ijiTECH International Journal of Innovative Technologies

WWW.IJITECH.ORG

ISSN 2321-8665
Vol.03,Issue.06,
August-2015,
Pages:1007-1013

Color Image Compression using Set Partitioning in Hierarchical Trees Algorithm

G. RAMESH[1], V.S.R.K SHARMA[2]

[1]PG Scholar, Dept of ECE, Krishna Murthy Institute of Technology & Engineering, JNTUH, India,
E-mail: rameshnaik436@gmail.com

[2]Associate Professor, Dept of ECE, Krishna Murthy Institute of Technology & Engineering, JNTUH, India,
E-mail: vsrksiva@gmail.com.

Abstract: A wavelet based image data codec based on the Set Partitioning In Hierarchical Trees (SPIHT) compression algorithm is proposed in this paper. The SPIHT algorithm has achieved notable success in still image coding. Sequences of images tend to demand enormous memory and storage capacities. For example, consider the Visible Human datasets from the National Library of Medicine. The cry sectional images of the Visible Female consist of images taken at one-third mm intervals resulting in a dataset of about 40 GB Access and transport of these data sets will stress existing processing, storage and transmission capabilities. Therefore, efficient compression should be applied to these data sets before storage and transmission. Many promising image compression algorithm based on wavelet transform (WT) was proposed recently. There are simple, efficient and have been widely used in many applications. Examples include the EZW (Embedded Zero tree Wavelet) algorithm, SPIHT (Set Partition In Hierarchical Trees) algorithm, the improved EZW algorithm of Said and Pearlman, and the SPECK (Set Partition Embedded Block) algorithm by, which offers comparable results to SPIHT with lower complexity. SPIHT as the state-of-the-art encoder has many attractive properties. It is an efficient embedded technique. SPIHT has been extended to 3D by Kim and Pearlman and it has been proved as a powerful tool to compress image sequences. 3D-SPIHT is the modern-day benchmark for three dimensional image compressions. Matlab was able to create adequate testing pictures for this paper.

Keywords: SPIHT algorithm, SPECK (Set Partition Embedded Block) algorithm, EZW (Embedded Zero tree Wavelet) algorithm.

I. INTRODUCTION

The development in technology and networking Discrete Wavelet Transform (DWT) provides a multi resolution image representation and has become one of the most important tools in image analysis and coding over the last two decades. Image compression algorithms based on DWT provide high coding efficiency for natural (smooth) images. As dyadic DWT does not adapt to the various space-frequency properties of Image as, the energy compaction it achieves is generally not optimal. However, the performance can be improved by selecting the transform basis adaptively to the image. Wavelet Packets (WP) represent a generalization of wavelet decomposition scheme. WP image decomposition adaptively selects a transform basis that will be best suited to the particular image. To achieve that, the criterion for best basis selection is needed. Coif man and Wickerhauser proposed entropy based algorithm for best basis selection in their work, the best basis is a basis that describes the particular image with the smallest number of basis functions.

It is a one-sided metric, which is therefore not optimal in a joint rate-distortion sense. A more practical metric considers the number of bits (rate) needed to approximate an image with a given error (distortion) but this approach and its variation presented can be computationally too intensive. In a fast numerical Implementation of the best wavelet packet algorithm is provided. Coding results show that fast wavelet packet coder can significantly out perform a sophisticated wavelet coder constrained to using only a dyadic decomposition, with a negligible increase in computational load. The goal of this paper is to demonstrate advantages and disadvantages of using WP decomposition in SPIHT-based codec. SPIHT algorithm was introduced by Said and Pearlman, and is improved and extended version of Embedded Zero tree Wavelet (EZW) coding algorithm Both algorithms work with tree structure, called Spatial Orientation Tree (SOT), that defines the spatial relationships among wavelet coefficients in different decomposition sub bands. In this way, an efficient prediction of significance of co efficient based on significance of their ‘‘parent’’ coefficients is enabled.

