Design and Implementation of EZW & SPIHT Image Coder for Virtual Images

Priyanka Singh *priyanka10ec@gmail.com*
Research Scholar, ECE Department
Amity University
Gurgaon (Haryana, India)

Priti Singh *pritip@rediffmail.com*
Professor, ECE Department
Amity University
Gurgaon (Haryana, India)

Abstract

The main objective of this paper is to designed and implemented a EZW & SPIHT Encoding Coder for Lossy virtual Images. Embedded Zero Tree Wavelet algorithm (EZW) used here is simple, specially designed for wavelet transform and effective image compression algorithm. This algorithm is devised by Shapiro and it has property that the bits in the bit stream are generated in order of importance, yielding a fully embedded code. SPIHT stands for Set Partitioning in Hierarchical Trees. The SPIHT coder is a highly refined version of the EZW algorithm and is a powerful image compression algorithm that produces an embedded bit stream from which the best reconstructed images. The SPIHT algorithm was powerful, efficient and simple image compression algorithm. By using these algorithms, the highest PSNR values for given compression ratios for a variety of images can be obtained. SPIHT was designed for optimal progressive transmission, as well as for compression. The important SPIHT feature is its use of embedded coding. The pixels of the original image can be transformed to wavelet coefficients by using wavelet filters. We have anaysized our results using MATLAB software and wavelet toolbox and calculated various parameters such as CR (Compression Ratio), PSNR (Peak Signal to Noise Ratio), MSE (Mean Square Error), and BPP (Bits per Pixel). We have used here different Wavelet Filters such as Biorthogonal, Coiflets, Daubechies, Symlets and Reverse Biorthogonal Filters .In this paper we have used one virtual Human Spine image (256X256).

Keywords: Image Compression, Embedded Zerotree Wavelet, Set Partitioning in Hierarchical Trees, CR, PSNR, MSE, BPP.

1. INTRODUCTION

With the growth of technology and the entrance into the Digital Age, the world has found itself a vast amount of information. Dealing with such enormous amount of information can often present difficulties. Digital information must be stored, retrieved, analyzed and processed in an efficient manner, in order for it to be put to practical use. Image compression is technique under image processing having wide variety of applications. Image data is perhaps the greatest single threat to the capacity of data networks [1,2].As image analysis systems become available on lower and lower cost machines, the capability to produce volume of data becomes available to more and more users. New data storage technologies have been developed to try to keep pace with the potential for data creation.

1.1 Image Compression

The fundamental components of compression are redundancy and irrelevancy reduction. Redundancy means duplication and Irrelevancy means the parts of signal that will not be noticed by the signal receiver, which is the Human Visual System (HVS).
There are three types of redundancy can be identified:

• **Spatial Redundancy** i.e. correlation between neighboring pixel values.
• **Spectral Redundancy** i.e. correlation between different color planes or spectral bands.

• **Temporal Redundancy** i.e. correlation between adjacent frames in a sequence of images.

Image compression focuses on reducing the number of bits needed to represent an image by removing the spatial and spectral redundancies. The removal of spatial and spectral redundancy is often accomplished by the predictive coding or transforms coding. Quantization is the most important means of irrelevancy reduction [2].

1.2 Basic Types of Image Compression

Basic types of image compression are lossless and lossy. Both compression types remove data from an image that isn't obvious to the viewer, but they remove that data in different ways. Lossless compression works by compressing the overall image without removing any of the image's detail. As a result the overall file size will be compressed. Lossy compression works by removing image detail, but not in such a way that it is apparent to the viewer. In fact, lossy compression can reduce an image to one tenth of its original size with no visible changes to image quality [2, 3].

