Enhanced Listless Block Tree Coding with Discrete Wavelet Transform for Image Compression

Chandandeep Kaur
Assistant Professor
Chandigarh University
Gharuan

Rana Gill
Assistant Professor
Chandigarh University
Gharuan

Dilpal Singh
Assistant Professor
Chandigarh University
Gharuan

ABSTRACT

Set Partitioning in Hierarchal Trees (SPIHT) is an efficient method for compressing images under low bit rates. No List SPIHT (NLS) and Wavelet Based Block Tree Coding (WBTC) are two enhanced algorithms of SPIHT. The WBTC algorithm works on blocks instead of pixels in SPIHT. The size of root block in WBTC varies from one step to another. This reduces the memory requirement to a great extent. NLS uses markers instead of lists used for the storage of coefficients in SPIHT. The three lists used in SPIHT to manage the significant coefficients grow exponentially with each step as more number of coefficients is tracked. Due to this feature SPIHT algorithm requires a lot of memory management and hence it is complex for hardware implementation. But the 8 different markers used in NLS removes this drawback of original algorithm. Listless Block Tree Coding algorithm (LBTC) is evolved by combining the WBTC and NLS algorithms. In this algorithm image compression is performed on the block basis and the significant coefficients are tracked with the help of different markers. The LBTC algorithm when combined with Discrete Wavelet Transform (DWT) performs even well in the terms of Peak Signal to Noise Ratio (PSNR) and Mean Square Error (MSE). In this paper arithmetic encoding is applied on the LBTC-DWT algorithm which further enhances the compressed image quality in terms of PSNR and MSE though the time taken increases.

Keywords

SPIHT, Block tree, NLS, LBTC, DWT, Arithmetic Encoding.

1. INTRODUCTION

Said and Pearlman in 1996 introduced an efficient method for the compression of images- SPIHT [1]. The algorithm uses the concept of Set Partitioning [2]. It partitions an image on the basis of parental hierarchy and hence forms various hierarchal sets. Thus at the root of tree there is a great grand pixel known as root and it has further many descendents depending upon the size of image. So there is a parent- child relation existing in the hierarchal tree which helps to explore the spatial self- similarity among the pixels at similar scales. According to the quantisation levels, pixels are checked for significance starting with the root pixel. Each set of pixels is compared with different threshold which decreases at each level.

The significant pixels are stored in three lists- List of Insignificant Sets (LIS), List of Insignificant Pixels (LIP), List of Significant Pixels (LSP). As the more and more number of coefficients are tracked, these lists grow in size hence rendering the algorithm very complex in terms of memory management. As the SPIHT algorithm uses the concept of hierarchy so if a pixel at higher level is insignificant i.e a zero tree [3], its descendants are also insignificant hence saving the time required to check all the pixels. SPIHT algorithm offers error protection to a large extent [4] and applies progressive coding i.e. progressive image transmission is possible with this concept [5]. But the algorithm is error prone and the error in a single bit can lead to misinterpretation of the whole image due to bit synchronisation property. So many other algorithms evolved from SPIHT by improving in terms of various factors as compression ratio [6], memory requirement [7, 8], speed [9, 10], complexity [11], quality of compressed image [12], error resilience [13, 14, 15], redundancy [16], etc. This paper concentrates on the LBTC [17] algorithm which is evolved by combining the concepts of two improved versions of SPIHT - NLS and WBTC. Another paper introduces the LBTC algorithm combined with DWT [18].

No List SPIHT (NLS) [19] is an advanced version of SPIHT which marks the significant coefficients by using different markers. It does not make use of lists as in original SPIHT. By using markers to track the significant coefficients, NLS becomes an efficient algorithm in terms of memory requirement and complexity. NLS algorithm uses a state table [20] to track the set partitions. The state table has 4 bit per coefficient Tracking of the coefficients is not only easier but also at a faster pace. With increase of significant coefficients at each level, the zero distortion reduces [21]. Thus it reduces the complexity of SPIHT algorithm and also increases the speed of execution. But the drawback of NLS algorithm is that it does not perform that efficiently at lower bit rates.

This paper gives a brief introduction of the SPIHT algorithm and its improved versions- NLS and WBTC. Another algorithm LBTC is also introduced which combines the concept of above two algorithms. The paper also presents the combination of LBTC algorithm with DWT and arithmetic encoding. The results and conclusions are analyzed in the last section.