II. IMAGE CODEC BASED ON SPIH

Set Partitioning in Hierarchical Trees (SPIHT) is an image compression algorithm that exploits the inherent similarities across sub bands in a wavelet decomposition of an image. It implies uniform quantization and bit allocation applied after wavelet decomposition. General description The algorithm codes the most important (in the sense of MSE reduction) wavelet transform coefficients in priority, and transmits the bits so that an increasingly refined The order in which coefficients are transmitted is recovered on the decoder using information of comparisons and sets being examined for

significance during the sort, sets are created using hierarchical tree structure, i.e. Set Partition in Hierarchical Trees. Set Partitioning Algorithm The SPHT algorithm is unique in that it does not directly transmit the contents of the sets, the pixel values, or the pixel coordinates. What it does transmit is the decisions made in each step of the progression of the trees that define the structure of the image. Because only decisions are being transmitted, the pixel value is defined by what points the decisions are made and their outcomes, while the coordinates of the pixels are defined by which tree and what part of that tree the decision is being made on. The advantage to this is that the decoder can have an identical algorithm to be able to identify with each of the decisions and create identical sets along with the encoder. The part of the SPHT that designates the pixel values is the comparison of each pixel value to $2n \leq | ci,j | < 2n+1$ with each pass of the algorithm having a decreasing value of n.
In this way, the decoding algorithm will not need be passed the pixel values of the sets but can get that bit value from a single value of n per bit depth level. This is also the way in which the magnitude of the compression can be controlled. By having an adequate number for n, there will be many loops of information being passed but the error will be small, and likewise if n is small, the more variation in pixel value will be tolerated for a given final pixel value. A pixel value that is $2n \leq | ci,j |$ is said to be significant for that pass.

By sorting through the pixel values, certain coordinates can be tagged at "significant" or "insignificant" and then set into partitions of sets. The trouble with traversing through all pixel values multiple times to decide on the contents of each set is an idea that is inefficient and would take a large amount of time. Therefore the SPHT algorithm is able to make judgments by simulating a tree sort and by being able to only traverse into the tree as much as needed on each pass. This works exceptionally well because the wavelet transform produces an image with Properties that this algorithm can take advantage of. This "tree" can be defined as having the root at the very upper left most pixel values and extending down into the image with each node having four (2 x 2 pixel group) offspring nodes (see figure 1).

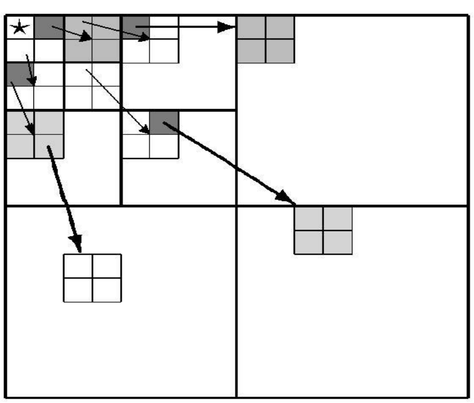

Fig. 1. Sorting the pixel values.

The wavelet transformed image has the desired property that the offspring of a node will have a smaller pixel magnitude value than the parent node. By exploiting this concept, the SPHT algorithm will not have to progress through all the pixels in a given pass if it need not go past a certain node in the tree. This means that the combined lists do not need to contain all the coordinates of every pixel, just those that will show the adequate comparison information. Unfortunately, using tree traversal algorithms would slow down the performance of the system and create unnecessary complex data structures. Instead of tree traversal, the SPHT algorithm uses sets of points to be able to hold the minimal amount of values and still make comparisons to other lists instead of many bulky tree structures. The following sets of coordinates are used to present the new coding method: H**:** set of coordinates of all spatial orientation tree roots (nodes in the highest pyramid level, the lowest resolution); i.e. Band I. D(i,j): set of coordinates of all descendants of node [i,j]; i.e. D(0,1) consists of the coordinates of the coefficients b1,….,b4,b11,…,b14,….,b44. Because the number of offspring's can either be zero or four, the size of D(i,j) is either zero or a sum of powers of four. O(i,j): set of coordinates of all offspring of node (i,j); i.e. O(0,1) consists of the coordinates of the coefficients b1,b2,b3, and b4. L(i,j): This is the set of coordinates to fall the descendants of the coefficients at location (i,j) except for the immediate off springs of the coefficient at location (i,j). In other words L(i,j)=D(i,j)-O(i,j) i.e. The set L(i,j) consists of coordinates of the coefficients b11,…..,b14,….b44.

