International Journal of Engineering Research and Development
e-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 9, Issue 12 (February 2014), PP. 13-20

Compression Efficiency of Different Embedded Image Compression Techniques with Huffman Encoding

Yasaswi Velamuri[1], Sandhya Patnayakuni[2], Nancharaiah Vejendla[3]
[1, 2] UG Student, ECE Department, Lendi Institute of Engg & Tech, Jonnada, VZM, INDIA.
[3]Associate Professor, ECE Department, Lendi Institute of Engg & Tech, Jonnada, VZM, INDIA.

Abstract:- Images require substantial storage and transmission resources. Image compression reduces the data required to represent these images. Image compression normally uses reduction in bits representing the pixels. Here we use the Discrete wavelet Transform (DWT) followed by EZW and SPIHT algorithms. Discrete wavelet transform reduces the noise levels and it is advantageous in its high speed and accuracy. EZW and SPIHT algorithms are lossy compression methods and the accuracy of image depends on bit rate. For further compression we use "Huffman coding" with EZW and SPIHT algorithms. The results are analyzed for "Lena" image.
Keywords:- zero tree**,** compression, wavelets, EZW, SPIHT, Huffman coding

I. INTRODUCTION

A. Wavelet Compression:

Advancements in the field of wavelets continue to be reported in many areas. Many authors have contributed to introduction of wavelet packet[4].

To represent complex signals efficiently there should be a localized function in both time and frequency domains. The wavelet function is localized in time domain as well as in frequency domain. It decomposes the image, and generates four different horizontal frequencies and vertical frequencies outputs. The outputs are referred to as approximation, horizontal detail, vertical detail, and diagonal detail. This approximation contains low frequency horizontal and vertical components of the image. The decomposition procedure is repeated on the approximation sub-band to generate the next level of the decomposition, and so on.

B. Embedded coding:

Using the embedded code the encoding will be stopped as soon the target bit stream is reached. The representation of image depends on the target bit stream or the distortion metric by which the obtained image is considered as lossy or lossless [5,6].

C. EZW and SPIHT algorithms:

These two algorithms are based on the concept of "Zero trees". The coefficients obtained from the DWT (2D-wavelet transform)[1] are compared with the threshold value.

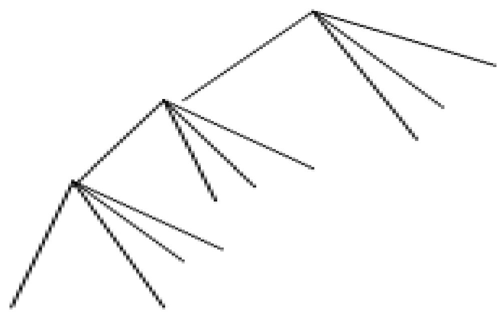

Fig1: Tree structure of wavelet decomposition

The concept of zero trees arises when the coefficients of a tree are less than the threshold value compared [2]. EZW is a simple and effective statistical image compression algorithm that doesn't require a train but needs a previously coded data. SPIHT algorithm is the modified version of EZW. Here the tree roots are separated from the tree efficiently and thus the compression will be more efficient. This is followed by the

Huffman coding which reduces the bits again to some extent but the main aim here is lossless transmission. The algorithms will be discussed in detail in the next section.

II. EZW ALGORITHM

Embedded zero tree wavelet algorithm is one of the effective image compression algorithm which is based on "parent- child" dependencies having the property that the bits are generated in order of importance which yields fully embedded code. This algorithm is flexible to terminate encoding or decode the bit stream at any time, and still produces same image.

J.M. Shapiro, in "**Embedded Image Coding using Zero trees of Wavelet Coefficients** [2]" on Signal Processing Dec 1993 stated that there would be correlations between information about an image at different resolution levels. The wavelet coefficients are encoded based on below concept as…

"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[2]."

