International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.2, Issue No.6, pp : 532-538 1 June 2013

EZW Algorithm and Computation of Its Coefficients for Image Compression by Using "Bottom-Up" Approach

M. Mesbahuddin Sarker
Institute of Information Technology, Jahangirnagar University, Bangladesh
E-mail: mail.sarker@yahoo.com

ABSTRACT

Different coding standards for image and video are already in existence such as JPEG for still image and MPEG for video coding. Both the standards utilize block based DCT (discrete cosine transform) technique to remove redundancy. But the problem with DCT approach is that at high compression the decoded image or video suffers from blocking artifacts. To avoid this problem, wavelet transformation is applied over the entire image or image frame. This paper presents a Bottom-Up algorithm by using EZW (Embedded Zerotree Wavelet) transformation technique for getting best compression ratio while maintaining the same quality. Experimental results demonstrate that the method is fast, robust and efficient enough to implement it in still and complex images with significant image compression.

Key Words: Embedded Zerotree Wavelet (EZW), Coding and Decoding, Quantization.

1. INTRODUCTION

Over the last few years, a substantial development is going on in the filed of managing large amounts of electronically stored image data mainly due to the interest of building multimedia information systems for database community and image database systems for computer vision community. Managing image data in this regard requires storage, retrieval and processing of pictorial entities. The convergence of database and image processing/pattern recognition technology yields the basis for the creation of such digital image archives. Moreover, with the growth and popularity of the world wide web (www), a tremendous amount of visual information is made accessible publicly. As a consequence, there is a growing demand for compression methods storage, retrieval and processing of pictorial entities from large image archives [David].

A fair amount of research work has been published in literature on image compression techniques using zerotree wavelet transform. The embedded zerotree wavelet (EZW) image compression algorithm proposed by Shapiro [Rabi, Shapiro] is a very efficient wavelet-based image coding algorithm [Shufang]. Its distinctive feature is that it uses a novel data structure of zerotree to efficiently represent zeros across different scales but at the same spatial location, resulting in the compact description of the significance map. 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 details, a property similar to JPEG encoded images [JPEG, Othman et. al.].

Coding an image using the ZEW scheme, together with some optimization, 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 [Lossless, Singara et. al.] compression is also possible with an EZW encoder, but of course with less spectacular results.

2. FEATURES OF EZW

EZW is the first algorithm that presents the new concept of embedded zerotree image compression. It possesses the following key characteristics:

A. Properties

- Producing a fully embedded bit stream;
- Providing competitive compression performance;

B. Features

- Multiresolution representation of an image using wavelet decomposition;
- Successive approximation quantizaiton of the significant coefficients, which generates both a multiprocessing representation of the magnitudes of significant coefficients and an embedded bit stream;
- Zerotree encoding of significance maps, which takes advantage of the self-similarity of zeros inherent in the hierarchical system, producing successful predication of zeros across different scales;
- Lossless data compression using adaptive arithmetic entropy coding [Kopp].

3. PRINCIPLE OF EZW ALGORITHM

The first step in the EZW coding algorithm is to determine the initial threshold. If we adopt bit plane coding then our initial threshold T_0 [Valens] will be

$$T_0 = 2^{\lfloor \log_2(Max(|\gamma(x,y)|))\rfloor}$$

International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume No.2, Issue No.6, pp : 532-538 1 June 2013

Here MAX(.) means the maximum coefficient value in the image and $\gamma(x,y)$ denotes the coefficient. With this threshold we enter the main coding loop (according to C-language):

```
threshold = initial_threshold;
do {
        dominant_pass(image);
        subordinate_pass(image);
        threshold = threshold/2;
   }
while (threshold>minimum_threshold);
```

In the EZW technique of Shapiro [Shapiro], the wavelet coefficients are coded using a number of scans and in each scan a stream of symbols are generated which are subsequently entropy coded. The symbols used are SP, SN, ZR and IZ which are defined as follows.

- **SP** (Significant Positive) - a coefficient is coded as SP when it is positive and its value is greater than a threshold.
- **SN** (Significant Negative) - a coefficient is coded as SN when it is negative and its value greater than the threshold.
- **ZR** (zerotree Root)- a coefficient is coded as ZR when its value and the values of all of its descendents are less than the threshold.
- **IZ** (Isolated Zero)- a coefficient is coded as IZ when its value is less than the threshold but there is at least one descendent whose value is greater than threshold.