One of the most successful applications of wavelet methods is transform-based image compression (also called coding). The overlapping nature of the wavelet transform alleviates blocking artifacts, while the multiresolution character of the wavelet decomposition leads to superior energy compaction and perceptual quality of the decompressed image. Furthermore, the multiresolution transform domain means that wavelet compression methods degrade much more gracefully than block-DCT methods as the compression ratio increases. Since a wavelet basis consists of functions with both short support (for high frequencies) and long support (for low frequencies), large smooth areas of an image may be represented with very few bits, and detail added where it is needed. Wavelet-based coding provides substantial improvements in picture quality at higher compression ratios. Over the past few years, a variety of powerful and sophisticated wavelet-based schemes for image compression, as discussed later, have been developed and implemented. Because of the many advantages, wavelet based compression algorithms are the suitable candidates for the new JPEG-2000 standard. Such a coder operates by transforming the data to remove redundancy, then quantizing the transform coefficients (a lossy step), and finally entropy coding the quantizer output. The loss of information is introduced by the quantization stage which intentionally rejects less relevant parts of the image information. Because of their superior energy compaction properties and correspondence with the human visual system, wavelet compression methods have produced superior objective and subjective results [4]. With wavelets, a compression rate of up to 1:300 is achievable [5]. Wavelet compression allows the integration of various compression techniques into one algorithm. With lossless compression, the original image is recovered exactly after decompression. Unfortunately, with images of natural scenes, it is rarely possible to obtain error-free compression at a rate beyond 2:1 [5-6]. Much higher compression ratios can be obtained if some error, which is usually difficult to perceive, is allowed between the decompressed image and the original image.

2. EMBEDDED ZERO TREE WAVELET (EZW)

The EZW algorithm was introduced in the paper of Shapiro [2]. The core of the EZW compression is the exploitation of self-similarity across different scales of an image wavelet transform. The Embedded Zero-tree Wavelet (EZW) algorithm is considered the first really efficient wavelet coder. Its performance is based on the similarity between sub-bands and a successive-approximations scheme. Coefficients in different sub-bands of the same type represent the same spatial location, in the sense that one coefficient in a scale corresponds with four in the prior level. This connection can be settled recursively with these four coefficients and its corresponding ones from the lower levels, so coefficient trees can be defined. In natural images most energy tends to concentrate at coarser scales (higher levels of decomposition), then it can be expected that the

nearer to the root node a coefficient is, the larger magnitudes it has. So if a node of a coefficient tree is lower than a threshold, it is likely that its descendent coefficients will be lower too.
We can take profit from this fact, coding the sub-band coefficients by means of trees and successive-approximation, so that when a node and all its descendent coefficients are lower than a threshold, just a symbol is used to code that branch The successive-approximation can be implemented as a bit-plane encoder. The EZW algorithm is performed in several steps, with two fixed stages per step: the dominant pass and the subordinate pass. In Shapiro's paper the description of the original EZW algorithm can be found. However, the algorithm specification is given with a mathematical outlook. We present how to implement it, showing some implementation details and their impact on the overall codec performance. Consider we need n bits to code the highest coefficient of the image (in absolute value). The first step will be focused on all the coefficients that need exactly n bits to be coded. In the dominant pass, the coefficients which falls (in absolute value) in this range are labeled as a significant positive/negative (sp/sn), according to its sign. These coefficients will no longer be processed in further dominant passes, but in subordinate passes. On the other hand, the rest of coefficients are labeled as zero-tree root (zr), if all its descendants also belong to this range, or as isolated zero (iz), if any descendant can be labeled as sp/sn. Notice that none descendant of a zero-tree root need to be labeled in this step, so we can code entire zero-trees with just one symbol. In the subordinate pass, the bit n of those coefficients labeled as sp/sn in any prior step is coded. In the next step, the n value is decreased in one so we focus now on the following least significant bit [7]. Compression process finishes when a desired bit rate is reached. That is why this coder is so called embedded. In the dominant pass four types of symbols need to be code (sp, sn, zr, iz), whereas in the subordinate pass only two are needed (bit zero and bit one). Finally, an adaptive arithmetic encoder is used to get higher entropy compression. EZW approximates higher frequency coefficients of a wavelet transformed image. Because the wavelet transform coefficients contain information about both spatial and frequency content of an image, discarding a high-frequency coefficient leads to some image degradation in a particular location of the restored image rather than across the whole image. Here, the threshold is used to calculate a significance map of significant and insignificant wavelet coefficients. Zerotrees are used to represent the significance map in an efficient way. Figure 1 shows the Embedded Zerotree Scanning process.