III. DISCRETE WAVELET TRANSFORMS IMPLEMENTATION

There exist two ways how to implement the computation of the discrete-time wavelet transform. The first approach uses convolution (filtering) with appropriate boundary handling, the second is a fast lifting approach, a refined system of very short filters which are applied in a way that produces the same result as the first approach, introducing significant computational and memory savings [1]. Lifting scheme is derived from a poly phase matrix representation of the wavelet filters, a representation that is distinguishing between even and odd samples. Using the algorithm of filter factoring, we split the original filter into a series of shorter filters (typically Laurent polynomials of first degree). Those filters are designed as lifting steps; each step one group of coefficients are lifted (altered) with the help of the other one (classical dyadic transform always leads to two groups of coefficients, low-pass and high-pass). Data flow in the final algorithm is presented in below Figure. Lifting scheme has one disadvantage – we have to factorize the filters before we can start the computation. In the case of the most widely used image processing wavelet filters – the Cohen-Daubechies-Feauveau 9/7-tap filters (CDF 9/7) – it is an easy task since the most efficient scheme has already been proposed and used (for example in the JPEG 2000 codec). Computational savings of the scheme are gained from the length of the filters (convolution with a 9-tap filter is slower than with a series of two-tap filters) and due to minimum dependency between the coefficients, the whole computation can be done in one memory block of original signal size. We also resolve the problem of boundary effect handling very easily, as we only have to copy one missing sample on each side, using half-sample symmetry.

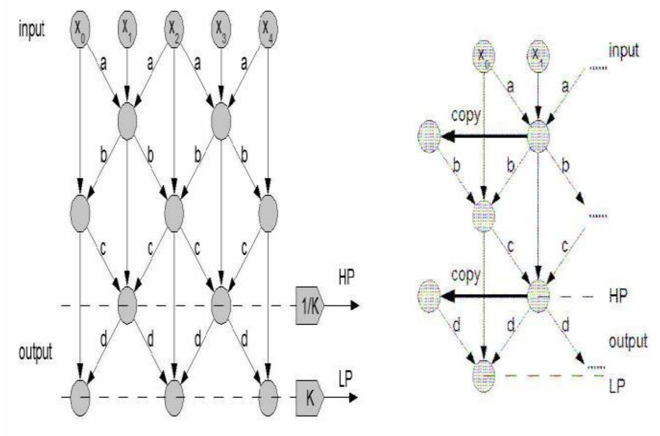

Fig.2. Data dependency diagram showing lifting scheme Implementation.

Lifting scheme is derived from a poly phase matrix representation of the wavelet filters, a representation that is distinguishing between even and odd samples. Using the algorithm of filter factoring, we split the original filter into a series of shorter filters (typically Laurent polynomials of first degree). Those filters are designed as lifting steps; each step one group of coefficients are lifted (altered) with the help of the other one (classical dyadic transform always leads to two groups of coefficients, low-pass and high-pass). Data flow in the final algorithm is presented in Fig. 3. Data dependency diagram showing the data flow according to CDF9/7 lifting scheme implementation. Source signal is defined as x, the result is taken from HP and LP branches for high-pass and low-pass coefficients, respectively; a,b,c,d and K are constants computed by filter factorization process. Picture on the right is demonstrating boundary handling. Since images are two-dimensional signals, we have to extend the scheme to 2D space by applying the transform row- and column-wise, respectively (taking separability of the transform into account). As a consequence four sub bands arise from one level of the transform – one low-pass sub band containing the coarse approximation of the source image called LL sub band, and three high pass sub bands that exploit image details across different directions – HL for horizontal, LH for vertical and HH for diagonal details. In the next level of the transform, we use the LL band for further decomposition and replace it with respective four sub bands. This forms the decomposition image.

