Still Image Compression by Combining EZW Encoding with Huffman Encoder

Janaki. R
Assistant Professor & Head
Department of Computer Science,
N.K.R. Govt. Arts College for Women,
Namakkal - 637 001.

Dr.Tamilarasi.A
Professor & Head
Department of MCA
Kongu Engineering College,
Perundurai - 638 052.

ABSTRACT

Image compression can improve the performance of the digital systems by reducing time and cost in image storage and transmission without significant reduction of the image quality. For image compression it is desirable that the selection of transform should reduce the size of resultant data set as compared to source data set. EZW is computationally very fast and among the best image compression algorithm known today. This paper proposes a technique for image compression which uses the Wavelet-based Image Coding in combination with Huffman encoder for further compression. A large number of experimental results are shown that this method saves a lot of bits in transmission, further enhances the compression performance. This paper aims to determine the best threshold to compress the still image at a particular decomposition level by combining the EZW encoder with Huffman Encoder. Compression Ratio (CR) and Peak-Signal-to-Noise (PSNR) is determined for different threshold values ranging from 6 to 60 for decomposition level 8.

Keywords

Image Compression, Embedded Zerotree Wavelet (EZW), Huffman Encoder.

1. INTRODUCTION

Image compression is very important for efficient transmission and storage of images. It reduces the number of bits needed to represent an image by removing the spatial and spectral redundancies. Discrete wavelet transform (DWT) [2] become a cutting edge technology in image data compression. Image compression is typically comprised of three basic steps. Firstly, the image is transformed into wavelet coefficients which are then quantized in a quantizer and finally thresholded which makes the coefficient smaller than a chosen threshold value (zero) obtained from the quantizer. As a result, some bits are reduced producing an output bit stream[3].

This paper contributes to the implementation of EZW algorithm with Huffman Encoding in which approximated and detailed components are further decomposed if the sum of information content is less than the information content of component which has been decomposed[4,5].

The main contribution of EZW encoding with Huffman Encoder is that it visually improves the compression of an image by increasing the decomposition level 8 as compared to the paper “Still Image Compression using Embedded Zerotree Wavelet Encoding” by V.S. Shingate et al.[1]

2. EZW ENCODING

The EZW encoder was originally designed to operate on images (2D-signals) but it can also be used on other dimensional signals. It 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. Using an embedded coding algorithm, an encoder can terminate the encoding at any point thereby allowing a target rate or target accuracy to be met exactly[6,7]. The EZW algorithm is based on four key concepts: 1) a discrete wavelet transform or hierarchical sub band decomposition, 2) prediction of the absence of significant formation 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 Huffman encoding[8].

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 sub bands decreases as the scale decreases (low scale means high resolution), so the wavelet coefficients will, on average, be smaller in the higher sub bands than in the lower sub bands. This shows that progressive encoding is a very natural choice for compressing wavelet transformed images, since the higher sub bands only add detail[9].
2. Large wavelet coefficients are more important than small wavelet coefficients.

These two observations are exploited by encoding the wavelet coefficients in decreasing order, in several passes. For every pass a threshold is chosen against which all the wavelet 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[4].

2.1 Concept of Zerotree

A wavelet transform transforms a signal from the time domain to the joint time-scale domain. i.e. the wavelet coefficients are two-dimensional. To compress the transformed signal not only the coefficient values, but also their position in time has to be coded. When the signal is an image then the position in time is better expressed as the position in space. After wavelet transforming an image it can be represented using trees because of the sub sampling that is performed in the transform. A coefficient in a lower sub band can be thought of as having four descendants in the next higher sub band as shown in Fig.2.1 The four descendants each also have four descendants in the next higher sub band, which gives a quad-tree, with every root having four leafs[10].

A zero tree is defined as a quad-tree of which all nodes are equal to or smaller than the root and the root is smaller than the threshold against which the wavelet coefficients are currently being measured. The tree is coded with a single symbol and reconstructed by the decoder as a quad-tree filled with zeroes[11]. The EZW encoder codes the zero tree based on the observation that wavelet coefficients decrease with scale. In a zero tree all the coefficients in a quad tree are smaller than the threshold if the root is smaller than this threshold. Under this case the whole tree can be coded with a single zero tree (T) symbol[8].

A scanning of the coefficient is performed in such a way that no child node is scanned before its parent. For an N scale transform, the scan begins at the lowest frequency sub band, denoted as LLN, and scans sub bands HLN, LHN, and HHN, at which point it moves on to scale N-1 etc.[12] The two such scanning patterns for a three-scale pyramid can be seen in Fig. 2.1. Note that each coefficient within a given sub band is scanned before any coefficient in the next sub band.