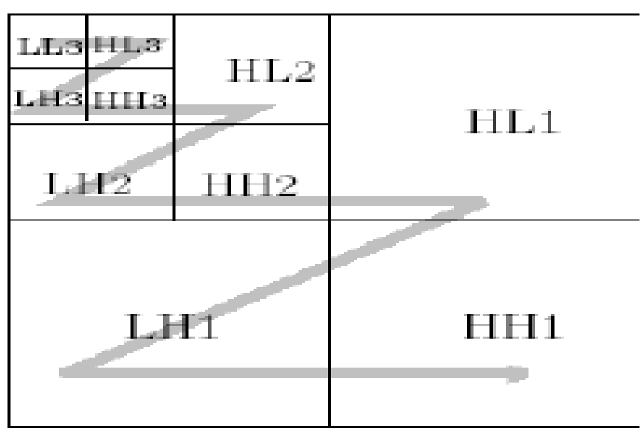

FIGURE 1 Scanning of a Zerotree

3. EZW ENCODING ALGORITHM

1. Initialization: Set the threshold T to the smallest power of that is greater than max (i,j) |c i,j|/2, where Ci,j are the wavelet coefficients.

2. Significance map coding: Scan all the coefficients in a predefined way and output a symbol when |Ci,j | > T. When the decoder inputs this symbol, it sets ci,j = ±1.5T.

3. Refinement: Refine each significant coefficient by sending one more bit of its binary representation. When the decoder receives this, it increments the current coefficient value by ±0.25T.

4. Set Tk = Tk-1/2, and go to step 2 if more iterations are needed [7-8].

The next scheme, called SPIHT, is an improved form of EZW which achieves better compression and performance than EZW.

4. SET PARTITIONING IN HIERARICHAL TREES (SPIHT)

SPIHT sorts the coefficients and transmits their most significant bits first. A wavelet transform has already been applied to the image and that the transformed coefficients are sorted.The next step of the encoder is the refinement pass. The encoder performs a sorting step and a refinement step in each iteration. SPIHT uses the fact that sorting is done by comparing two elements at a time, and each comparison results in a simple yes/no result. The encoder and decoder use the same sorting algorithm, the encoder can simply send the decoder the sequence of yes/no results, and the decoder can use those to duplicate the operations of the encoder. The main task of the sorting pass in each iteration is to select those coefficients that satisfy 2n<=|ci,j |<2n+1. This task is divided into two parts. For a given value of n, if a coefficient ci,j satisfies |ci,j|>=2n, then that it is said as significant; otherwise, it is called insignificant. The encoder partitions all the coefficients into a number of sets Tk and performs the significance test.

$$S_n(T) = \begin{cases} 1, \max_{(i,j)\in T} |C_{I,J}| \geq 2^N \\ 0, Otherwise. \end{cases}$$

…………………………..eq. (1)

On each set Tk. The result may be either “no” This result is transmitted to the decoder. If the result is “yes,” then Tk is partitioned by both encoder and decoder, using the same rule, into subsets and the same significance test is performed on all the subsets. This partitioning is repeated until all the significant sets are reduced to size 1. The result, Sn(T), is a single bit that is transmitted to the decoder.
The sets Tk are created and partitioned using a spatial orientation tree. This set partitioning sorting algorithm uses the following four sets of coordinates:

1. The set contain the coordinates of the four offspring of node is Off [i,j]. If node is a leaf of a spatial orientation tree, then Off [i, j] is empty.

