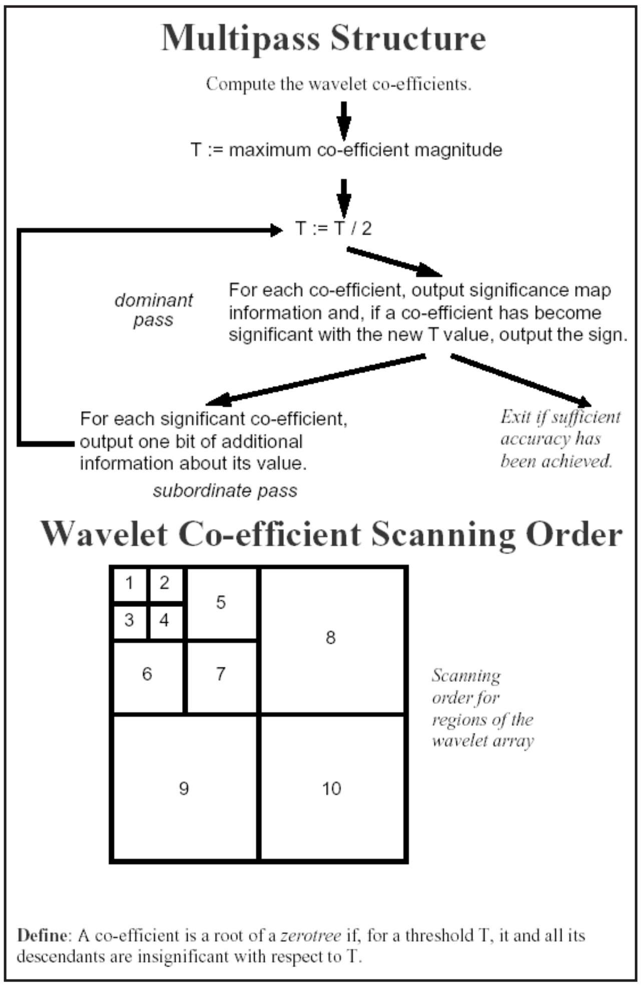
(5) Encode wavelet coefficients using EZW algorithm – Morton scan is used for scanning wavelet coefficients.

(6) Encoding stops when final threshold value is achieved, the procedure is discussed in below diagram

Huffman coding algorithm is applied to the ezw bit stream to achieve the final compressed image.

For reconstruction of the compressed image to obtain the original image inverse of the above mentioned steps is applied.

To implement above algorithm we have used – 'matlab v6.5 & wavelet toolbox'


Multipass Structure
Compute the wavelet co-efficients.
T := maximum co-efficient magnitude
T := T / 2
dominant pass
For each co-efficient, output significance map information and, if a co-efficient has become significant with the new T value, output the sign.
For each significant co-efficient, output one bit of additional information about its value.
subordinate pass
Exit if sufficient accuracy has been achieved.
Wavelet Co-efficient Scanning Order
1
2
3
4
5
6
7
8
9
10
Scanning order for regions of the wavelet array

Define: A co-efficient is a root of a *zerotree* if, for a threshold T, it and all its descendants are insignificant with respect to T.

5. Results

In the experiment the original image 'lenna.bmp' having size 256 × 256 (65536 Bytes). Firstly original image is applied to the compression program, EZW encoded image is obtain, which is further compressed using entropy coding. To reconstruct compressed image, compressed image is applied to decompression program, by which EZW decoded image is obtained, which is given to smoothing program to smoothen EZW decoded image. Compression Ratio, PSNR and SNR, are obtained for the original and reconstructed images. Resulted images are shown below.

Results for the image Cameraman.Tif for various Thresholds is given below:

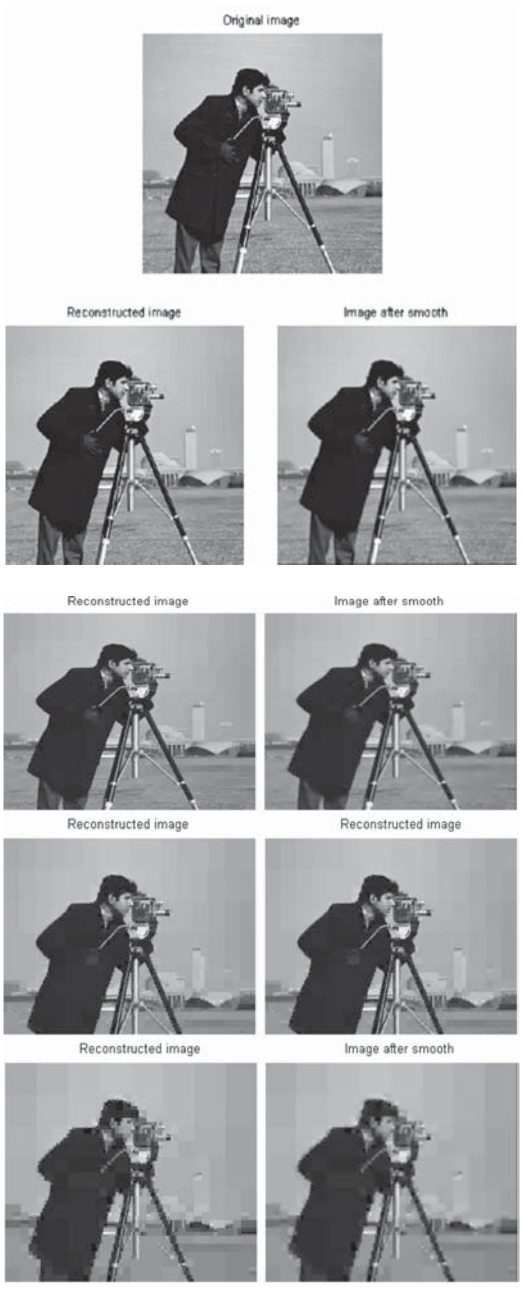


Table 1

Image:	*1*	*2*	*3*	*4*	*5*
Parameter	*TH.=0.06*	*TH.=0.1*	*TH.=0.13*	*TH.=0.3*	*TH.=0.6*
Compression Ratio	5.57	10.37	21.77	52.19	151.36
PSNR	35.55	30.75	26.76	23.18	20.30
Bpp	1.43	0.77	0.36	0.15	0.05
Compression Time	182.62	102.21	64.59	43.34	35.47
Reconstruction Time	448.86	272.37	148.89	78.54	37.18
File Size Original	65240	65240	65240	65240	65240
File Size Compressed	11697	6287	2996	1250	431

6. Conclusion

Implementing Embedded Zerotree Wavelet (EZW) for the compression of still images, far better results are obtained as compare to previous methods. This approach utilizes zerotree structure of wavelet coefficients very effectively, which results in higher compression ratio and better PSNR and SNR. Simulation results have proved that the EZW provides significant performance improvement compared with previous coders.

References

[1] [Alg95] Algazi, V. R. and R. R. Estes, Analysis based Coding of Image Transform and Subband Coefficients. *Proceedings of the SPIE*, **2564** (1995), 11–21.

[2] [Cre97] Creusere, C. D., A New Method of Robust Image Compression based on the Embedded Zerotree Wavelet Algorithm. *IEEE Transactions on Image Processing*, **6**, (10) (1997), 1436–1442.

[3] [Sha93] Shapiro, J. M., Embedded Image Coding Using Zerotrees of Wavelet Coefficients. *IEEE Transactions on Signal Processing*, **41**, (12) (1993), 3445–3462.