IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 5, Ver. IV (Sep. – Oct. 2015), PP 29-37
www.iosrjournals.org

Implementation of Hybrid Wavelet Transform for Adaptive Lossless Image Compression

Dibas Sil[1,] Bramah Hazela[2]
[1,2] Department of Computer Science & Engineering Amity School of Engineering & Technology Amity University Lucknow Campus, India

Abstract: *In digital image processing, Image compression is a type of data compression causing reduction in image size but maintain the image quality. It reduce cost and time in image storage and transmission and thus can enhance the performance of the digital system. The main purpose is to reduce and restored the volume of data such that the real image will be perfectly rebuild from the compressed image. So, a proper selection of encoding and decoding algorithm is require for image compression. Among the algorithms, EZW is one of the best image compression algorithm and is also computationally fast. Here, a method for lossless compression of image is projected which uses the Embedded Zero trees of Wavelet Transforms in combination with Huffman coding and LZW algorithm for further compression. The objective is to calculate the optimal threshold at each specific level of decomposition for the compression of a digital image. Tables of result are shown using this proposed method for 8-decomposition level of image size 256*256 which determined Ratio of compression, Bits per pixel and PSNR for each distinct values of threshold vary from 4 to 64.*
Keywords: *Image Compression, Discrete Wavelet Transform, Embedded Zero tree coding of Wavelet Transform (EZW), Huffman Coding and Lempel-Ziv-Welch(LZW) algorithm.*

I. Introduction

In digital computing, an image is a two dimensional signal defined by the mathematical function f (h, k) where h is a horizontal co-ordinate and k is a vertical co-ordinate. The value of f (h, k) at any point denotes the pixel value at that point of an image.Image compression is necessary for many applications that require huge amount of data storage, retrieval and transmission such as for documents, multimedia, medical imaging, and video

conferencing. This helps to reduce file size and capable of storing more images into limited memory space. Image compression are of two types i.e. lossy and lossless. In a lossless image compression technique, original image can be perfectly rebuild from the compressed image and compression takes place without remove any information. It is also called entropy coding. Lossless compression doesn't provide higher compression ratio but it is useful in image compression. It is basically used for image archiving purposes and also for storing legal or medical records. Lossless image compression techniques are LZW coding, Run-length coding and Huffman coding.

But using a Lossy technique of compression, the actual image will not be perfectly rebuild after compression causing reduction in image quality because most of the details are discarded without much change in the view of the image causing higher compression ratio. Lossy image compressions are become effective in applications such as television broadcasting, live video streaming or conferencing etc. Techniques for lossy image compression are Predictive Coding, Discrete Transform of Cosine (DCT), Fourier Transform and Discrete Transform of Wavelet (DWT) [3].

A. Wavelet Transformation

A wavelet is a signal having amplitude starting from zero gradually increases and further decreases to zero i.e. integrate to zero. Again, every wavelets have different compact signal representation like orthogonal or biorthogonal. A wavelet transform can be used for the decomposition of wavelet of any signal into many shifted representations of the actual source wavelet to component wavelets and the resultant
wavelet coefficient values remove fine details of the signal.
Different wavelets have different sparse representation of signals, so it is need to examine which is best suited for image compression [1], [2].

B. Embedded Zero tree Coding of Wavelet Transform

The EZW coding scheme proposed by Jerome M. Shapiro [4] is a fast, simple and effective image compression algorithm, where significant coefficients are represented in the form of symbols, resulting an embedded list of code.