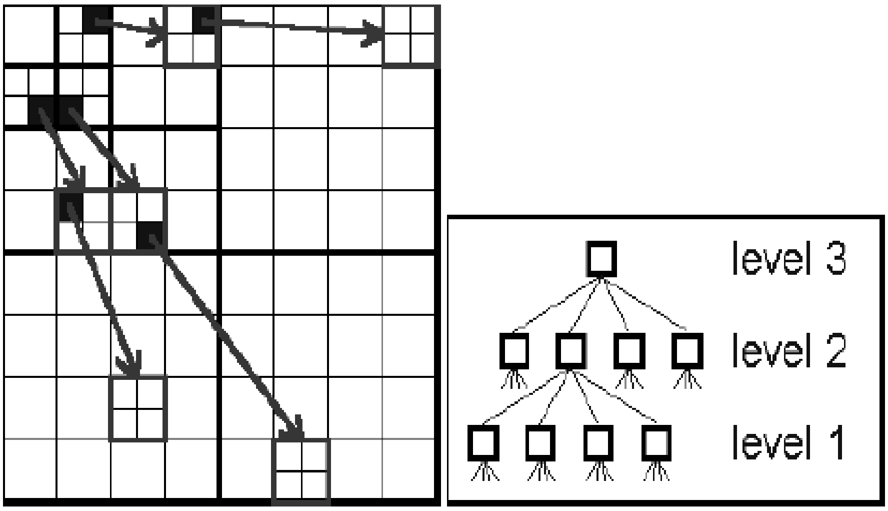


Fig 2.1 The relation between wavelet coefficients in sub bands as quad tree

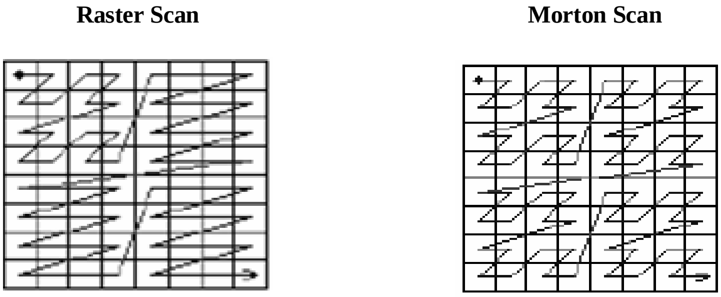


Fig 2.2 Different scanning patterns for scanning wavelet coefficients

Given a threshold level T to determine whether a coefficient is significant, a coefficient x is said to be an element of a zero tree for threshold T if itself and all of its descendents are insignificant with respect to T. An element of a zero tree for threshold T is a zero tree root if it is not the descendents of a previously found zero tree root for threshold T, i.e., it is not predictably insignificant from the discovery of a zero tree root at a coarser scale at the same threshold[13].

A zero tree root is encoded with a special symbol indicating that the insignificance of the coefficient at finer scales is completely predictable. The significance map can be efficiently represented as a string of symbols from a 3-symbol alphabet which is then entropy encoded[14].

3. HUFFMAN CODING

Huffman coding is an entropy encoding algorithm used for lossless data compression. The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. It uses a specific method for choosing the representation for each symbol, resulting in a prefix code that expresses the most common source symbols using shorter strings of bits than are used for less common source symbols.

The Huffman algorithm is based on statistical coding, which means that the probability of a symbol has a direct bearing on the length of its representation. The more probable the occurrence of a symbol is, the shorter will be its bit-size representation. In any file, certain characters are used more than others. Using binary representation, the number of bits required to represent each character depends upon the number of characters that have to be represented. Using one bit we can represent two characters, i.e., 0 represents the first character and 1 represents the second character. Using two bits we can represent four characters, and so on[11].

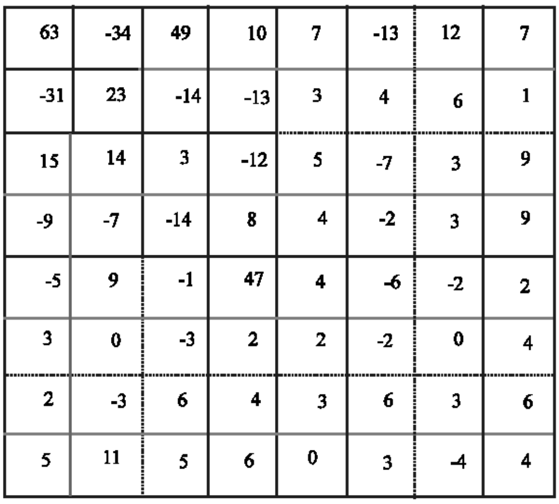
Unlike ASCII code, which is a fixed-length code using seven bits per character, Huffman compression is a variable-length coding system that assigns smaller codes for more frequently used characters and larger codes for less frequently used characters in order to reduce the size of files being compressed and transferred[15].