The main concepts in EZW algorithm are as follows…

- ❖ Sub band decomposition based on hierarchy.
- ❖ Zero tree coding of wavelet coefficients which gives significant maps consisting of binary bits.
- ❖ Successive approximation quantization.
- ❖ Arithmetic multilevel coding.

A. Decomposition of sub bands:

After generating wavelet coefficients and converting it to matrix form the encoding must be done. Before encoding hierarchical ***sub band decomposition*** has to be done as shown in fig-1. Here the sub bands LL_1, HH_1, HL_1, LH_1 and every coefficient represents special area corresponding to 4 ×4 original picture. The figure is as shown below.

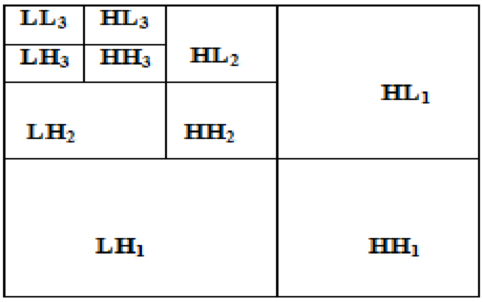

Fig2. Three scale wavelet decomposition: image divided into four sub bands.

B. Scanning order:

In order to encode the coefficients scanning has to be done and the lower sub bands should be scanned completely before going to the higher sub bands. The scan order seems to be of some influence of the final compression results. Basically there are two types of scanning procedures namely

(1) Raster scan

(2) Morton scan

In [Sha93] a raster scan order is used, while in [Alg95] some other scan orders are mentioned. Morton scan is widely used as it is more accurate and produces standard results. The coefficient is being compared with the given threshold and when all the wavelet coefficients have been reached the threshold and is lowered the image is scanned again to add more detail to the already encoded image.

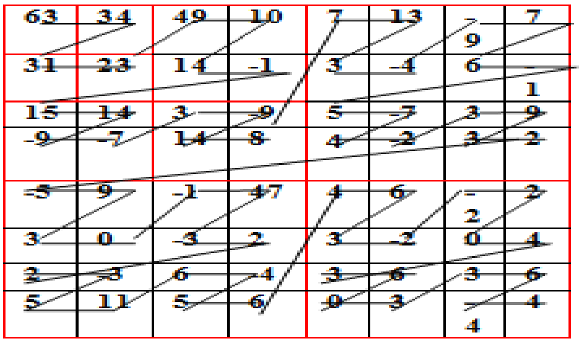

Fig3: Different wavelet coefficients

C. EZW encoding:

Now by using zero tree concepts as mentioned earlier the zero trees are formed when the coefficients are lesser than the threshold compared. coefficients should be encoded by using the following flowchart.
The following flow chart explains the encoding a coefficient of significant map which tells the position of insignificant bits at the decoder