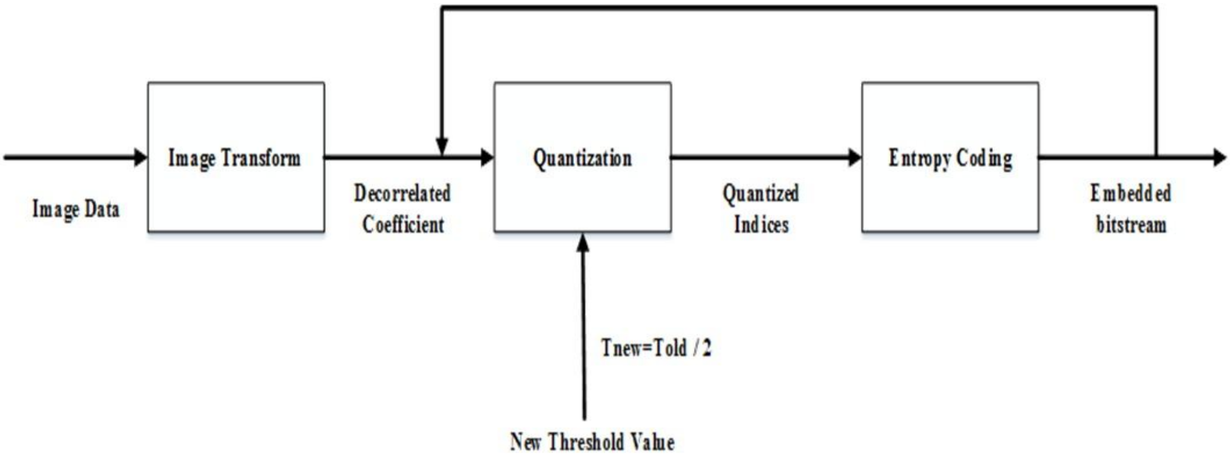


Fig.1. Embedded Encoding System

An Embedded zero tree encoder is particularly designed to perform along with wavelet coefficients. This approach was formerly used to apply on two dimensional signals like images only but now it can be utilized on signals of more dimension. Figure 1 shows the simplified diagram of embedded coding system. It is an advanced coding technique to encode an image in the form of stream of bits where each bits are produced based on significance. Like JPEG encoding technique, as the size of stream of bits increases, more details will be added in the decoded output images, a similar property. The EZW algorithm has four key features: i) a hierarchical decomposition into sub bands or a discrete transformation of wavelet, ii) zero tree coding for finding significant wavelet coefficients, iii) approximate quantization of wavelet coefficients, and iv) lossless data compression using Huffman encoding[5].

The EZW encoder can basically performs on two factors are i) regular images in general consist a low pass band. While applying wavelet transformation on an image, the sub band's energy gradually decreases along with scale but resolution increases. On average, lower sub bands have larger value of wavelet coefficients compared to the higher sub bands. Hence advanced embedded encoding scheme is an ideal choice for images having wavelet transform, because higher sub bands contain only details. ii) Small coefficients of wavelet are less significant compared to large coefficients of wavelet [6], [7].

The above factors are implemented in several passes by applying on the coefficients of wavelet in descending order. A threshold is selected in each passes whose value is half of the large wavelet coefficients to measure every coefficients of wavelet. For the greater value of wavelet coefficient against the threshold it will be encoded as significant and erased from the image but the smaller coefficient will stay for the following passes. After the completion of scanning of every coefficients, the threshold value is dropped by half and again the image is checked in the following passes to attach more necessary features to the previously coded image. This procedure works repeatedly till the end of encoding of all the coefficients.

1) Theory of Zero tree

An image after wavelet transform can be represented using trees where a coefficient of wavelet in a sub lower band can have four in the following sub higher band at each decomposition level as shown in Figure 2. Sub lower band have four offspring each of having additional four offspring in the next sub higher band, and thus forms a tree [8]. A zero tree is a simple tree shaped structure having root is equivalent or greater than every offspring but lesser than the threshold on which the coefficients of wavelet are compared. The embedded coder encodes zero tree using a particular character symbol and the decoder decodes it as a tree consists of zero values [9]. It encodes the zero based tree comparing the coefficients of wavelet with threshold. In this tree if the root and its offspring coefficients are smaller than this threshold and also all the coefficients are lesser than the threshold at that point, the entire tree will be represented with a symbol. The checking of coefficients are accomplished from parent to child node. For each M level, the check starts from sub lowest
band, represented as LLM, to sub other bands HLM, LHM, and finally to highest sub band HHM, till the level M-1 etc. [4]. A three level tree of such patterns is shown in Figure 2. Here, notice each coefficient of wavelet in the next higher band is checked after the coefficient of particular lower band. A coefficient is significant or not can be determine by comparing with a threshold value TH which is half of the largest coefficient, a coefficient is a zero root tree against the threshold TH if root and its successors cannot be significant for TH. Thus all descendants of a zero root tree for threshold TH contain zeroes for the particular pass which will be recheck in the next passes [10].