2. The set contain the set of coordinates of the descendants of node is called Des[i,j].

3. The set contain the set of coordinates of the roots of all the spatial orientation trees called R.

4. Next the set is a difference set Des[i,j]- Off[i,j]. This set contains all the descendants of tree node except its four offspring as Diff [i,j].

The spatial orientation trees are used to create and partition the sets Tk. The partitioning rules are given below:

1. Each spatial orientation tree need initial set.

2. If set Des[i, j] is significant, then it is partitioned into Diff[i, j] plus the four single element sets with the four offspring of the node.

3. If Diff[i, j] is significant, then it is partitioned into the four sets Des[k, l], where k=1..4 of node .

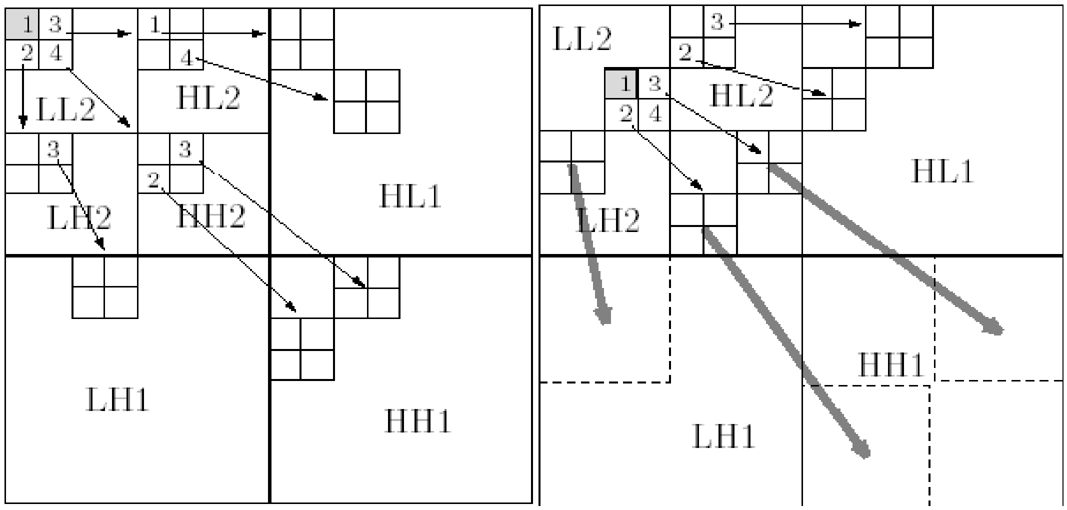

FIGURE 2 Shows the Spatial Orientation Trees in SPIHT.

5. SPIHT Algorithm

It is important to have the encoder and decoder test sets for significance .So the coding algorithm uses three lists called SP for list of significant pixels, initialized as empty, IP is list of insignificant pixels for the coordinates of all the root node belongs to root set R, and IS is list of insignificant sets to the coordinates of all the root node in R that have descendants and treated as special type entries [5].

Procedure:

Step 1: Initialization: Set n to target bit rate.
for each node in IP do:
if Sn [i, j] = 1,(according to eq 4.1)
move pixel coordinates to the SP and
keep the sign of ci,j ;

Step 2: for each entry in the IS do the following steps:
if the entry is root node with descendants
if Sn(Des[i, j]) = 1, then
for each offspring (k, i) in Off[i, j] do:
if (Sn(k, i) = 1) then
{ add to the SP,
output the sign of ck,l;}
else
attach (k, l) to the IP;
if (Diff[i, j] <> 0)
{move (i, j) to the end of the IS,
go to X;}
else
remove entry from the IS;
If the entry is root node without descendants then
output Sn(Diff[i, j]);
if Sn(Diff[i, j]) = 1, then
append each (k, l) in Off(i, j) to the IS as a special
entry and remove node from the IS:

Step 3: Refinement pass: for each entry in the SP, except those included in the last process for sorting, output the nth most significant bit of |i,j|;

Step 4: Loop: reduced n by 1 and go to X if needed.