4. IMPLEMENTATION

Coding the wavelet coefficients is performed by determining two lists of coefficients:

1. The dominant list D contains information concerning significance of coefficients, which will be coded using Huffman encoding.

2. The significant list S contains the amplitude values of the significant coefficients, which will undergo uniform scalar quantization followed by Huffman encoding.

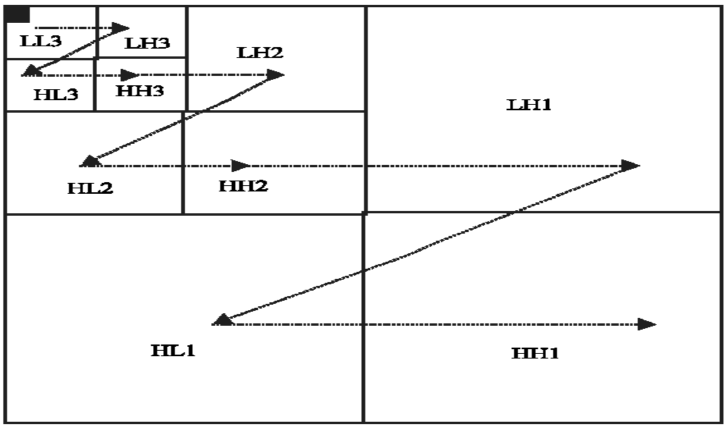


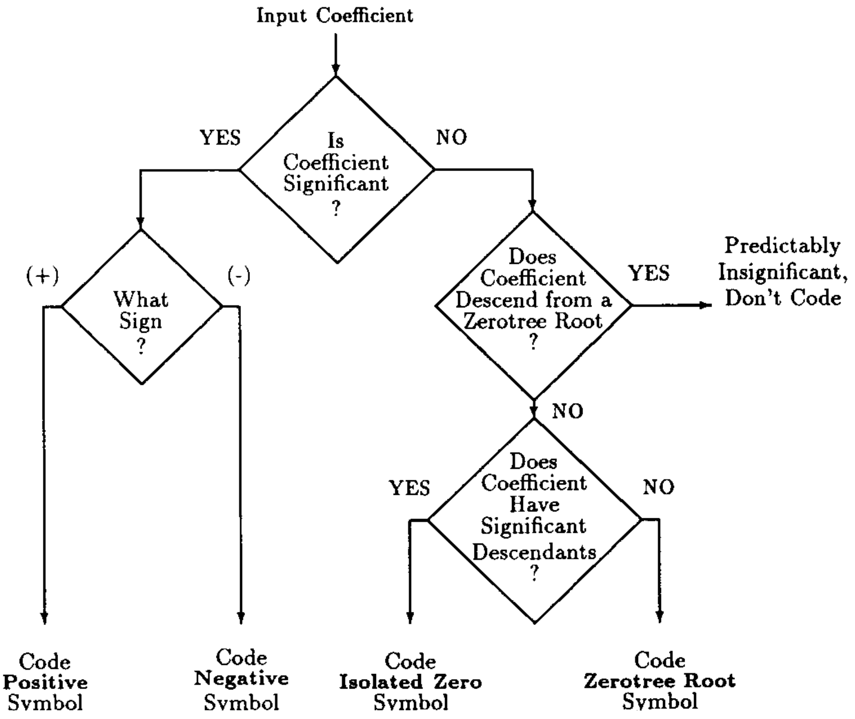
63	-34	49	10	7	-13	12	7
-31	23	-14	-13	3	4	6	1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	9
-5	9	-1	47	4	-6	-2	2
3	0	-3	2	2	-2	0	4
2	-3	6	4	3	6	3	6
5	11	5	6	0	3	-4	4

Fig.4.1 Example of decomposition to three resolution for an 8*8 matrix

Significance test :

The wavelet transform coefficients are scanned for the path as shown in the fig below. In our implemented method, we have used Mortan scan, which is more accurate and produces standard results.