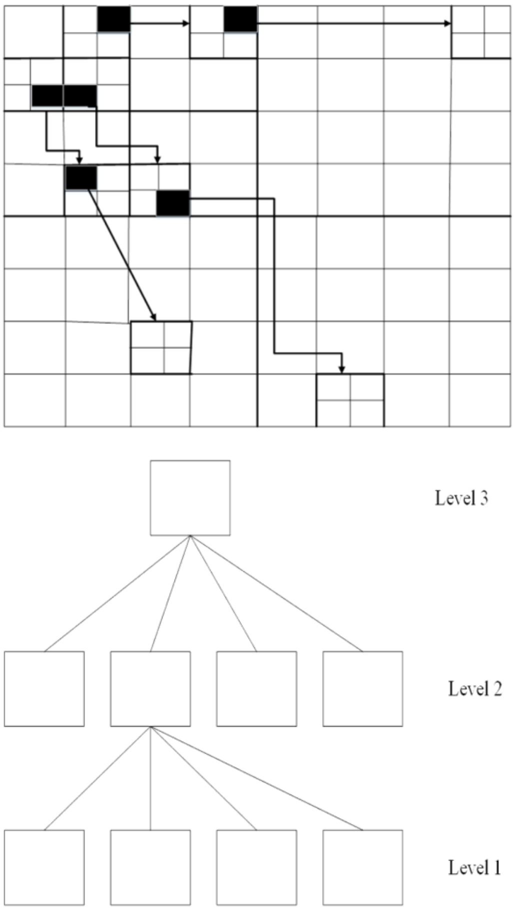


Fig.2. Representation of Wavelet Coefficient in different sub bands as a Quad Tree

C. Huffman Coding

Huffman coding is a coding technique suggested by Dr. D.

A. Huffman in the year 1952 used to create least redundant code. This coding is an efficient entropy based coding technique commonly required for lossless compression of data. It can used any types of data. It uses mutable size code list to encode a symbol depend on the possibility of appearance of symbol. This encoder use a particular technique to encode each symbol in the form of precede code to represent the symbols as a short code word whose size is smaller than the size of original source symbol [12]. The Huffman algorithm is one kind of statistical coding, where length of prefix code of each symbol depends on its probability of existence [13]. The length of prefix code of each symbol is inversely proportional to its probability of appearance i.e. higher probability of appearance of a symbol means shorter bit size of prefix code. This algorithm uses mutable length code and assigns lesser code for common symbols and longer code for rare symbols for the reduction of file size and help in compression. Thus for more and less frequent symbols in symbol stream, Huffman coding generates small and large codes causing a formation of bit stream [17].

D. Lempel Ziv Welch (LZW) Coding

The first original approach to data compression was published by Lempel Ziv in the year 1977, which was replaced by an alternative approach in the year 1978. Again further Terry Welch's modified the algorithm and was published in the year 1984. LZW compression algorithm is very simple. It substitute input strings of characters with single code word [18]. It doesn't perform any analysis, but simply check the presence of input string of characters in the string table and if presence, then output the code of prefix string of characters, otherwise includes latest string to the table of strings. Hence the compression takes place when a short code word is resulted as a replacement for large string [19]. Due to LZW algorithm, the code words generated can be of any length, but it resulted the code word whose size is less than the size of original input string of characters. The first 256 ASCII codes are by default assigned in the string table and as the algorithm proceeds the remaining codes are assigned to the string table [20]. Now implementation of lossless image compression is performed using the above three encoder and the aim is to calculate the optimal threshold at each specific level of decomposition for the compression of a digital image.