2. WAVELET BASED BLOCK TREE CODING

Wavelet Based Block Tree Coding (WBTC) [22] algorithm is an improved version of SPIHT algorithm in terms of memory requirement, speed and complexity. The original SPIHT algorithm performs the significant test on the each and every pixel of the input image while following the hierarchal tree. It starts with the parent pixel (root) and goes down for checking the significance of branch pixels. Hence it follows a certain pattern known as Spatial Orientation Tree (SOT) [23] while performing the significance test. If the parent pixel is insignificant, the child pixel is also insignificant hence can be skipped thus forming a zero tree. The WBTC algorithm uses the concept of Zero Tree along with the Zero Block. The WBTC algorithm divides the input image into blocks of varying sizes as 2x2, 4x4 and 8x8. All the blocks are checked for significance starting with root block of varying sizes by using the concept of SPIHT. The insignificant block is marked as Zero Block and the significant block is further checked for the significance of individual pixels by using the concept of Quad- Tree Partitioning [24, 25]. So the SPIHT algorithm has various SOT's while the WBTC has only a single SOT. WBTC uses three lists to track the significant coefficients similar to SPIHT. Hence the WBTC algorithm is useful in efficient memory management as instead of Zero Trees, it keeps Zero Blocks which are made up of many Zero Trees [26]. By working on blocks (a set of pixels) instead of individual pixels the algorithm is executed at a faster speed as compared to the original SPIHT algorithm. WBTC algorithm uses wavelet transform [27] for the decomposition of an image. The results vary depending upon the choice of wavelet. The complexity of WBTC is decided by the level of decomposition of an image and the depth to which the algorithm is applied. More the number of decomposition levels, more the complexity of algorithm increases. But smaller the size of root block more is the quality of compressed image. So while applying the WBTC algorithm if the quality is increased, the complexity is also increased and vice- versa. Due to use of lists in the algorithm, the complexity increases as more and more number of coefficients are tracked as the list nodes grow exponentially with each level. This feature makes the WBTC algorithm undesirable for hardware applications. So if instead of lists, some markers of fixed size are used to track the significant coefficients, the algorithm is even more efficient. The same is discussed in the next section.

3. LISTLESS BLOCK TREE CODING WITH DWT

The combination of WBTC and NLS is seen in this algorithm. The LBTC algorithm divides an image into blocks and performs block processing on the respective blocks. It does not use lists to store the significant coefficients but the markers are used to keep their track. Eight different markers are used depending upon the level of their significance. The algorithm exploits linear indexing by using one- dimensional array to scan the significant coefficients. Mortan Scan Sequence is used for exploring the coefficients linearly in one dimension.

Discrete Wavelet Transform (DWT) [28] decomposes an image into blocks and the algorithm is applied on each block. In this paper the DWT is applied using fast lifting approach which applies filters on various blocks. Use of filters helps in saving the computations and memory requirement. DWT divides an image into four blocks i.e four subbands are formed, LL, LH, HL and HH band.

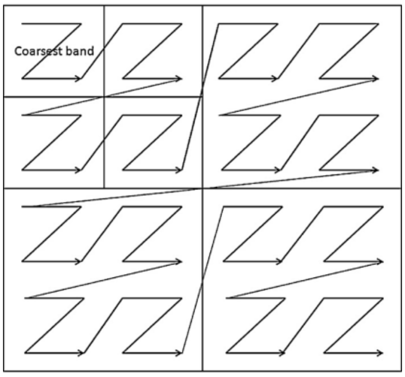

Fig 1: Mortan Scan Sequence

The coarse band i.e LL band is further subdivided into four bands by applying DWT at higher levels. While LL band has approximation details, the other bands represent details of an image across various directions. Depending upon the level of DWT applied, an image is divided into various detailed levels which may be upto 2- level or more decompositions. The enhanced LBTC algorithm used in this paper uses 4 markers as compared to 8 markers used in the previous techniques.

The four markers used in this algorithm are:

MIP: marks insignificant pixels

MNP: marks newly significant pixels

SFC: significant first child in a set of tree consisting of descendants

SFG: significant first grandchild in a tree of grand descendants

The use of lesser number of markers for the storage of significant coefficients as compared to previous techniques results in saving memory requirement and this feature also increases the speed of execution of the algorithm. The algorithm works in four steps after setting a threshold which is reduced at each level of the hierarchal tree by a factor of 2. Initial threshold set for the root block of an image is

$$T= \log2 \max(c\ (i,j))$$

A coefficient at the root level is significant if its magnitude, $s \geq 2^T$.