Input Coefficient
Is Coefficient Significant ?
YES
NO
What Sign ?
(+)
(-)
Does Coefficient Descend from a Zerotree Root ?
YES
Predictably Insignificant, Don't Code
NO
Does Coefficient Have Significant Descendants ?
YES
NO
Code Positive Symbol
Code Negative Symbol
Code Isolated Zero Symbol
Code Zerotree Root Symbol

Flow chart for encoding a co-efficient of the significance map

EZW coding algorithm:

Each coefficient is assigned a significance symbols (P, N, Z, T), by comparing with the actual threshold.

1. P (significance and positive): if the absolute value of the coefficient is higher than the threshold T and is positive.
2. N (significance and positive): if the absolute value of the coefficient is higher than the threshold T and is negative.
3. T (zerotree): if the value of the coefficient is lower than the threshold T and has only insignificant descendants
4. Z (isolated zero): if the absolute value of the coefficient is lower than the threshold T and has one or more significant descendents.

The insignificant coefficients of the last sub bands, which do not accept descendents and are not themselves descendents of a zerotree, are also considered to be zero tree.

The significance symbols are then placed in a list D which is subjected to Huffman encoding.

The dominant list and the significance list are shown below:

D: P N Z T P T T T T Z T T T T T T T T P T T

S: 1 0 1 0

Huffman coding algorithm:

The steps involved in encoding dominant list D is as follows:

5. In the dominant list since the probability of occurrence of the symbol T is more when compared to others, this symbol should be coded with the less number of bits.

6. The other symbols probability of occurrence are less when compared to the symbol T, they should be coded with more number of bits.

7. After encoding all the symbols with binary digits, a separator is appended to the end of the encoded bit stream to indicate the end of the stream.

For eg:

In our algorithm to encode the symbols P, N, Z and T we use the binary bits as follows:

P is encoded as 1110

N is encoded as 110

Z is encoded as 10

since the probability of occurrence is less when compared to T.

T is encoded as 0 (since the probability of occurrence is more when compared to other bits)

Then we insert a separator bits i.e. a stream of 1's .Here we used 11111 to indicate the end of the bit stream.

For decoding from the encoded bit stream the dominant list symbols are found and from these symbols the pixels values of the image are predicted.

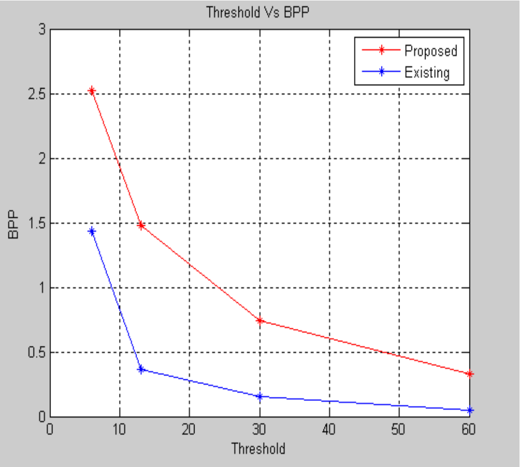
5. RESULTS AND DISCUSSION

Firstly, original image is applied to the compression program, EZW encoded image is obtain, which is further compressed using Huffman Encoding. To reconstruct compressed image, compressed image is applied to decompression program, by which EZW decoded image is obtained. Compression Ratio (CR) and Peak-Signal-to-Noise Ratio (PSNR) are obtained for the original and reconstructed images. In the experiment the original image 'Cameraman.tif' having size 256 x 256 (65,536 Bytes). The different statistical values of the image Cameraman.tif for Various Thresholds are summarized in the table.

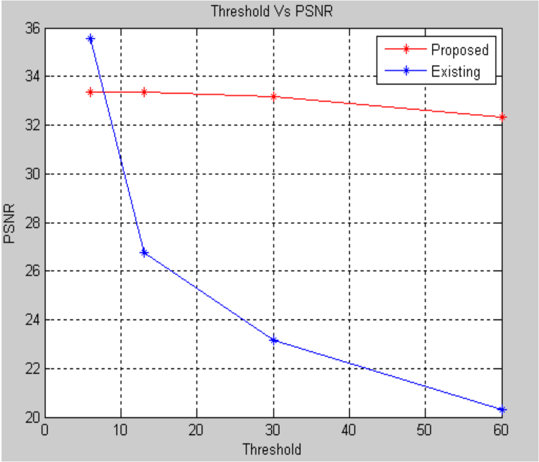
Image:Cameraman.tif   Image Size : 256*256