II. Implementation

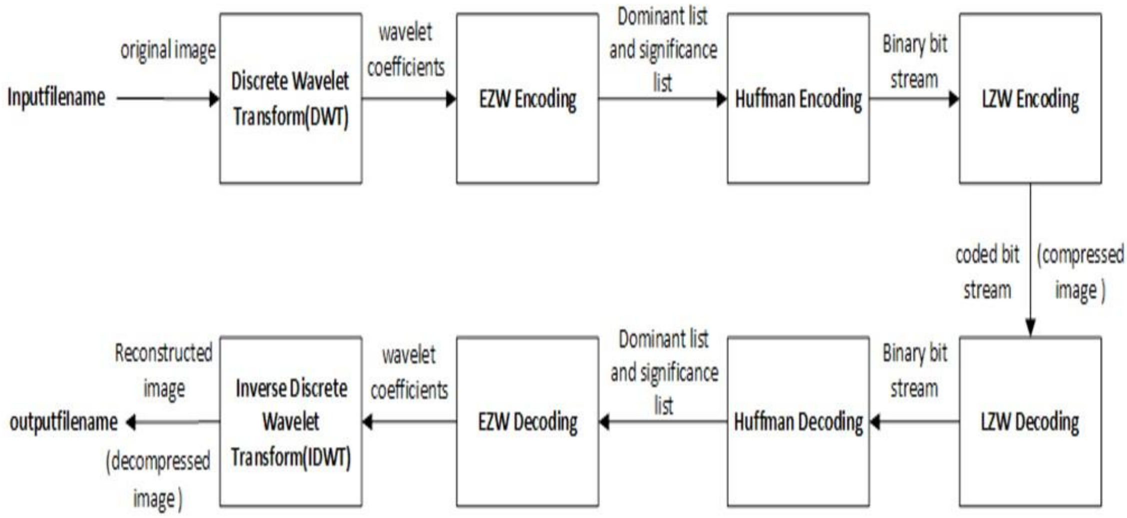


Fig.3. Hybrid Method

Steps of hybrid method is shown above in the Figure 3 are described below:

Step1: Apply bi-orthogonal wavelet decomposition using discrete wavelet transform (DWT) takes place on original image and generate the wavelet coefficients for corresponding pixel values.
Step2: Apply EZW encoding algorithm, first consider a decreasing sequence of thresholds (THo, TH1,….,THN-1) with THi = (THi-1)/2 and the absolute value of coefficient is less than twice of THo.

i. Perform dominant pass to store values of coefficients in the form of symbols as shown in Figure 4 in the dominant list.
ii. Perform subordinate pass to store the magnitude of coefficients already found to be significant in the form of binary bits (0 and 1) in the subordinate list.

Step3: Repeat Step 2 for next lower threshold value until it reached the user defined threshold value.
Step4: Encode the dominant list using Huffman encoding algorithm like if probability of appearance of a symbol in the dominant list is more than the others, then this symbol should