3.1 Psuedo Code for LBTC- DWT with Arithmetic Encoding

3.1.1 Checking for Insignificant Pixels

- check the value of pixel at root

- pixel value is compared with the already threshold

- if pixel value lesser than threshold, it is insignificant and marked as MIP

- further check the significance at the next levels of tree and the insignificant pixels marked as MIP

- if pixel significant, mark it as MNP and move to next coefficient

- otherwise move to next block

3.1.2 Checking for Insignificant Sets

- block of descendants in the tree are checked for significance

- the pixels in the block are compared with the changing threshold at each level

- if significant, it is marked as MNP and move to the next block

- if block is insignificant, it is skipped and marked as zero block

- significant block is further quad splitted into four blocks and each of the block undergoes through significance test

- if the descendant block significant, it is marked as SFC otherwise skip to the next descendant block

- the significant descendant block is further quad splitted and each of the grand descendant block goes through significance test

- significant grand descendant block is marked as SFG, else move to the next grand descendant block

- the process is carried out until the algorithm reaches to a single coefficient

- significant pixel is marked as MNP while the insignificant pixel is marked as MIP

3.1.3 Arithmetic Encoding of Significant Pixels

- the significant pixels marked in the previous step are refined here

- the pixels marked insignificant are skipped

- Arithmetic encoding is applied on the refined significant pixels

4. ANALYSIS OF PARAMETERS

Various parameters are analysed in this section by applying the LBTC- DWT algorithm and LBTC- DWT with Arithmetic Encoding. The two codecs are compared in the terms of Peak signal to Noise Ratio (PSNR) and the encoding and decoding time.

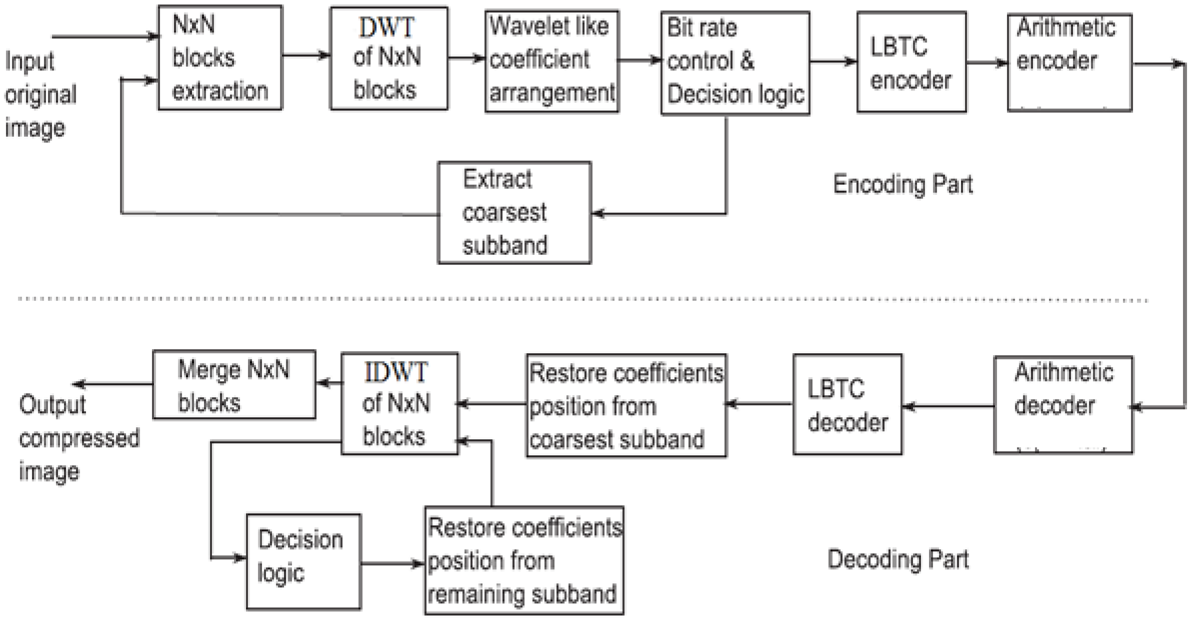

Fig 2: Block Diagram of Proposed LBTC-DWT with Arithmetic Encoding

4.1 Peak Signal to Noise Ratio

Quality of the compressed image is measured by analyzing various parameters as PSNR and Mean Square Error (MSE).

$$PSNR = 10\ \log_{10}\left(\frac{R^2}{MSE}\right)$$

where R is the maximum fluctuation in the input image data type. For double-precision floating-point data type, R is 1 and for 8-bit unsigned integer data type, R is 255. The PSNR is improved by a wide factor by using the algorithm introduced in this paper as compared to the previous techniques.