Every scan consists of two passes. The Dominant pass generates the symbol stream and the Subordinate pass refines the coefficient values, which are coded as SP or SN in the previous passes. Compression obtained is more when there is more ZR as in that case no information about the descendent coefficients needs to be sent.

4. BOTTOM-UP SEARCH

Searching the target coefficients from bottom-to-up. Every child node is scanned before its parent. For example, all 16 (figure 1) coefficients in HH1 level are belonging to their parents in level HH2. Similarly these 4 coefficients (in HH2 level) are dependent on their parent in level HH3. So searching from bottom-to-up, we can also identify which coefficients are significant or insignificant. This technique takes less computation as well as reduces the computation time. In figure 1, the input image is divided (according to discrete wavelet transform decomposes) into four different subbands : LL subband which contains the low frequency image content in the reduced resolution and the other bands LH, HL, HH containing the high frequency information.

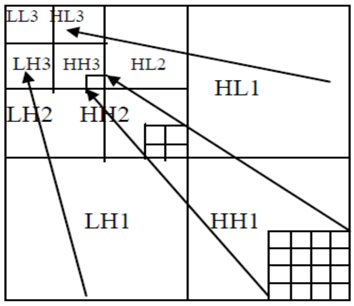

Fig. 1: Zerotree data structure for Bottom up search

5. CODING THE IMAGE

There are two types of passes performed: a dominant pass and a subordinate pass.

Dominant Pass:
Finds coefficients/pixel values above a certain threshold, which have not yet been found to be significant in the same relative order as the initial scan. After finding, dominant pass divides the pixel values in the following four categories:

i. A significant coefficient/pixel value is coded as SP when it is positive and its value is greater than a threshold [Gonzalez].
ii. An insignificant coefficient is coded as SN when it is negative and its value is greater than threshold.
iii. A zerotree coefficient is coded as ZR when its value and the values of all of its descendents are less than the threshold.
iv. An isolated zero coefficient is coded as IZ when its value is less than the threshold but there is at least one descendent whose value is greater then the threshold.

Only the significant coefficients are added to a subordinate list for quantization, and these coefficients are then set zero (0) for the next dominant pass.

Subordinate Pass:
After each dominant pass, a subordinate pass is then performed on the subordinate list, which contains all pixel values previously found to be significant. The subordinate pass performs pixel value quantization which achieves compression [Valens] by telling the decoder with a symbol roughly what the pixel value is instead of exactly what the pixel value has been.

Let T_i be the initial threshold, then the reconstruction magnitudes of the significant pixel values are taken as the center of the uncertainly interval [Raghu et. al.].

If the reconstruction magnitude is less than 1.5 x T_i than set with symbol "0"

If the reconstruction magnitudes is greater than or equal to $1.5xT_i$ than set with symbol “1”, where i=0,1,2,.....

Thus from the decoders' viewpoint the rough estimate of a significant pixels value is getting more refined and accurate as more subordinate passes are made. So, the subordinate passes quantize pixel values to a two symbol alphabet which then get encoded by using an adaptive arithmetic [Ponalagusam et. al.] coder and thus achieving compression.

6. DECODING THE IMAGE

Decoding an image compression by Shapiro's [Rabi, Ponalagusam et. al.] algorithm needs the initial threshold, the original image size, the subband decomposition scale and, of course, the encoded bit stream. The decoder then decompresses the arithmetically encoded files into symbol files, creates all the proper size subbands required since it knows the subband decomposition scale and the original image size, and proceeds to undo the Shapiro compression since it knows the initial threshold and the subband scanning order.