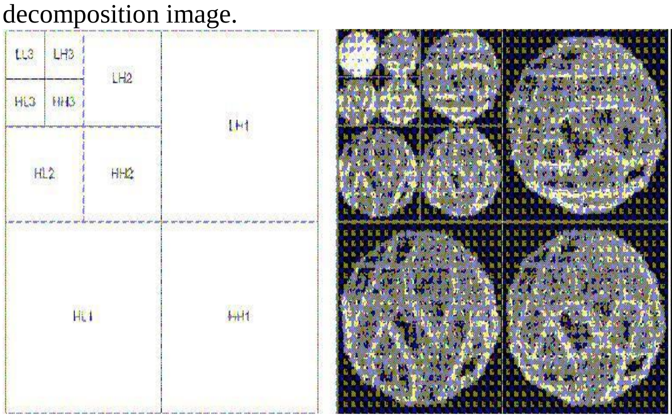

Fig.3. Frequency domain in the form of a decomposition image.

IV. PROPOSED SYSTEM

We have proposed an implementation of DWT-SPIHT coder and decoder for demonstration purposes. This coder is limited to images of square resolutions of power of 2 and grayscale (256 levels, that is 8 bit per pixel) information only. It produces a reconstructed output in spatial domain for comparison and also automatically computes PSNR (Peak Signal to Noise Ratio) as a measure of the difference between source and compressed (destination) images. SPIHT is very vulnerable to bit corruption, as a single bit error can introduce significant image distortion depending of its location. Much worse property is the need of precise bit synchronization, because a leak in bit transmission can lead to complete misinterpretation from the side of the decoder. Input parameters are: source image, wavelet name, transform depth and requested algorithm efficiency in bits per pixel (bpp). From wavelet name we recognize whether it is a part of MATLAB wavelet toolbox and use the appropriate wavelet toolbox functions [5] then, or employ a self-made CDF9/7 lifting scheme implementation (which is widely available). Both transforms produce the same decomposition image of specified depth. SPIHT coder is applied then, processing the exact number of bits computed from the bpp quantifier. Resulting bit stream is taken as a source for the decoder, that produces a reconstructed image. SPIHT is very vulnerable to bit corruption, as a single bit error can introduce significant image distortion depending of its location. Much worse property is the need of precise bit synchronization, because a leak in bit transmission can lead to complete misinterpretation from the side of the decoder.

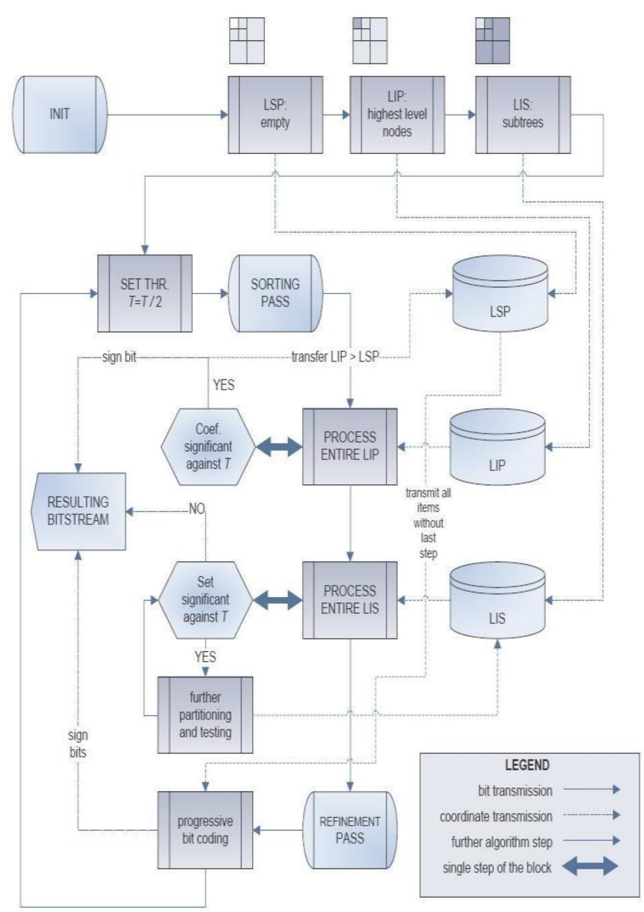

Fig.4. SPIHT algorithm scheme.