6. Experimental Results & Analysis

In this paper we have implemented our result on a grayscale virtual image named Humane Spine having size (256X256) using various Wavelet filter families. Here we used MATLAB 2011(a) software and wavelet toolbox for analysing our results. The results of experiments are used to find the CR (Compression Ratio), BPP (Bits per Pixel), PSNR (Peak Signal to Noise Ratio) values and MSE (Mean Square Error) values for the reconstructed images. Fig 3(a) shows the Original Image and fig 3(b) shows the compressed image by EZW. Similarly fig 4(a) shows original image and fig 4(b) shows the compressed image by SPIHT image compression algorithms. The result got by EZW & SPIHT is shown in the Following Tables 1-2.Table 1 shows the results of CR, BPP, MSE & PSNR by using EZW algorithm. Table 2 shows the values for SPIHT Algorithms.

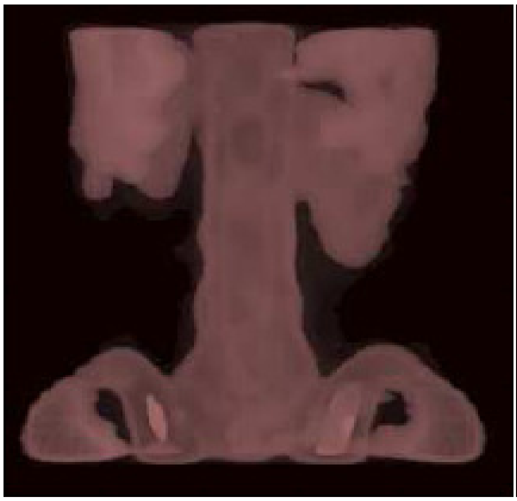


FIGURE 3(a) Original Image 3(b) Compressed Image by EZW


FIGURE 4(a) Original Image 4(b) Compressed Image by SPIHT

Image ---Spine, Size---2.37 KB, Entropy--4.102				
Wavelet	**CR**	**BPP**	**mse**	**psnr db**
	%		**Db**	
bior3.1	13.98	1.12	0.57	50.55
dmey	19.06	1.52	0.26	54.04
db8	19.13	1.53	0.26	53.98
sym5	17.91	1.43	0.27	53.89
coif2	17.91	1.43	0.47	51.46
rbio4.4	20.29	1.62	0.25	54.09

TABLE 1: Various Parameters Values of Different Wavelet for EZW Algorithm

Image ---Spine , Size---2.37 KB, Entropy--4.102				
Wavelet	**CR %**	**BPP**	**mse Db**	**psnr db**
bior3.1	8.73	0.7	0.61	50.29
dmey	7.85	0.63	0.42	51.94
db8	8.23	0.66	0.41	51.99
sym5	7.63	0.61	0.4	52.1
coif2	7.7	0.62	0.6	50.35
rbio4.4	8.77	0.7	0.38	52.32

TABLE 2: Various Parameters Values of Different Wavelet for SPIHT Algorithm

The graphical representation of CR, BPP, PSNR and MSE values are expressed as a bar graph are shown in Fig. 5, 6, 7 and Fig. 8. The main features of EZW include compact multiresolution representation of images by discrete wavelet transformation, zerotree coding of the significant wavelet coefficients providing compact binary maps, successive approximation quantization of the wavelet coefficients, adaptive multilevel arithmetic coding, and capability of meeting an exact target bit rate with corresponding rate distortion function (RDF) [7]. The SPIHT method provides highest image quality, progressive image transmission, fully embedded coded file, Simple quantization algorithm, fast coding/decoding, completely adaptive, lossless compression.