Fig. 2a : Original 512x512 grey-scale image of Lena

Fig. 2b: Reconstructed image with EZW coefficients

7. DOMINANT AND SUBORDINATE PASSES

Here a 4 scale wavelet image is used as a sample to show the algorithm. The original image (8x8) is shown in figure 2a and after applying bottom up search with EZW coefficients the decoded/reconstructed image is shown in figure 2b. Images of different dominant and subordinate passes are given from figure 3a to 6c and figure 7a is the final reconstructed image. The procedure of encoding together with decoding for the image is listed below:

Level 1

Dominant Pass 1: The threshold T_0 =32, and the quantization is 1.5 x T_0 = 48. The output symbols are as follows:

SP(63), SN(-34), IZ(-31), ZR(23), **SP(49)**, ZR(10), ZR(14), ZR(-13), ZR(15), IZ(14), ZR(-9), ZR(-7), ZR(-1), **SP(47)**, ZR(-3), ZR(2). Since each level requires 2 bits for fixed-length encoding, it needs 16x2=32 bits for this pass.

So the significant coefficients in the dominant pass are: SP(63), SN(-34), SP(49), SP(47).

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

Fig. 3a: Original image (8x8)

48	-48	48	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	48	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 3b: Image with quantization value 48(level 1)

The quantization value is 48, and we put this value instead of all significant coefficients: 63, -34, 49, 47 and the reconstructed image looks like figure 3b. Difference between magnitude of elements and reconstructed values are shown in table 1.

Table 1: Original and reconstructed values when threshold is 24 and quantization value is 48.

Coefficient Magnitude	Difference between elements and quantization	Reconstructed Magnitude	Symbol
63	63-48=15	48+8=56	1
-34	34-48=-14	48-8=40	0
49	49-48=1	48+8=56	1
47	47-48=-1	48-8=40	0

Calculate the reconstruction value: The correction term is $\pm T_0/4$ = ±8. If the difference between the magnitude of elements and quantization value is positive then add this value (8) with quantizaton value (48). If the difference is negative than subtract this value (8) from quantizaiton value (48), which is also shown in table 1. Now the corresponding reconstruction values for significant coefficients 63, -34, 49, 47 are 56, -40, 56, 40 respectively. Transmitting the correction term costs a single bit for each, so 4 bits are required in this pass. Therefore, at the end of the first pass, we have used 32+4=36 bits. With 36 bits, zero in the image replaces the subordinate list, and the image looks like 3c. For the next dominant pass, we use * for all significant values in figure 4a [Othman et. al].

56	-40	56	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	40	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 3c: Image with decoded values (level 1)

Subordinate Pass 1: 4 symbols are generated corresponding to the significant coefficients 63, 34, 49, 47 in the subordinate list, and their symbols are also listed in table 1. This is the end of level 1.

Level 2

Dominant Pass 2 : The threshold T_1 = 16, and the quatization is 1.5x T_1 = 24. The output symbols are as follows:

SN(-31), SP (23), ZR (10), ZR(14), ZR(-13), ZR(15), IZ(14) ZR(-9), ZR(-7), which need 9x2=18 bits.

Total Significant coefficients in the dominant pass are : SP(63), SN(-34), SN(-31), SP(23), SP(49), SP(47).

*	*	*	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	*	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

Fig. 4a: Original image for level 2

56	-40	56	0	0	0	0	0
-24	24	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	40	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 4b: Image with quantization value 24 (level 2)

The quantization value is 24, the corresponding reconstruction values for significant coefficients 63, -34, -31, 23, 49, 47 are 56, -40, -24, 24, 56, -40 and the reconstructed image looks like figure (e). Difference between magnitude of elements and reconstructed values are shown in table 2.

Table 2: Original and reconstructed values when threshold is 16 and quantization value is 24.

Coefficient Magnitude	Difference between elements and quantization	Reconstructed Magnitude	Symbol
63	63-56=7	56+4=60	1
-34	34-40=-6	40-4=36	0
-31	31-24=7	24+4=28	1
23	23-24=-1	24-4=20	0
49	49-56=-7	56-4=52	0
47	47-40=7	40+4=44	1

Calculate the reconstruction value : The correction term is $\pm T_1/4 = \pm 4$. If the difference between magnitude of elements and quantization value is positive then add this value (4) with quantizaton value (24). Similarly, if the difference is negative then subtract this value (4) from quantization value (24), which is also shown in table 2. Transmitting the correction terms costs a single bit for each, so we require 6 bits. Therefore, at the end of the second pass, we have used 36+18+6=60 bits. With 60 bits, zero in the image replaces the subordinate list, and the image looks like 4c. Again for the next dominant pass, we use * for all significant values in figure 5a.