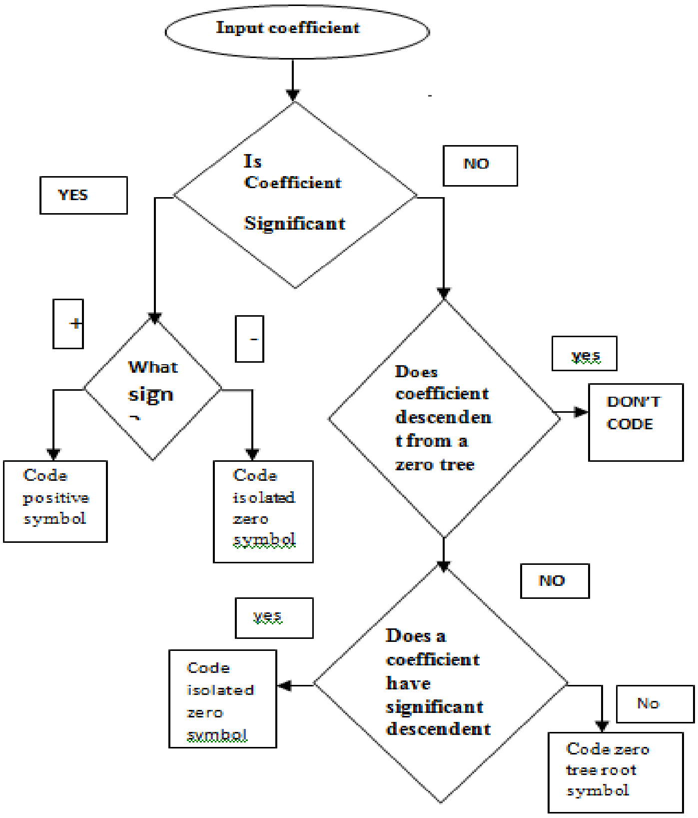

Fig4: Encoding of coefficients

In EZW, we have to maintain two separate lists.

- **Dominant List**: ***coordinates*** of coefficients not yet found significant
- **Subordinate List:** ***magnitudes*** of coefficients already found to be significant

The encoding of coefficients is as shown in the flow chart (fig. 4).
Here, the encoding of coefficients is done 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. There are two types of passes.

1. Dominant Pass (Significance Map Pass)

- Coefficients on Dominant List (i.e. currently insignificant.) are compared to threshold.
- The coefficient is verified whether it is significant by comparing with threshold.
- The resulting significance map is zero-tree coded and sent. Significance of a coefficient is given as:
- Zero tree Root
- Positive Significant
- Isolated Zero
- Negative Significant

For each coefficient that has now become significant (POS or NEG) put its magnitude on the Subordinate List (making it eligible for future refinement) remove it from the Dominant List (because it has now been found significant)

2. Subordinate Pass (Significance Coefficient Refinement Pass)

- Provide next lower significant bit on the magnitude of each coefficient on Subordinate List.
- Halve the quantizer cells to get the next finer quantizer

- If magnitude of coefficient is in upper half of old cell, provide "1"
- If magnitude of coefficient is in lower half of old cell, provide "0"
- Entropy code the sequence of refinement bits.

Now repeat with next lower threshold and stop when total bit budget is exhausted. The encoded stream is an embedded stream. We can terminate at any time before reaching the **full-rate** version.

III. SPIHT ALGORITHM

This is also a type of embedded coding. It sorts the information and corrects the error from the beginning to the end of a compressed file. If error is detected but is not corrected, the decoder simply rejects the data and displays the image prior to the errors. Here two types of data will be generated.

- ***Sorting* the information (Need of error protection)**
- Uncompressed sign and ***refinement*** bits (No need of error protection since only one pixel is effected)[3].

Here we have children for the coefficients obtained from the ***DISCRETE WAVELET TRANSFORM***. The children are again divided into grand children, great grand children, etc. In the below figure (fig.5.) we can observe that all the coefficients have four children except the first one coefficients in the LL sub band and the coefficients in the highest sub bands (LH1,HL1,andHH1)[7].

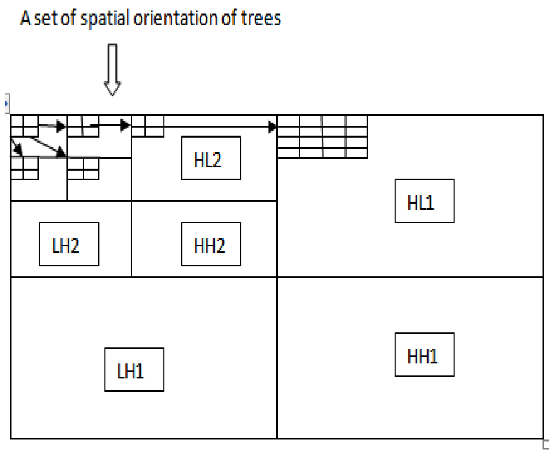

Fig5. Parent and child relationship in SPIHT

Here the coefficients are noted by (i,j). Where **'i'** represents the rows and '**j'** represents the column respectively.

O(i,j): Set of coordinates of all off springs of node (i,j) (children only)

D(i,j): Set of coordinates of all descendants of node (i,j) (Children, grand children, Great grand children, etc).