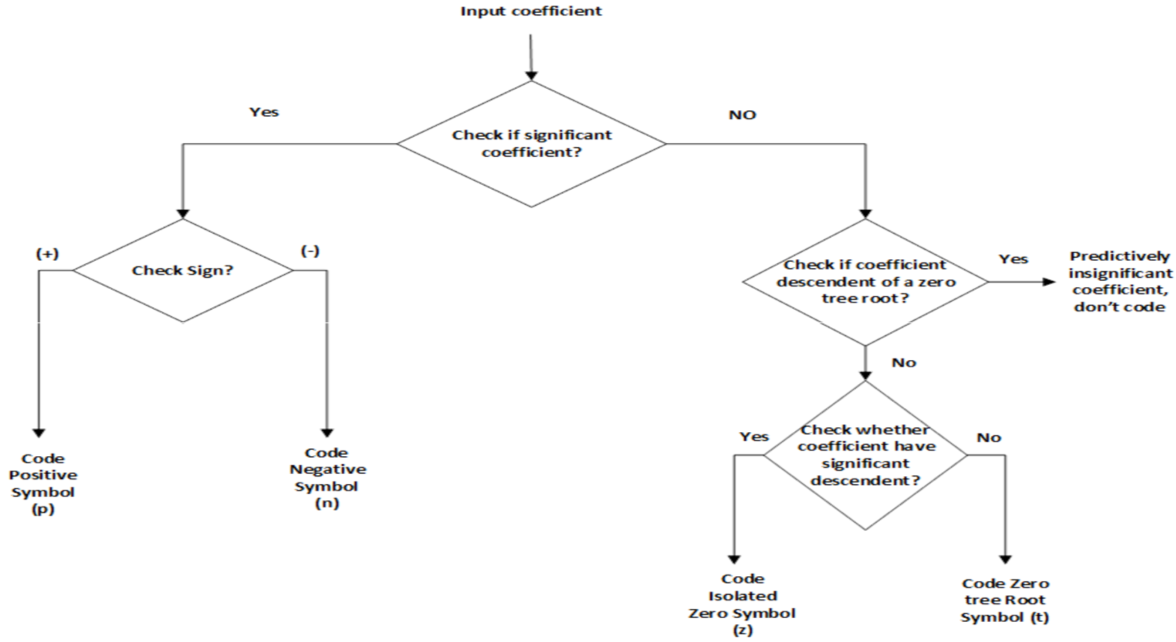


Fig.4. Flow chart for the dominant pass

Be encoded with the minimum size of bits. Again for lower probability of appearance of a symbol in the dominant list, the symbol will be encoded with the maximum size of bits and at the end, a divider is added to specify the completion of stream of bits. In my approach, the positive symbols is replaced with three ones and one zero, the negative symbols is replaced with two ones and one zero, isolated zero symbol is replaced with single one and zero and since zero root tree symbol has highest probability of appearance, hence it is replaced with single zero. Then after a divider consists of four consecutive ones is added to specify the completion of stream of bits. [7].

Step 5: Apply LZW encoding algorithm on the bit stream. Consider a code list consists of first 256 ASCII codes of
characters,

Step 5.1: Initialize, a variable WORD for storing first input character;
Step 5.2: Check the presence of any other input characters, if present, then store next input character into another variable SYMBOL;
Step 5.3: Check if WORD and SYMBOL together present in the code list then store both the WORD and SYMBOL together in the WORD variable else output the previous code number for WORD, add both WORD and SYMBOL together to the code list and store the value of SYMBOL into WORD variable;
Step 5.4: Repeat step 5.2 for all input characters otherwise stop and output the code for WORD;
Step 6: The coded bit stream generated due to LZW encoding algorithms is decoded to binary bit stream using LZW decoding algorithm as follows:
Step 6.1: first read the old code;
Step 6.2: output the old code;
Step 6.3: store the old code in a variable SYMBOL;
Step 6.4: Check the presence of any other input characters, if present, then read the latest code;
Step 6.5: Check if latest code is not in the code list, then store the value of old code along with WORD and SYMBOL together in a variable WORD else store the value of latest code in WORD variable and output the value of WORD variable;
Step 6.6: Store the first character of WORD in SYMBOL variable;
Step 6.7: Add values of old code and SYMBOL variable together to the code list;
Step 6.8: Replace latest code in old code
Step 6.9: Repeat step 6.4 for all input characters;
Step 7: The decoded binary bit stream are then subjected to Huffman decoding algorithm to generate dominant and subordinate list and further using EZW decoding algorithm, it recovered to wavelet coefficients which again reconstructed to image using inverse discrete wavelet transform.

III. Results And Discussion

TABLE I. The numerical results for different thresholds applied on the **saturn.bmp** image of size **256*256** as shown below:

Parameter	**TH-4**	**TH-16**	**TH-32**	**TH-64**
Actual File Size in bytes	66560	66560	66560	66560
File Size after Compression in bytes	7410	2882	1647	904
Compression Ratio (CR)	11.307	4.398	2.513	1.379
Bits Per Pixel (Bpp)	0.90	0.35	0.20	0.11
Peak-Signal-to-Noise Ratio (PSNR)	38.98	37.70	35.28	31.93
Encoding Time (Sec)	375.9460	95.7337	34.8961	16.4361
Decoding Time (Sec)	213.1085	93.1313	52.7021	27.5211
Total Time(Sec)	589.0545	188.8650	87.5982	43.9572

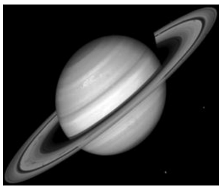


Fig.5. Original Image

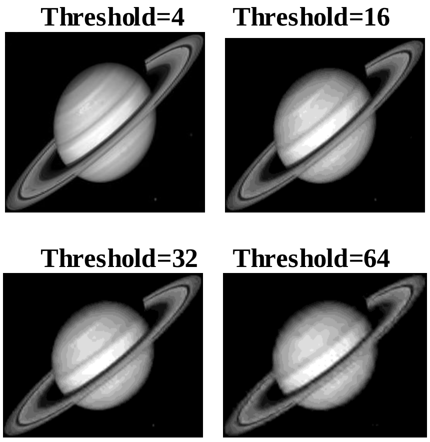


Fig.6. Reconstructed Image.