60	-36	52	0	0	0	0	0
-28	20	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	44	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 4c: Image with decoded values (level 2)

Subordinate Pass 2 : 6 symbols are generated corresponding to the significant coefficients 63, 34, 31, 23, 49, 47 in the subordinate list, and their symbols are also listed in table 2. This is the end of level 2.

Level 3

Dominant Pass 3 : The threshold T_2 = 8, and the quantization is 1.5xT_2 = 12. The output symbols are as follows:
SP(10), SP(14), SN(-13), SP(15), SP(14), SN(-9), ZR(-7), ZR(3), **SN(-12), SN(-14), SP(8),** ZR(7), **SP(13)**, ZR(3), ZR(4), **SN(-12)**, ZR(7), ZR(6), ZR(-1), ZR(3), **SP(9)**, ZR(3), ZR(2), ZR(-5), **SP(9)**, ZR(3), ZR(0), ZR(2), ZR(-3), ZR(5), **SP(11)**, which need 31x2=62 bits.

Total significant coefficients in the dominant pass are: SP(63), SN(-34), SN(-31), SP(23), SP(49), SP(10), SP(14), SN(-13), SP(15), SP(14), SN(-9), SN(-12), SN(-14), SP(8), SP(13), SN(-12), SP(9), SP(9), SP(47), SP(11).

*	*	*	10	7	13	-12	7
*	*	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	*	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

Fig. 5a: Original image for level 3

60	-36	52	12	0	12	-12	0
-28	20	12	12	0	0	0	0
12	12	0	-12	0	0	0	12
-12	0	-12	12	0	0	0	0
0	12	0	44	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 5b: Image with quantization value 12 (level 3)

The quantization value is 12, and using this value to the corresponding significant coefficients, the reconstructed bands look like figure 5b. Now correction term is $\pm T_2/4 = \pm 2$. Transmitting the correction term costs a single bit for each, so it requires 20 bits.

Table 3: Original and reconstructed values when threshold is 8 and quantization value is 12.

Coefficient Magnitude	Difference between elements and quantization	Reconstructed Magnitude	Symbol
63	63-60=3	60+2=62	1
-34	34-36=-2	36-2=34	0
-31	31-28=3	28+2=30	1
23	23-20=3	20+2=22	1
49	49-52=-3	52-2=50	0
10	10-12=-2	12-2=10	0
14	14-12=2	12+2=14	1
-13	13-12=1	12+2=14	1
15	15-12=3	12+2=14	1
14	14-12=2	12+2=14	1
9	9-12=-3	12-2=10	0
12	12-12=0	12+0=12	1
14	14-12=2	12+2=14	1
8	8-12=-4	12-2=10	0
13	13-12=1	12+2=14	1
12	12-12=0	12+0=12	1
9	9-12=-3	12-2=10	0
9	9-12=-3	12-2=10	0
47	47-44=3	44+2=46	1
11	11-12=-1	12-2=10	0

Calculate the reconstruction value : Accordingly at the end of the 3rd pass, we have used 60+62+20 = 142 bits. With 142 bits zero in the image replaces the subordinate list, and the image looks like 5c. Difference between magnitude of elements and reconstructed values is also shown in table 3. Again for the next dominant pass, we use * for all significant values in figure 6a.

62	-34	49	10	0	14	12	0
-30	22	12	-14	0	0	0	0
14	14	0	0	0	0	0	0
-10	0	-14	10	0	0	0	0
0	0	0	46	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Fig. 5c: Image with decoded values (level 3)

Subordinate Pass 3: 20 symbols are generated corresponding to the significant coefficients as above, and their symbols are listed in table 3. This is the end of level 3.

Level 4

Dominant Pass 4 : The threshold $T_3 = 4$, and the quantization is 1.5x $T_3 = 6$. The output symbols are as follows:

SN(-7), IZ(3), SP(7), ZR(3), SP(4) , SP(7), SP(6), ZR(-1), SP(5), SN(-7), SP(4), ZR(-2), SN(-5), ZR(3), ZR(0), ZR(2), ZR(-3), SP(5), SP(6), SN(-4), SP(5), SP(6), SP(4), SP(6), ZR(3), ZR(-2), ZR(-2), ZR(2), ZR(0), SP(4), ZR(3), SP(6), ZR(0), ZR(3), SP(6), SN(-4), SP(4), which need 38x2=76 bits.