Table 1 gives the analysis of various images on the basis of PSNR values. It can be seen from the above table that the algorithm introduced in this paper improves the PSNR of an image as compared to the previous techniques. This method is performed on both gray as well as color images. The performance of algorithm on both gray and color images is almost same. The same can also be performed on other images and characters

Table 4.1.1: Analysis of PSNR Value of Various Images by Applying LBTC-DWT with and Without Arithmetic Encoding

	PSNR (in db)	
Image	LBTC-DWT With Arithmetic Encoding	LBTC-DWT Without Arithmetic Encoding
Lena (Gray)	37.5881	37.1325
Baboon (Gray)	29.7382	29.1239
Kidney Scan (Color)	47.1809	46.7364
Spine Scan (Color)	50.1083	49.7387

4.2 Execution Time

Though the quality of an image is improved by using the LBTC- DWT with Arithmetic Encoding but time taken for the execution of the algorithm is more than the previous techniques. The same is shown in the table 4.2.1 in which the time taken for encoding as well as decoding process is shown for various images. Here also the analysis is performed on both the color and gray images. The time taken for the execution of color images is lesser as compared to the gray images of same quality. Hence the algorithm performs faster for color images. It is also analysed that the encoding process is slow as compared to the decoding process. This result holds for LBTC- DWT algorithm perfomed along with the arithmetic encoding as well as when the LBTC- DWT algorithm is performed without the arithmetic encoding.

Table 2: Analysis of Encoding and Decoding Time of LBTC-DWT With and Without Arithmetic Encoding

IMAGE	Encoding Time(In Sec)		Decoding Time (In Sec)	
	LBTC-DWT With Arithmetic Encoding	LBTC-DWT Without Arithmetic Encoding	LBTC-DWT With Arithmetic Encoding	LBTC-DWT Without Arithmetic Encoding
Lena (Gray)	42.2437	30.4856	35.2356	22.3654
Baboon (Gray)	44.5125	32.6256	36.5624	23.1564
Spine Scan	38.5624	26.3564	30.2354	18.5623
Kidney Scan	38.5835	26.4589	30.2314	18.2354

5. CONCLUSION

The algorithm is applied on various images. The results are shown by comparing the purposed algorithm with the previous techniques as LBTC- DWT.

Though the quality of compressed image is improved by some factor but the time taken for the execution of algorithm is more than the time taken by LBTC- DWT.

Figure 3 shows the image of Lena in gray scale compressed by using Enhanced LBTC- DWT algorithm in which the Arithmetic encoding is applied upon the compressed image. Figure 4 shows the Lena image compressed by using the LBTC- DWT algorithm.

It can be observed that the quality of former image is better than the latter one.

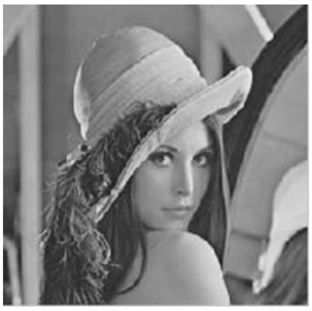

Fig 3: Compressed Lena Image using Enhanced LBTC-DWT Algorithm

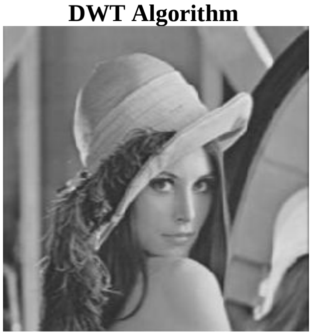

Fig 4: Compressed Lena Image using LBTC-DWT Algorithm

The same observation can be made from the figures 5 and 6 which show the image of a baboon in gray scale. As it can be seen from the figures that the image of baboon compressed by using the Enhanced LBTC- DWT method is better than the one compressed by LBTC- DWT algorithm.