H(i,j): Set of all tree roots (nodes in the highest pyramid level) (Parents)

L(i,j): D(i,j) **minus** O(i,j) (All descendants except the offspring i.e., Grandchildren, great grand, etc).

A ***significance function*** $S_n(r')$ which decides the significance of the set of coordinates, with respect to threshold 2n is defined by:

$$S_n(r') \; \begin{cases} 1, \text{ if } Max_{(i,j)\epsilon r}\{|C_{i,j}|\}>2^n \\ 0, \text{else where} \end{cases}$$

Where $C_{i,j}$ is the wavelet coefficients. The coding starts by running two passes. In the algorithm, three ordered lists are used to store the significance information during set partitioning.

- List of insignificant sets (LIS)
- List of insignificant pixels (LSP)
- List of insignificant pixels (LIP) [4].

The algorithm will be as follows

A. Initialization:

- $n=\log_2$ (max |coefficient|)
- LIP: All the elements of H
- LIS: Set LIS= (i,j)ϵ H, where D(i,j) ≠ Ø and set each entry in LIS as type A
- LSP: Empty at beginning

B. Sorting pass:

All the coefficients are compared with the respected threshold.

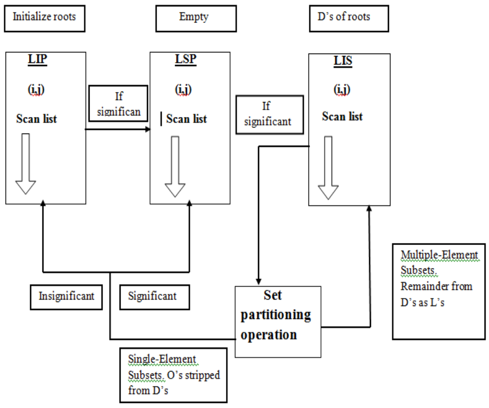

Fig6. Sorting pass in SPIHT

The sets or pixels are moved to the corresponding LIP, LSP, and LIS depending on whether the coefficient is significant or not.

The following is the example for an SPIHT algorithm. The coefficients are as follows. The significance of 0,1), (1,0),(1,1) are first tested and later the direct descendants.

0	1	2	3	4	5	6	7
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

Fig7. Example for SPIHT

C. Refinement pass:

This browses the significant coefficients (from LSP). Based on the current threshold, single bit output is obtained.

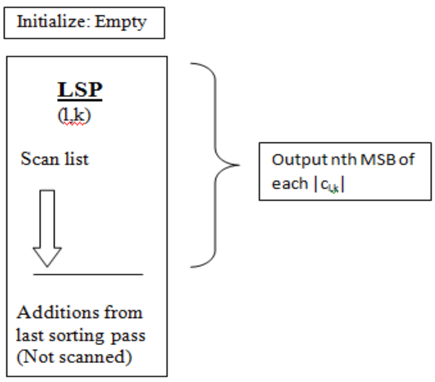

Fig8: Refinement pass

D. Quantization Pass:

After finishing two passes, the ***threshold is halved*** and with this new threshold, the two passes, Sorting pass and refinement pass are executed again. This process continues until desired number of output bits are

obtained. In the example shown in **fig. 7** the output bit stream is given as "11100011100010000001010110000".
The following figure shows the SPIHT algorithm for image compression