Parameter	**TH=6**	**TH=10**	**TH=30**	**TH=60**
Original File Size(bytes)	65240	65240	65240	65240
Compressed File Size (bytes)	11186	10870	9944	8437
Compression Ratio (CR)	5.83	6.00	6.56	7.73
Bits Per Pixel (Bpp)	2.52	1.48	0.74	0.33
Peak-Signal-to-Noise Ratio (PSNR)	33.3617	33.3704	33.1611	32.33
Encoding Time (Sec)	295.09	175.29	99.36	56.87
Decoding Time (Sec)	867.62	611.16	359.83	202.98
Total Time(Sec)	1162.71	786.45	459.19	259.85

Thus, it can be concluded that EZW encoding combined with Huffman encoder gives excellent results. By choosing suitable threshold value compression ratio as high as 8 can be achieved.



5.1 (a) THR Vs BPP



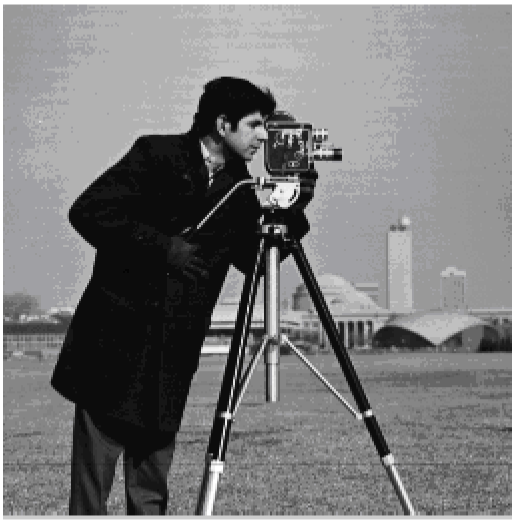
5.2 (b) THR Vs PSNR

The curves of THR Vs BPP and THR Vs PSNR have been calculated and depicted in the fig 5.1 (a) & (b) respectively. In which Image encoded using EZW algorithm for 8 level decomposition and compressed using Huffman Encoder gives better BPP and PSNR values than Image Compressed directly using Huffman Encoder.

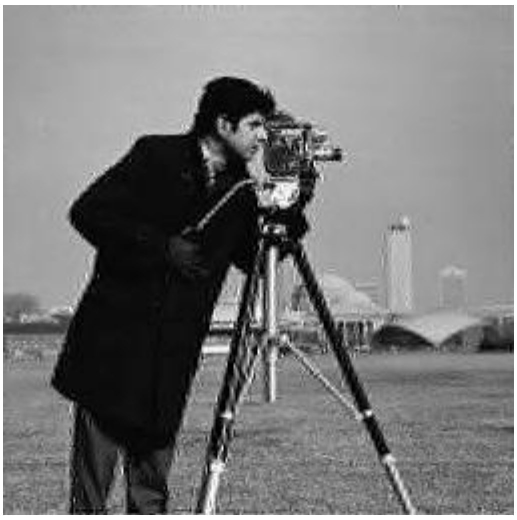
6. CONCLUSION

A technique for image compression which uses the Wavelet-based Image Coding in combination with Huffman encoder is proposed here. This approach utilizes zero tree structure of wavelet coefficients at decomposition level 8 with Huffman encoder is very effectively, which results in higher compression ratio and better PSNR. The algorithm is tested on different images, and it is seen that the results obtained by Image encoded using EZW algorithm and compressed using Huffman Encoder are consistently better than these obtained by Image Compressed directly using Huffman Encoder. It is also observed that the results are better than these reported in [1].