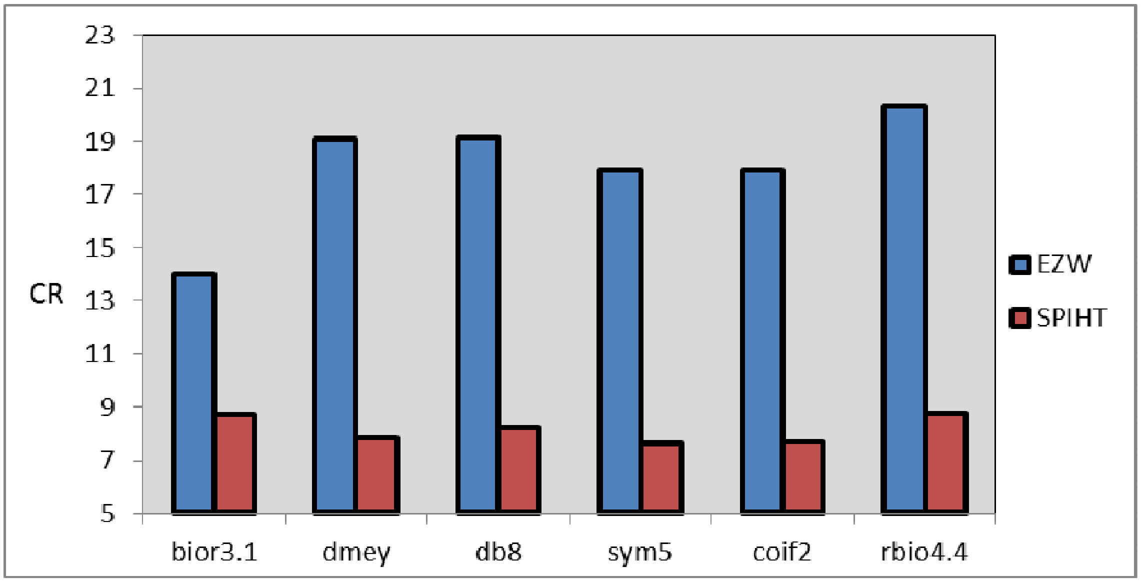

FIGURE 5 Comparison Chart of CR using EZW & SPIHT Algorithms.

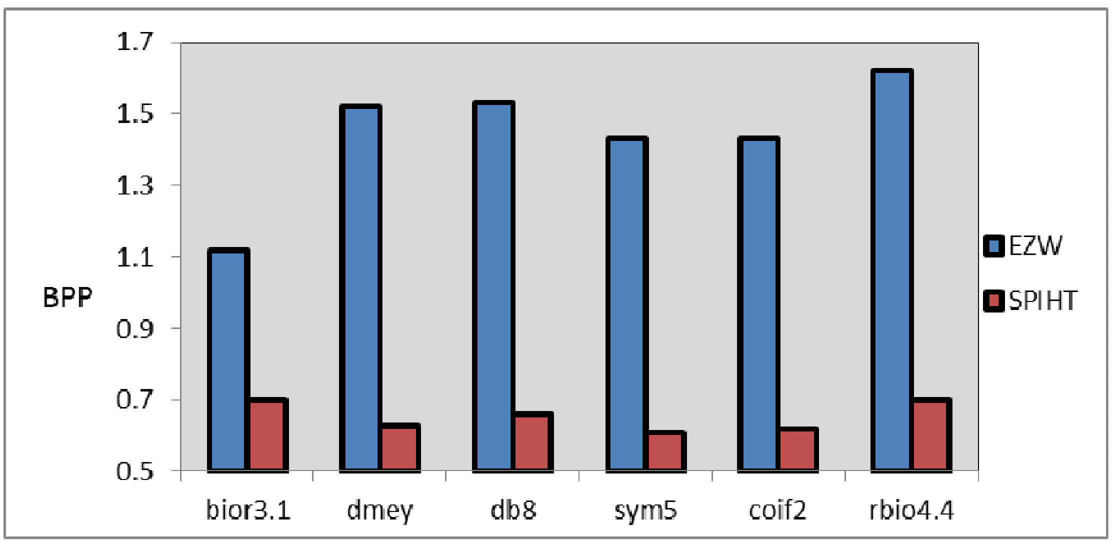

FIGURE 6 Comparison Chart of BPP using EZW & SPIHT Algorithms.