Total significant coefficients in the dominant pass are: SP(63), SN(-34), SN(-31), SP(23), SP(49), SP(10), SP(14), SN(-13), SP(15), SP(15), SP(14), SN(-9), SN(-7), SN(-12), SN(-14), SP(8), SP(7), SP(13), SP(4), SN(-12), SP(7), SP(6), SP(5), SN(-7), SP(4), SP(9), SN(5), SP(9), SP(47), SP(5), SP(11), SP(6(, SN(-4), SP(5), SP(6), SP(4), SP(6), SP(4), SP(6), SP(6), SN(4), SP(4).

*	*	*	*	7	*	*	7
*	*	*	*	3	4	6	-1
*	*	3	*	5	-7	3	*
*	-7	*	*	4	-2	3	2
-5	*	-1	*	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	*	5	6	0	3	-4	4

Fig. 6a: Original image for level 4

62	-34	49	10	6	14	-12	6
-30	22	14	-14	0	6	6	0
14	14	0	-12	6	-6	0	10
-10	-6	-14	10	6	0	0	0
-6	10	0	46	6	6	0	0
0	0	0	0	0	0	0	6
0	0	6	-6	0	6	0	6
6	10	6	6	0	0	-6	6

Fig. 6b: Image with quantization value 6 (level 4)

Table 4: Original and reconstructed values when threshold is 8 and quantization value is 6.

Coefficient Magnitude	Difference between elements and quantization	Reconstructed Magnitude	Symbol
63	63-62=1	62+1=63	1
-34	34-34=0	34+0=34	1
-31	31-30=1	30+1=31	1
23	23-22=1	22+1=23	1
49	49-49=0	49+0=49	1
10	10-10=0	10+0=10	1
14	14-14=0	14+0=14	1
-13	13-14=-1	14-1=13	0
15	15-14=1	14+1=15	1
14	14-14=0	14+0=14	1
-9	9-10=-1	10-1=9	0
-7	7-6=1	6+1=7	1
-12	12-12=0	12+0=12	1
-14	14-14=0	14+0=14	1
8	8-10=-2	10-1=9	0
7	7-6=1	6+1=7	1
13	13-14=-1	14-1=13	0
4	4-6=-2	6-1=5	0
-12	12-12=0	12+0=12	1
7	7-6=1	6+1=7	1
6	6-6=0	6+0=6	1
5	5-6=-1	6-1=5	0
-7	7-6=1	6+1=7	1
4	4-6=-2	6-1=5	0
9	9-10=-1	10-1=9	0
-5	5-6=-1	6-1=5	0
9	9-10=-1	10-1=9	0
47	47-46=1	46+1=47	1
5	5-6=-1	6-1=5	0
11	11-10=1	10+1=11	1
6	6-6=0	6-6=0	1
-4	4-6=-2	6-1=5	0
5	5-6=-1	6-1=5	0
6	6-6=0	6+0=6	0
4	4-6=-2	6-1=5	0
6	6-6=0	6+0=6	1
4	4-6=-2	6-1=5	0
6	6-6=0	6+0=6	1
6	6-6=0	6+0=6	1
-4	4-6=-2	6-1=5	0
4	4-6=-2	6-1=5	0

Calculate the reconstruction value :The quantization value is 6, and using this value corresponding to the significant coefficients, the reconstructed bands look like figure 6b. Now correction term is $\pm T_3/4 = \pm 1$. Transmitting the correction term costs a single bit for each, so it requires 41 bits. Therefore, at the end of 4th pass, we have used 142+76+41=259 bits. With 259 bits, zero in the image replaces the subordinate list, and the image looks like 6c. Difference between magnitude of elements and reconstructed values are shown in table 4.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	0	5	6	0
15	14	0	-12	5	-7	0	9
-9	-7	-14	9	5	0	0	0
-5	9	0	47	5	6	0	0
0	0	0	0	0	0	0	5
0	0	6	-5	0	6	0	6
5	11	5	6	0	0	-5	5

Fig. 6c: Image with decoded values (level 4)

Subordinate Pass 4: 41 symbols are generated corresponding to the significant coefficients as above, and their symbols are listed in table 4. This is the end of level 4. After end of level 4, the final reconstructed image looks like figure 7a.