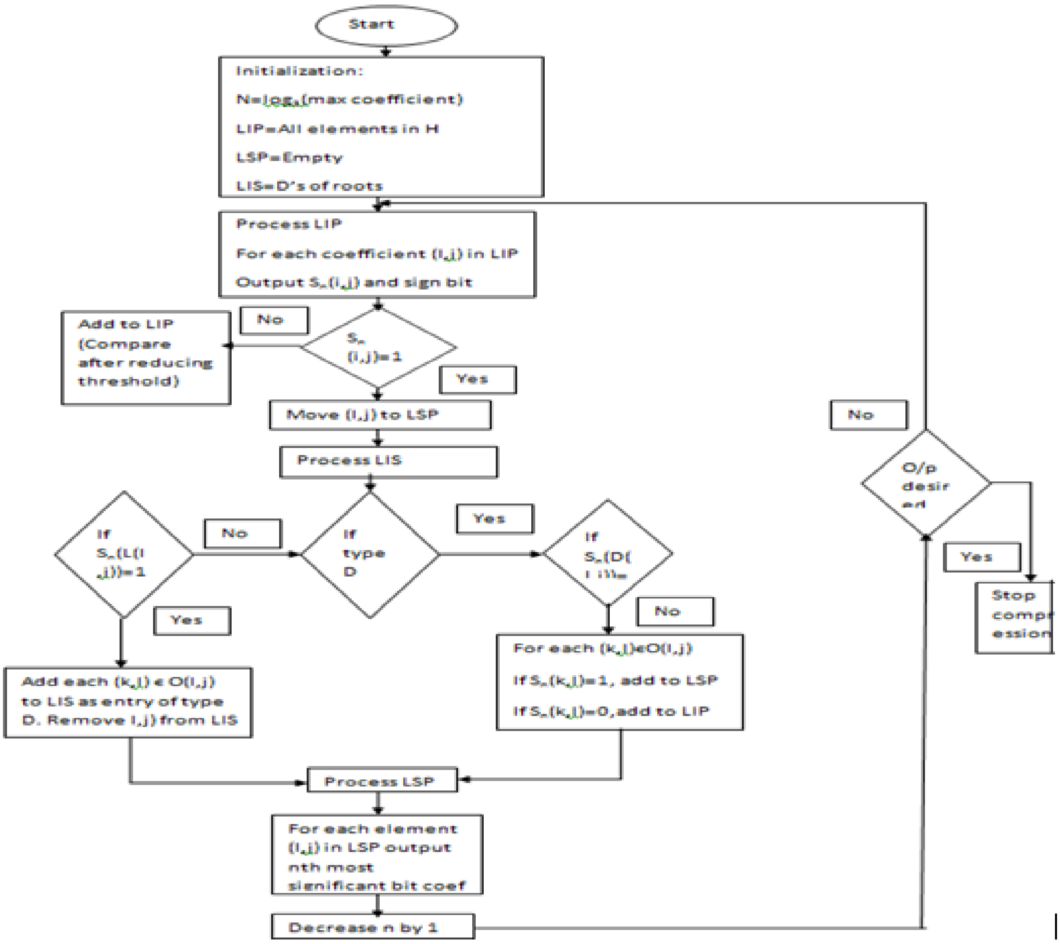

Fig9: Flow chart for SPIHT algorithm

In the above example shown in Fig.7, '000' appears with greatest probability value. Here we propose the Huffman coding for variable probabilities. First divide the every binary output stream in to 3 bits as a group; 111 000 111 000 100 000 010 101 100 00. In their process, there will have remaining 0, 1, 2 bits can not participate. The following is the bit stream.

'000' →	'01'	'100' →	'11'
'001' →	'100000'	'101' →	'101'
'010' →	'1001'	'110' →	'10001'
'011' →	'100001'	'111' →	'00'

Fig10. Bit stream in Huffman coding

The emergence of statistical probability of each symbol grouping results as follows
P('000')=0.3333 P('001')=0
P('010')=0.1111 P('011')=0
P('100')=0.2222 P('101')=0.1111
P('110')=0 P('111')=0.2222

According to above probability results, by applying Huffman encoding obtain the code word book as following table.