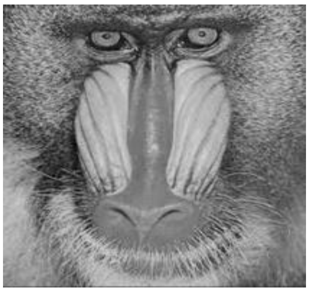

Fig 5: Compressed Baboon Image using Enhanced LBTC-DWT algorithm

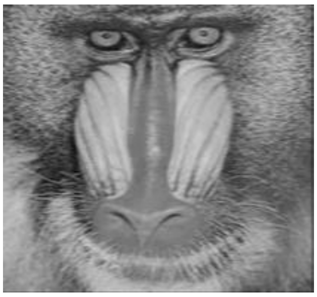

Fig 6: Compressed Baboon Image using LBTC-DWT algorithm

Figures 7 and 8 show image of scan of human kidney in colored scale. It can be concluded from the images that quality of figure 7 is better than the figure 8 where the figures 9 and 10 also draw the same conclusion in which the images of scan of human spine is shown in gray scale.

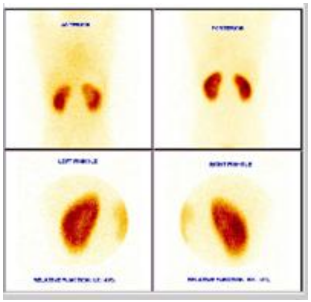

Fig 7: Compressed Image of Human Kidney Scan using Enhanced LBTC-DWT Algorithm

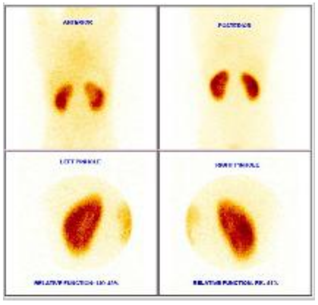

Fig 8: Compressed Image of Human Kidney Scan using LBTC-DWT Algorithm

The image shown in the figure 9 is compressed by using Enhanced LBTC- DWT algorithm which is introduced in this paper while the figure 10 image is compressed by using previous technique i.e. LBTC- DWT in which the compressed image does not go through the Arithmetic Encoding before going to the decoder stage.

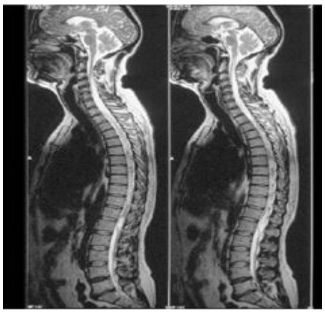

Fig 9: Compressed Image of Human Spine Scan using Enhanced LBTC-DWT Algorithm

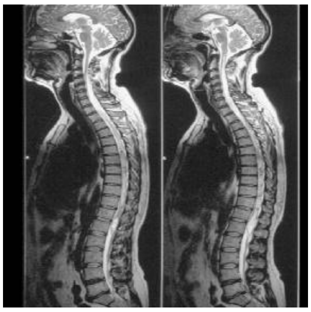

Fig 10: Compressed Image of Human Spine Scan using LBTC-DWT Algorithm

Hence from this paper the conclusion can be drawn that the technique introduced in this paper i.e. Enhanced LBTC- DWT Algorithm which uses Arithmetic Encoding along with LBTC- DWT codec before going to the decoder stage enhances the quality of compressed image as compared to the LBTC- DWT algorithm introduced in the previous papers.

6. REFERENCES

[1] Said A, Pearlman WA., “A New fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees”, IEEE Trans Circuits Syst Video Tech., Jun. 1996;6(3):243–50

[2] Pearlman W.A., Islam A., Nagaraj N., Said A., “Efficient Low Complexity Image Coding with Set-Partitioning Embedded Block Coder”, IEEE Trans. Circuits Syst. Video Technol., (14) pp. 1219–1235, 2004

[3] J.M Shapiro, “Embedded Image Coding Using Zero Trees of Wavelet Coefficients”, IEEE trans on signal processing, (41):3445-3462, Dec. 1993

[4] E.Khan, M.Ghanabari, “Error Detection And Correction Of Transmission Errors In SPIHT Coded Images”, IEEE, ICIP, Jun. 2002:689-692

[5] C.L Tung, T.S Chen, “A New Improvement of SPIHT Progressive Image Transmission”, Proc of IEEE 5th Int. Symposium on Multimedia Software Engineering, (7), Jun. 2003

[6] T.Brahmi, A.Melit, “Improvements to SPIHT for Lossless Image Coding”, IEEE, Jun. 2006