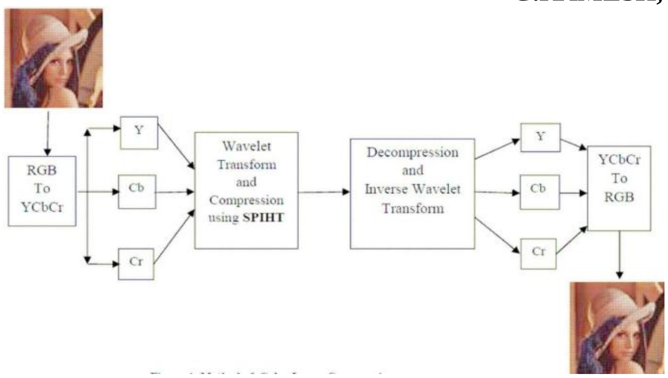

Fig.5. Method of Color Image Compression.

Wavelet decomposition is dyadic in a case when only the LLk sub band is further transformed. It results in a new set of sub bands: HHkþ1, HLkþ1, LHkþ1, LLkþ1. Dyadic decomposition used in image compression will thus generate hierarchical pyramidal structure, as shown in Fig. 6. If the dyadic decomposition of N levels is performed (N times transforming the low–low sub band) the result will be 3N þ 1 sub bands. The WP decomposition is a generalization of wavelet dyadic decomposition, where further wavelet transform on detail sub bands is possible, potentially producing up to 4N final sub bands. A single wavelet packet decomposition thus provides a multitude of choices from which the best representation with respect to a design objective (e.g.compression efficiency) can be found. In order to achieve compression gain while keeping the computational load reasonably low, two entities are needed: a criterion (cost function) for basis comparison and a fast search algorithm, which finds the best basis from the set of all possible bases. Therefore, in this work we use Shannon entropy of wavelet coefficients as cost function for WP bases comparison. The cost is expressed in terms of bits and it provides fast and fairly accurate estimation of the actual output bits that will be spent in coding of the coefficients. In order to find the best basis we apply adaptive search using single spatial tree algorithm.

There are two possible approaches for finding the best tree fully grown tree and ‘‘greedy’’ grown tree. To compare complexity of each one the following notation is introduced here: L _ L is the size of a square image, N is the maximum allowed depth of the decomposition, a0 is the constant specifying per-pixel complexity of DWT for a wavelet filter of a specific length, and a1 is the constant specifying per-pixel complexity of the computation of the specific cost measure. With this notation the complexity of the dyadic wavelet transform f D is

$$f_D = \sum_{k=0}^{N-1} \alpha_0 (2^{-k}L)^2 = \alpha_0 \frac{4^{N-1}-1}{3\times 4^{N-2}} L^2 \leqslant \alpha_0 \frac{4}{3} L^2. \qquad (1)$$

Since the decomposition is fixed, no computation of cost measure is necessary and the complexity is dependent only on a0. In a case where the full growth is employed, the best basis is searched over all possible bases. With respect to the given cost measure, the optimal basis can always be found. If all the intermediate sub bands are preserved. In this case the complexity is independent of the finally selected tree. When the greedy search happens to grow the decomposition tree up to the fully grown tree, then Depending on the source image, the complexity generally falls between the two specified extremes. Thus, greedy approach is in general computationally less complex on the expense of potentially suboptimal performance.

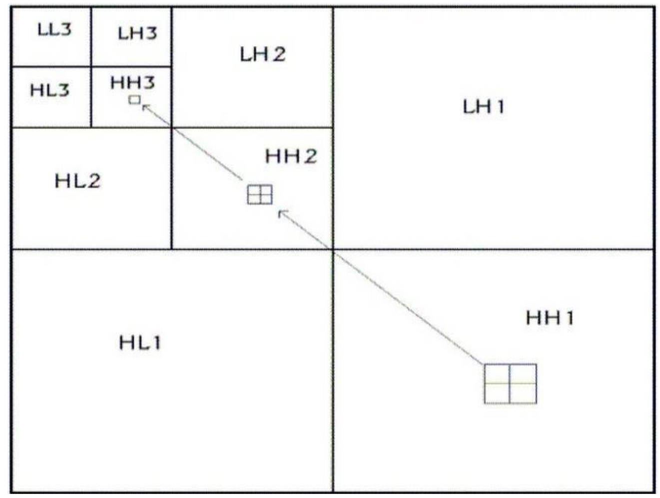

Fig.6.Pyramidal Structure of 3-level wavelet decomposition.