TABLE II. The numerical results for different thresholds applied on the **girl.bmp** image of size **256*256** as shown below:

Parameter	**TH-4**	**TH-16**	**TH-32**	**TH-64**
Actual File Size in bytes	65792	65792	65792	65792
File Size after Compression in bytes	13716	5100	2863	1476
Compression Ratio (CR)	20.929	7.782	4.369	2.252
Bits Per Pixel (Bpp)	1.67	0.62	0.35	0.18
Peak-Signal-to-Noise Ratio (PSNR)	31.12	31.03	30.79	28.60
Encoding Time (Sec)	1317	241.4827	108.9147	33.0219
Decoding Time (Sec)	374.7391	187.2177	124.4143	53.1388
Total Time(Sec)	1691.7391	428.7004	233.3290	86.1607

Fig.7. Original Image

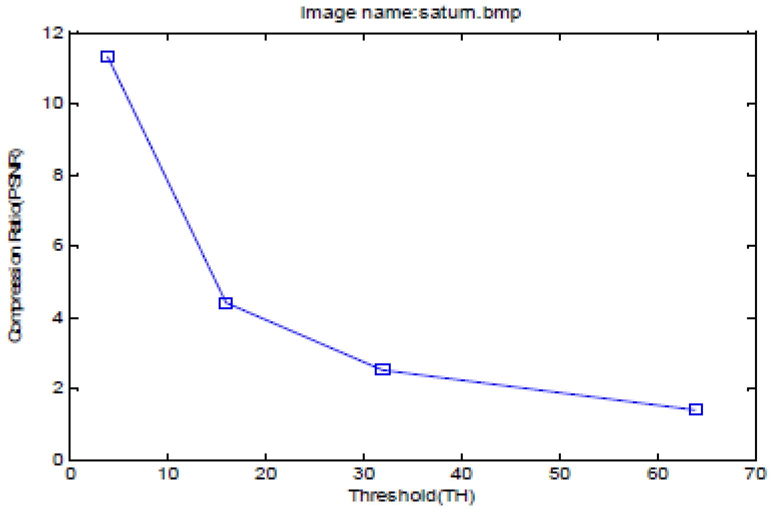
Threshold=4

Threshold=16

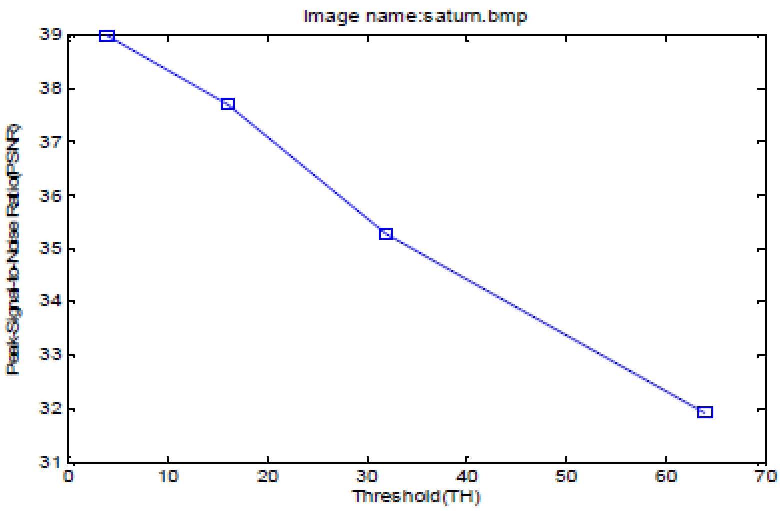
Threshold=32

Threshold=64

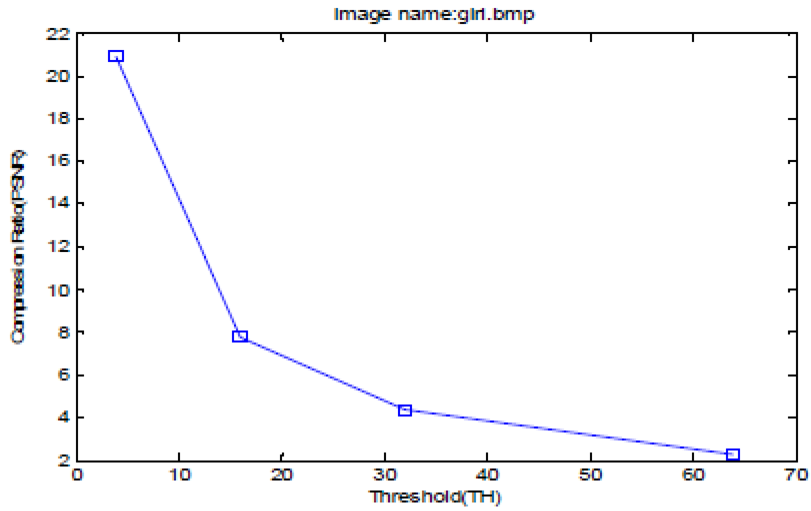
The graphical representation of Threshold vs Compression Ratio of the image **saturn.bmp** as shown below:



The graphical representation of Threshold vs PSNR of the image **saturn.bmp** as shown below:



The graphical representation of Threshold vs Compression Ratio of the image **girl.bmp** as shown below:



The graphical representation of Threshold vs PSNR of the image **girl.bmp** as shown below:

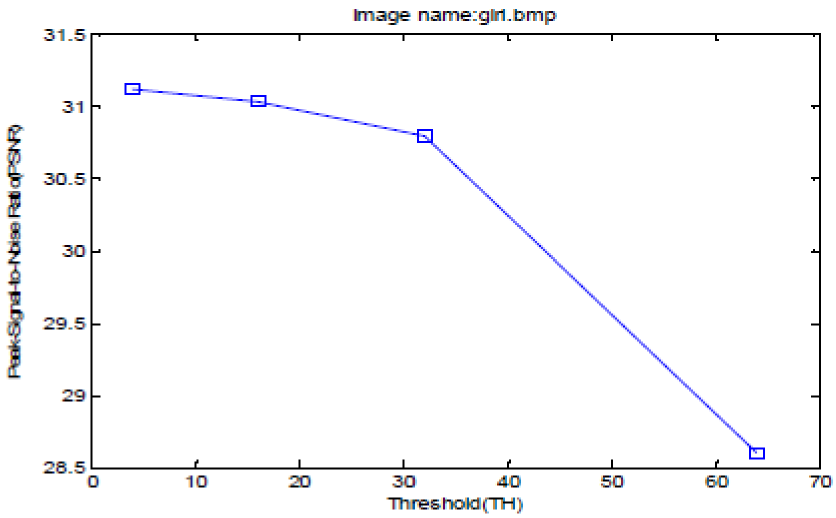


Fig.8. Reconstructed Image

IV. Conclusion And Future Scope

A hybrid compression technique that uses EZW coding in along with Huffman coding and Lempel Ziv Welch coding is proposed here. This methodology implemented discrete wavelet transform to generate wavelet coefficients which will applied to EZW coding algorithm and formed encoded symbols of wavelet coefficients using zero tree structure at 8- decomposition level which will again implemented by Huffman coding and produced binary bit stream, which will further applied to LZW coding and resulted a compressed image with higher compression ratio and much better PSNR value [6]. This proposed hybrid method is applied on various images, and the results obtained by using these three algorithms are consistently much better compared to the standalone algorithm. It yields good quality of the rebuilt image with higher ratio of compression and low bit per pixel. Again, this system performs well with less complexity and decompressed image visual quality will be getting improved with lower threshold value [21]. In future, the focus is to improve performance of this system and also compression ratio with another new techniques. Again, this proposed method can be experimented on different kinds of data compression like audio, video, text but at this moment it is applicable to images only.