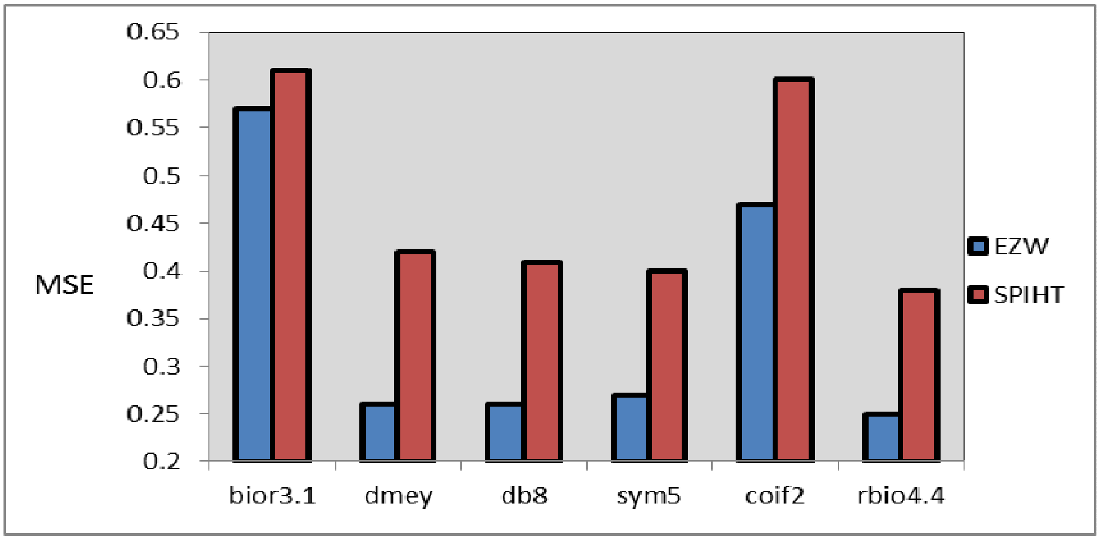

FIGURE 7 Comparison Chart of MSE using EZW & SPIHT Algorithms.