[7] K.Siva Nagi Reddy, V.Sidda Reddy, “Efficient Memory and Low Complexity Image Compression Using DWT with Modified SPIHT Encoder”, International Journal of Scientific & Engineering Research, (3) 8, pp. 2229-5518, Aug. 2012

[8] Y.Sun, H.Zhang, “Real-Time Implementation of a New Low-Memory SPIHT Image Coding Algorithm Using DSP Chip”, IEEE Trans. Image Processing, (11)9, Sept. 2002

[9] H.Minghe, Z.Cuixiang, “ Application Of Improved SPIHT for Multispectral Image Compression”, 5th Int. Conf. On Computer Science & Education, China, Aug. 2010:1058-1061

[10] Y.Jin, H.Lee, “A Block-Based Pass-Parallel SPIHT Algorihtm”, IEEE Tran Circuits And Systems Video Tech.,(22) July 2012

[11] P.Singh, M.N.S.Swamy, “Block Tree Partitioning for Wavelet Based Color Image Compression”, IEEE, ICASSP, Jun. 2006

[12] J.Zhu, S.Lawson, “Improvements to Spiht for Lossy Image Coding”, IEEE, Jan. 2001

[13] S.Zaibi, V.Kerbaol, “Efficient Source and Channel Coding for Progressive Image Transmission over Noisy Channels”, IEEE, Feb. 2002

[14] M.A.Khan, E.Khan, “Error Resilient Technique for SPIHT Coded Color Images”, IEEE, Sept. 2009

[15] Y.Hue, W.A Pearlman, “Progressive Significance Map and Its Application to Error-Resilient Image Transmission”, IEEE Trans. Image Processing, (21) No. 7, July 2012

[16] L.Zhu, Y.Yang, “Embeded Image Compression Using Differential Coding and Optimization Method”, IEEE, 2011

[17] R. K. Senapati, U.C. Pati, “Listless Block-Tree Set Partitioning Algorithm for Very Low Bit Rate Embedded Image Compression”, International Journal of Electronics and Communications (AEU), 2012

[18] C.Kaur, S.Budhiraja, “Listless Block Tree Coding with Discrete Wavelet Transform for Embedded Image Compression at Low Bit Rate” International Journal of Computer Applications, May. 2013

[19] F.W.Whheler, W.A,Pearlman, “SPIHT Image Compression Without Lists”, in Proc IEEE, ICASSP, (4), pp. 2047-2050, Jun. 2000

[20] W.K.Lin, N.Burgess, “Listless Zero Tree Coding for Color Images”, in Proc of 32nd Asilomar Conf on Signals, Sys. And Computers,(1), pp. 231-235, Nov. 1998

[21] J.W.Han, M.C.Hwang, “Vector quantizer based block truncation coding for color image compression in LCD overdrive,” IEEE Trans. Consumer Electron., (54)4, Nov. 2008, pp.1839–1845

[22] R.Praba1, C.Vasanthanayaki, “Enhanced Wavelet Block Tree Based Image Coding Algorithm”, Int. Conf. on Control, Automation,Communication And Energy Conservation, Jun. 2009

[23] Pearlman W.A., Islam A., Nagaraj N., Said A., “Efficient low complexity image coding with set-partitioning embedded block coder’, IEEE Trans. Circuits Syst. Video Technol., 2004(14) pp. 1219–1235

[24] Munteanu A., Cornelis J, “Wavelet Image Compression – The Quadtree Coding Approach. IEEE Trans. on Information Technology.in Biomedicine,1999 (3), :176–18

[25] Deepali Ladhi, Richa Khandelwal, “Implementation of Progressive Block Coder for Image Compression System Using Quad-Tree Partitioning Approach”, International Journal of Information Technology Convergence and Services (IJITCS) (2)1, Feb. 2012

[26] C.D. Creusere, “A New Method of Robust Image Compression Based on Embedded Zerotree Wavelet Algorithm”, IEEE Trans. on Image Proc., (6)10, pp. 1436-42, Oct. 1997

[27] W. A. Pearlman and A. Said, “Image Wavelet Coding Systems: Part II of Set Partition Coding and Image Wavelet Coding Systems,” Found. Trends Signal Process, (2)3, pp. 181–246, 2008

[28] J. Jyotheswar, S. Mahapatra, “Efficient FPGA Implementation of DWT and Modified SPIHT for Lossless Image Compression”, Journal of Sys. Arch., (53), pp.369-378, 2007.