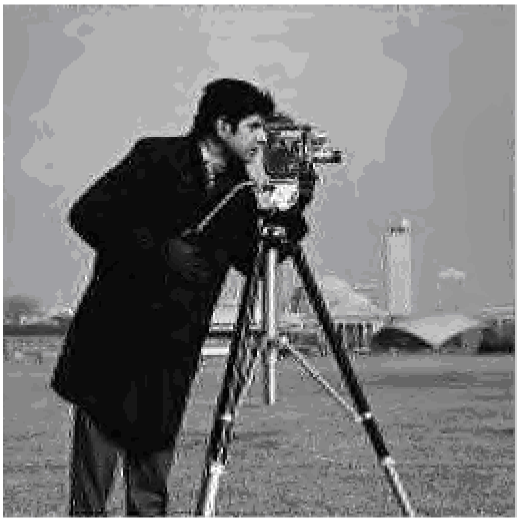
Input Image



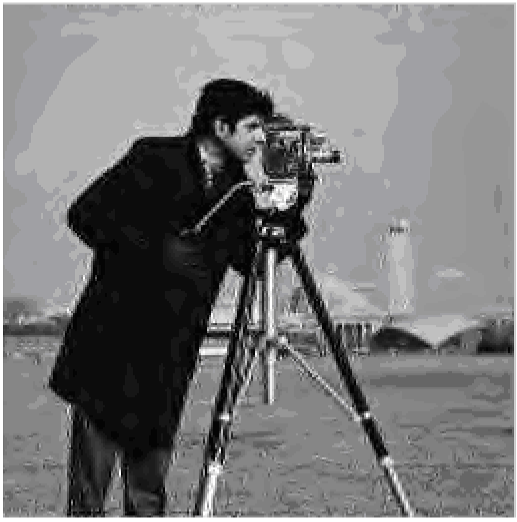
Threshold=6



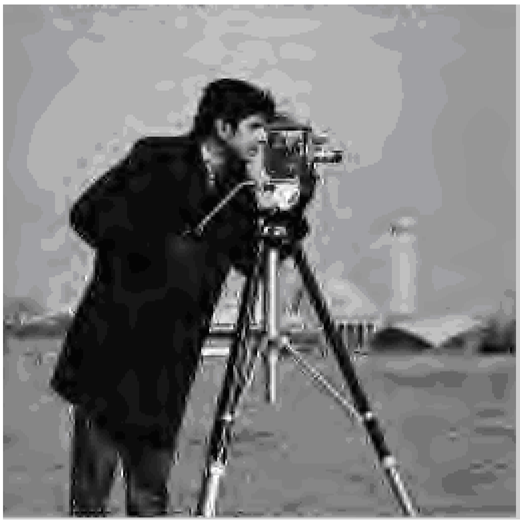
Threshold=10



Threshold=30



Threshold=60



7. REFERENCES

[1] V.S. Shingate, T. R. Sontakke & S.N. Talbar, **“**Still Image Compression using Embedded Zerotree Wavelet Encoding”, International Journal of Computer Science & Communication”, Vol. 1, No. 1, January-June 2010, pp.21-24

[2] Marcus J. Nadenau, Julien Reichel and Murat Kunt, “Visually Improved Image Compression by Combining a conventional Wavelet-Codec with Texture Modeling,” IEEE Transactions on Image Processing, Vol. 11, No. 11, Nov 2002, pp. 1284-1294.

[3] Rafael C. Gonzalez, Richared E. Woods, Steven L. Eddins, Digital Image Proceeing Using MATLAB, 2008, Pearson Education

[4] David Salomon, Data Compression - The Complete Reference, Springer,2004, 3rd edition.

[5] K. Sayood, Introduction to Data Compression , 2nd Ed, 2000 Academic Press, Morgan Kaufmann publishers.

[6] 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), p. 1436-1442.

[7] Shapiro, J. M., Embedded Image Coding Using Zerotrees of Wavelet Cefficients, IEEE Transactions on Signal Processing, 41, No. 12 (1993), p. 3445-3462.

[8] Kharate G.K., Ghatol A. A. and Rege P. P., " Image Compression Using Wavelet Packet Tree," ICGST-GVIP.

[9] S. D. Servetto, K. Ramchandran, and M. T. Orchard, "Image coding based on a morphological representation of wavelet data", IEEE Trans. Image Processing, vol. 8, pp. 1161-1174, Sept. 1999.

[10] Vinay U. Kale & Nikkoo N. Khalsa,International Journal of Computer Science & Communication "Performance Evaluation of Various Wavelets for Image Compression of Natural and Artificial Images",Vol. 1, No. 1, January-June 2010, pp. 179-184,

[11] Tripatjot Singh, Sanjeev Chopra, Harmanpreet Kaur, Amandeep Kaur,Image Compression Using Wavelet and Wavelet Packet Transformation,IJCST Vol.1, Issue 1, September 2010.

[12] Uytterhoeven G., "Wavelets: Software and Applications", U. Leuven Celestijnenlaan, Department of Computer Science, Belgium, 1999.

[13] Jerome M. Shapiro," Embedded Image Coding Using Zerotrees of Wavelet Coefficients," IEEE Transactions on Signal Processing, December 1993.

[14] Lotfi A. A., Hazrati M. M., Sharei M., Saeb Azhang, " Wavelet Lossy Image Compression on Primitive FPGA", IEEE, pp. 445-448, 2005.

[15] 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.