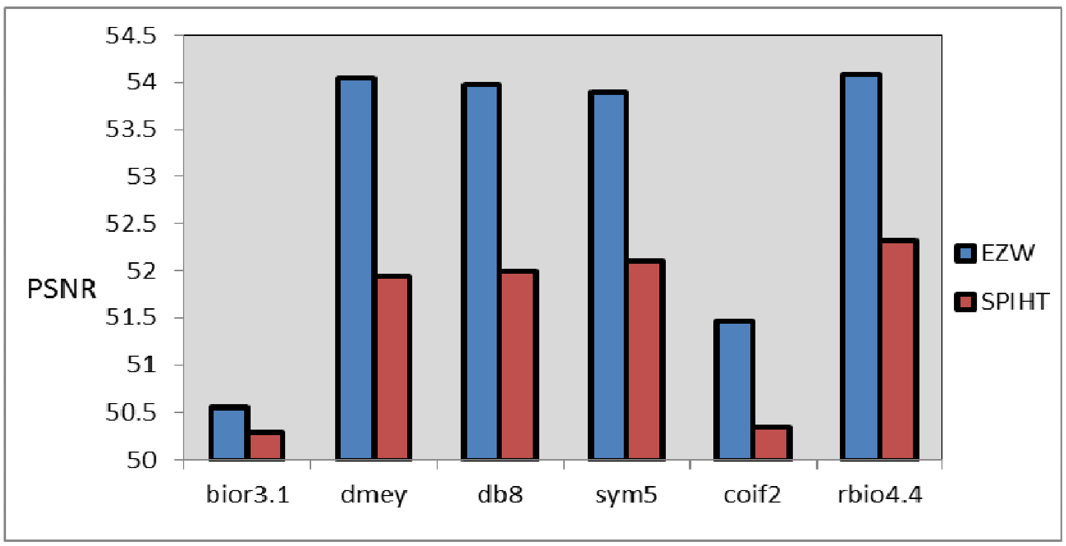

FIGURE 8 Comparison Chart of PSNR using EZW & SPIHT Algorithms.

7. CONCLUSION & FUTURE WORK

In this paper, the results of two different wavelet-based image compression techniques are compared. The effects of different wavelet functions filter orders, number of decompositions, image contents and compression ratios are examined. The results of the above techniques EZW and SPIHT are compared by using four parameters such as CR, BPP, PSNR and MSE values from the reconstructed image. These compression algorithms provide a better performance in picture quality at low bit rates. We found from our experimental results that SPIHT is better algorithm than EZW. Its results are 60-70% better than EZW as we can see from Table 1-2.We have analyzed that CR is reduced. BPP is also low comparative to EZW.MSE is low and PSNR is increased by a factor of 13-15%.as we can verify from our experiments. The above algorithms can be used to compress the image that is used in the web applications. Furthermore in future we can analyze different Image coding algorithms for improvement of different parameters.

8. REFERENCES

[1] R.Sudhakar, Ms R Karthiga, S.Jayaraman, "Image Compression using Coding of Wavelet Coefficients – A Survey", ICGST-GVIP Journal, Volume (5), Issue (6), June 2005.

[2] J. M. Shapiro, "Embedded image coding using zero trees of wavelet Coefficients", IEEE Trans. Signal Processing, vol. 41, pp. 3445- 3462, 1993.

[3] Basics of image compression - from DCT to Wavelets: a review.

[4] Bopardikar, Rao "Wavelet Transforms: Introduction to Theory and Applications."

[5] T.Ramaprabha M Sc M Phil ,Dr M.Mohamed Sathik, "A Comparative Study of Improved Region Selection Process in Image Compression using SPIHT and WDR" International Journal of Latest Trends in Computing (E-ISSN: 2045-5364) Volume 1, Issue 2, December 2010

[6] Shamika M. Jog, and S. D. Lokhande, "Embedded Zero-Tree Wavelet (EZW) Image CODEC" ICAC3'09, January 23–24, 2009, Mumbai, Maharashtra, India.

[7] S.P.Raja, A. Suruliandi "Performance Evaluation on EZW & WDR Image Compression Techniques", IEEE Trans on ICCCCT, 2010.

[8] Loujian yong, Linjiang, Du xuewen "Application of Multilevel 2- D wavelet Transform in Image Compression". IEEE Trans on 978-1-4244-3291-2, 2008.

[9] Javed Akhtar, Dr Muhammad Younus Javed "Image Compression With Different Types of Wavelets" IEEE Trans on Emerging Technologies, Pakistan, Nov 2006.

[10] Rafael C. Gonzalez and Richard E. Woods, "Digital Image Processing", 2nd Edition, Prentice Hall Inc, 2002.

[11] Khalid Sayood, "Introduction to Data Compression", 3rd Edition 2009

[12] G. Sadashivappa, K.V.S. Ananda Babu, "WAVELET FILTERS FOR IMAGE COMPRESSION, AN ANALYTICAL STUDY" ICGST-GVIP journal, volume (9), Issue (5), September 2009, ISSN: 1687-398X

[13] Lou jian yong,Lin jiang and Du xuewen "Application of Multilevel 2-D wavelet Transform in Image Compression"IEEE Trans on Signal Processing 978-1-4244-3291-2, 2008.