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

Fig. 7a: Final reconstructed image with all decoded values

8. CONCLUSION AND FURTURE WORKS

Wavelet transformation technique is a lossy compression technique [Saravanam et. al.]. But if we look over the original and reconstructed images, both the images look almost the same except some coefficients. The coefficients, which are different from the original image, are marked bold-faced/colored in figure 7a. This minor difference between the original and reconstructed images is out of human eyesight. Compression depends not only on scaling, but also on the number of levels of filtering. However, there are still some rooms for improvement in this algorithm. For example, the correlation of the magnitudes of significant coefficients within each subband is not efficiently exploited, and coefficients in each subband are scanned and encoded line-by-line. In our experimental image "Lena" of size 512x512, number of levels of filtering is 4 and number of passes is 6 and we achieved the reconstructed image like as original with high compression ratio. We examined if these passes are less then 6 with the number of levels of filter being 4, the reconstructed image is good enough. But using more passes than 6, the image reconstruction would be the same. So we can also conclude that 6-passes with levels of filtering being 4 are the appropriate operations.

9. REFERENCES

1. *David Salomon (2001): Data Compression : The Complete Reference, 2nd Edition Springer.*
2. *Gonzalez, R.C. and Woods, R.E.(2005): Digital Image Processing 2nd ed., Pearson Education, India, 2005.*
3. *JPEG-2000 requirements and profiles version 4.0, ISO/IEC/JTCI/SC/WG1 N1105R, December 1980.*
4. *Kopp M (1996): Lossless Wavelet Based Image Compression with Adaptive 2D Decomposition, Institute of Computer Graphics, Technical University of Vienna. http://wscg.zcu.cz/wscg96/papers96/Kopp_96.pdf*
5. *Lossless Data Compression : http://www.data-compression.com/lossless.html*
6. *Othman O. Khalifa, Sering Habib Harding and Aisha-Hassan A. Hashim (2008): Compression using Wavelet Transform, Signal Processing: An International Journal, Volume (2), Issue (5), pp. 17-26.*
7. *Ponalagusamy R. and Saravanan C. (2007): Analysis of Medical Image Compression using Statistical Coding Methods, Advances in Computer Science and Engineering: Reports and Monographs, Imperial College Press, UK, Vol.2., pp 372-376, 2007.*
8. *Rabi Polikar: Wavelet Tutorial. http://users.rowan.edu/~polikar/WAVELETS/WTpart1.html*
9. *Raghu K. and Biswas P.K. : Fractal Based Image Compression.*
10. *Saravanan C. & Ponalagusamy R. (2009): Lossless Grey-scale Image Compression using Source Symbols Reduction and Huffman Coding,*

International Journal of Image Processing (IJIP), Volume (3): Issue (5).

11. *Shapiro J. M. (1993): Embedded Image Coding Using Zerotrees of Wavelet Coefficients, IEEE Trans. On Signal Processing, SP-41, pp.3445-4462, December.*

12. *Shufang Wu: Embedded Zerotree Wavelet – An Image Coding Algorithm, www.cs.sfu.ca/CC/820/li/material/presentations/paper11.ppt*

13. *Singara Singh , R. K. Sharma, M.K. Sharma (2009): Use of Wavelet Transform Extension for Graphics Image Compression using JPEG2000 Framework, International Journal of Image Processing, Volume 3, Issue 1, Pages 55-60.*

14. *Valens. C. (1999): Embedded Zerotree Wavelet Encoding, http://perso.wanadoo.fr/polyvalens/clemens/ezw/ezw.html*

ACKNOWLEDGMENTS

I am very grateful to Professor P.K. Biswas, Department of Electronics and Electrical Communication Engineering, Indian Institute of Technology, Kharagpur, India, while I attended his lecturers at the University of Kaiserslautern, Germany.