Acknowledgement

I would like to acknowledge and give credits to the following individuals who assisted me for the completion of this research paper. I am very thankful to the Head of the Department of Computer Science, **Dr. Deepak Arora,** to encourage me and provided all kinds of facilities for the completion of this thesis. I also like to thanks my guide, **Mr. Bramah Hazela,** whose support and guidance helped me to complete the work**.** Finally a special thanks to my friends and family and also God, who made everything possible.

References

[1]. David Salomon, Data Compression - The Complete Reference, Springer, 2004, 3rd edition.
[2]. K. Sayood, Introduction to Data Compression , 2nd Ed, 2000 Academic Press, Morgan Kaufmann publishers.
[3]. Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Digital Image Processing Using MATLAB, 2008, Pearson Education
[4]. Shapiro, J. M., "Embedded Image Coding Using Zero trees of Wavelet Coefficients", IEEE Transactions on Signal Processing, 41, No. 12 (1993), p. 3445-3462.
[5]. Kharate G.K., Ghatol A. A. and Rege P. P., "Image Compression Using Wavelet Packet Tree," ICGSTGVIP.
[6]. V.S. Shingate, T. R. Sontakke & S.N. Talbar, "Still Image Compression using Embedded Zero tree Wavelet Encoding", International Journal of Computer Science & Communication, Vol. 1, No. 1, January-June 2010, pp.21-24.
[7]. Janaki. R, Tamilarasi. A, "Still Image Compression by Combining EZW Encoding with Huffman Encoder", International Journal of Computer Applications, Vol 13, No. 7, January 2011.
[8]. 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.
[9]. Tripatjot Singh, Sanjeev Chopra, Harmanpreet Kaur, Amandeep Kaur, "Image Compression Using Wavelet and Wavelet Packet Transformation", IJCST Vol.1, Issue 1, September 2010.
[10]. Creusere, C.D., A New Method of Robust Image Compression Based on the Embedded Zero tree Wavelet Algorithm, IEEE Transactions on Image Processing, 6, No. 10 (1997), p. 1436-1442.
[11]. Vartika Singh, "A Brief Introduction on Image Compression Techniques and Standards", International Journal of Technology and Research Advances Vol. 2, Issue 2, January 2013.
[12]. Mamta Sharma, "Compression Using Huffman Coding" IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010.
[13]. Jagadeesh. B, Ankitha Rao, "An approach for Image Compression Using Adaptive Huffman Coding", International Journal of Engineering Research & Technology, Vol. 2, Issue 12, December 2013.
[14]. Saravanan. C, Ponalagusamy. R, "Lossless Grey-scale Image Compression using Source Symbols Reduction and Huffman Coding".
[15]. Aarthi. R.S., Muralidharan. D, Swaminathan. P "Double Compression of Test Data using Huffman Code", Journal of Theoretical and Applied Information Technology, 15 May 2012.
[16]. Alarabeyyat A, Al-Hashemi1. S, Khdour. T, Hjouj Btoush. M, S. Bani- Ahmad. S, Al-Ha shemi. R, "Lossless Image Compression Technique Using Combination Methods", Journal of Software Engineering and Applications,2012.
[17]. Pujar. H, Kadlaskar. L. M, "A New Lossless Method of Image Compression and Decompression Using Huffman Coding Technique", Journal of Theoretical and Applied Information Technology, Vol. 15, No. 1, 2010.
[18]. Md. Rubaiyat Hasan, "Data Compression using Huffman based LZW Encoding Technique", International Journal of Scientific & Engineering Research, Vol. 2, Issue 11, November 2011.
[19]. Monika and Pragati Kapoor, "A Huffman based LZW Approach to Perform Image Compression", 2007.
[20]. Saravanan. C, Surender M "Enhancing Efficiency of Huffman Coding using Lempel Ziv Coding for Image Compression", International Journal of Soft Computing and Engineering, January 2013.
[21]. Vinay U. Kale, Shirish M. Deshmukh, "Visually Improved Image Compression by Combining EZW Encoding with Texture Modeling using Huffman Encoder", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No. 11, May 2010.