Number of remaining bits	Bits stream	Remain bits

Fig11. Compression in Huffman coding

Based on the compression as shown above, we can obtain a data of 25 bits unlike the 29 bits in the previous case of SPIHT. Now the newly coded data will be as given here. “10 00 01 00 01 11 01 1001 101 11 00” corresponding to previous one as “111 000 111 000 100 000 010 101 100 00”. The last two bits are left as before without further coding. This is because the Huffman coding is applied by taking three bits and the remaining two can be left as shown in figure

Advantages:
- We can interrupt decoding at any time and the result of maximum possible details can be reused with a precision of one bit.
- Used for transmission of files over net so that used with lower connection speed can download small part of file.

Disadvantages:
- Single bit error leads distortion. Hence vulnerable to bit corruption.
- Any leak during transmission of bits results in misinterpretation of bits.

IV. HUFFMAN CODING

It is an entropy coding algorithm which results in optimum code. It has highest efficiency and hence used for lossless compression. It refers the use of a variable length code table for encoding a source symbol. The probability of a symbol has a direct bearing on the length of its representation[8]. The more probable the occurrence of a symbol is, the shorter will be its bit-size representation [9].
The procedure is as follows:
- List the symbols in the order of decreasing probabilities.
- Combine the probabilities of two symbols having lowest probabilities, then set the resulting probabilities in decreasing order. The procedure is repeated till two probabilities remain.
- Start encoding the last reduction, which consists of exactly two ordered probabilities. Assign ‘0’ as first digit in the code words for all symbols associated with first probability; assign ‘1’ for the second probability.
- Go back and assign ‘0’ and’1’ to the second digit for the two probabilities that were combined in previous reduction step, retaining all assignments made in previous step.
- Keep regressing this way until the first column is reached.

V. MODIFIED SPIHT ALGORITHM

This encodes the sub band pixels by performing initialization, sorting pass and refinement pass similar to the previously discussed algorithm of SPIHT, but differences of initialization and sorting pass exists.
Advantage:
Here when the maximum value of a coefficient block put into LIB is small enough, only one bit will be used to represent it. The four coefficients of it will not be coded until the current threshold is smaller than maximum value. Thus this algorithm avoids repeated coding and early coding for non important coefficients.

VI. EXPERIMENTAL RESULTS

Computational formulae for PSNR and Mean square error are as given below

$$PSNR = 10 * \lg \left[\frac{(2^n - 1)^2}{MSE}\right]$$

$$MSE = \sum_{i=0}^{K-1} \sum_{j=0}^{M-1} \frac{\left(f(i,j) - f1(i,j)\right)^2}{M * K}$$

These results are analyzed for different wavelet families of ***biorthogonal***, ***daubechies***, and ***coiflets*** for the Lena image and the experimental results are as shown below.

threshold	Bior 4.4		Bior 6.8		Db4		Db10		Coif4		Coif5	
	Bit rate	PSNR	Bit rate	PSNR	Bit rate	PSNR	Bit rate	PSNR	Bit rate	PSNR	Bit rate	PSNR
100	0.12	23.53	0.14	23.95	0.14	23.61	0.14	23.49	0.13	23.71	0.13	23.73
80	0.12	23.53	0.14	23.95	0.14	23.61	0.14	23.49	0.13	23.71	0.13	23.73
60	0.31	26.21	0.34	26.54	0.34	26.20	0.36	26.39	0.35	26.21	0.34	26.98
50	0.31	26.21	0.34	26.54	0.34	26.20	0.36	26.39	0.35	26.21	0.34	26.98
30	0.71	27.65	0.75	27.83	0.78	27.36	0.82	27.98	0.77	27.21	0.75	28.64
20	0.71	27.65	0.75	27.83	0.78	27.36	0.82	27.98	0.77	27.21	0.75	28.64
10	1.37	27.83	1.42	27.94	1.52	27.52	1.59	28.17	1.49	27.59	1.46	28.85

Table : Bit Rate And Psnr Values Of Different Wavelets Applied To Ezw Algorithm

Bit rate	PSNR(dB)											
	Db1	Db2	Db4	Db8	Db10	Bior1.1	Bior2.2	Bior4.4	Bior6.8	Coif1	Coif4	Coif5
0.1	22.83	23.55	24.27	24.18	24.21	22.83	24.09	24.36	24.52	23.70	24.36	24.36
0.2	24.91	25.85	26.56	26.44	26.50	24.91	26.53	27.03	27.13	25.99	26.79	26.89
0.3	26.53	27.57	28.21	28.08	28.05	26.53	28.20	28.67	28.74	27.75	28.42	28.51
0.4	27.72	28.92	29.63	29.40	29.42	27.72	29.69	30.30	30.36	29.12	29.87	29.95
0.5	28.92	30.29	31.01	30.79	30.83	28.92	30.97	31.63	31.69	30.51	31.33	31.44
0.6	30.06	31.37	32.07	31.85	31.90	30.06	31.91	32.60	32.69	31.55	32.34	32.45
0.7	30.99	32.26	33.09	32.82	32.86	30.99	32.97	33.68	33.80	32.47	33.32	33.40
0.8	31.80	33.28	34.09	33.77	33.84	31.80	34.11	34.79	34.85	33.51	34.39	34.52
0.9	32.73	34.27	35.11	34.79	34.89	32.73	34.88	35.68	35.72	34.55	35.35	35.48
1	33.58	35.10	36.03	35.69	35.80	33.58	35.68	36.41	36.46	35.34	36.19	36.29

Table: Psnr Values At Different Bit Rates Of Different Wavelets Applied To Spiht Algorithm

Fig: 12 Experimental results for the image "LENA"

VII. CONCLUSION

In this paper we have implemented the two algorithms, SPIHT and EZW with Huffman encoding for different wavelet families and compared the PSNRs and bitrates of all these families. These algorithms were tested on ***Lena*** image, and from the results it can be seen that among these different wavelet families, in the ***biorthogonal*** wavelet family '***bior4.4 & bior 6.8'*** wavelet types, in the ***daubechies*** wavelet family ***'db4 & db10'*** wavelet types, and in the ***coiflet*** wavelet family ***'coif5'*** wavelet types having good PSNR at low bitrates. These algorithms have good quality and high compression ratio as compared to the previously implemented lossless image compression techniques.

REFERENCES

[1]. Surya Pemmaraju and Sunanda Mitra, "Efficient image coding using multiresolution wavelet transform and vector quantization" IEEE Southwest Symposium on Image Analysis and Interpretation, IEEE press, Santa Fe, New Mexico, April, 1996,pp:243-248.

[2]. Shapiro, "Embedded image coding using zerotree of wavelet coefficients" IEEE Transactions on Signal Processing, vol.41, pp.3445- 3462, Dec.1993

[3]. A.Said and W.A.Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees" IEEE Transactions on Circuits and systems for Video Technology, vol.6, no.3,pp.243-250,Jun.1996.

[4]. Wei Li,Zhen Peng pang "SPIHT Algorithm with Huffman Encoding" Intelligent Information technology and Security Informatics(IITSI), 2010 Third International Symposium on. 22 april 2010

[5]. Creusere, C.D., "A New method of Robust Image Compression Based on the Embedded Zerotree Wavelet Algorithm, IEEE Transactions on Image processing, 6, No. 10(1997), pp.1436-1442.

[6]. Kharate G.K., Ghatol A. A. and Rege P.P., "Image Compression Using Wavelet Packet Tree," ICGSTGVIP.

[7]. Lu and W.A. Pearlman, Dec.1998, An Efficient Low- complexity Audio Cider Delivering Multiple Levels of Quality for Interactive Applications, IEEE Workshop on Multimedia Signal Processing.

[8]. Tripatjot Singh, Sanjeev Chopra, Harmanpreet Kaur, Amandeep Kaur, "Image compression Using Wavelet and Wavelet packet Transformation" IJCST Vol.1, Issue 1, september 2010.

[9]. Kharate G.K., Patil V.H., " Color Image Compression Based on Wavelet Packet Best Tree" IJCSI International Journal of Computer Science Issues, Vol.7,Issue 2, No 3, march 2010.