In the proposed implementation, each of the N steps of the dyadic wavelet decomposition is followed by the greedy growing tree algorithm on the obtained high-pass sub bands. The growth of the tree is controlled with parameter dpN, that defines the coarsest scale of the su band that can be produced by the greedy growth. Each sub band that is on the scale finer than d is decomposed depending on the decision given by the cost function. The complexity for this type of decomposition is upper bounded with

$$\begin{aligned} f_{G_d} &= f_D + 3\left(\alpha_0 \sum_{k=1}^{d-1} 2^{-2k}(d-k) \right. \\ &\quad \left. + \alpha_1 \sum_{k=1}^{d-1} 2^{-2k}(d-k+1)\right) L^2 \\ &= f_D + \frac{1}{3}\left(\alpha_0\left(\frac{1}{4^{d-1}} + 3d - 4\right) \right. \\ &\quad \left. + \alpha_1\left(\frac{-2}{4^{d-1}} + 3d - 1\right)\right) L^2 \\ &\leqslant \left(\alpha_0\left(d + \frac{4}{3}\right) + \alpha_1 d\right) L^2. \end{aligned} \qquad (2)$$

It can be seen that the complexity increases linearly with d. The lower bound of the complexity, for the case when only the immediately lower level of the dyadic decomposition tree is grown, is given by Since the algorithm execution time largely depends on the wavelet transform, this determines the overall WP SPIHT execution time to be at least two times longer than for the baseline SPIHT. Pseudocodes for WP decomposition/reconstruction and greedy tree growth algorithm are given in Tables 1–2. Dyadic wavelet decomposition interleaved with greedy growth decomposition of the high-pass sub bands is defined by function ‘‘Decompose’’ (Table 1), where the greedy growth itself is performed with the function ‘‘WPanalysis’’ (Table 2).

$$f_{G_i'} = f_D + 3(\alpha_0 + 2\alpha_1)\sum_{k=1}^{d-1} 2^{-2k} L^2$$
$$= f_D + \frac{4^{d-1}-1}{4^{d-1}}(\alpha_0 + 2\alpha_1)L^2 \approx 2f_D. \qquad (3)$$

The ‘‘treeinfo’’ vector contains a description of the chosen basis and is necessary at the decoder for proper reconstruction. This information represents a bit-stream overhead, but its influence on compression results is negligible. Note that when d ¼ this is equivalent to performing only dyadic decomposition and in this case the WP-SPIHT coder produces results identical to the baseline SPIHT coder. The image is reconstructed as shown in Table 2, using function. In the proposed implementation, each of the N steps of the dyadic wavelet decomposition is followed by the greedy growing tree algorithm on the obtained high-pass sub bands. The growth of the tree is controlled with parameter dpN, that defines the coarsest scale of the sub band that can be produced by the greedy growth.

Table.1. Pseudo code for wavelet packets decomposition

function Decompose(image, N, d) $\rightarrow$ {S, $cost_{image}$, treeinf}

- image $\rightarrow LL_0$
- $0 \rightarrow cost_{image}$
- [ ] $\rightarrow$ treeinf
- for $k = 1$ to N
    - $DWT(LL_{k-1}) \rightarrow [[LL_k\ HL_k]^T[LH_k\ HH_k]^T] \rightarrow LL_{k-1}$
    - if $k<d$
        - WPanalysis($HL_k, d-k$, treeinf) $\rightarrow$ {$HL_k, cost_{HL}$, treeinf}
        - WPanalysis($LH_k, d-k$, treeinf) $\rightarrow$ {$LH_k, cost_{LH}$, treeinf}
        - WPanalysis($HH_k, d-k$, treeinf) $\rightarrow$ {$HH_k, cost_{HH}$, treeinf}
    - else
        - $cost(HL_k) \rightarrow cost_{HL_k}$
        - $cost(LH_k) \rightarrow cost_{LH_k}$
        - $cost(HH_k) \rightarrow cost_{HH_k}$
    - $cost_{image} + cost_{HL_k} + cost_{LH_k} + cost_{HH_k} \rightarrow cost_{image}$
    - define parent–children relations
    - resolve parental conflicts
- $cost_{image} + cost(LL_N) \rightarrow cost_{image}$
- $LL_0 \rightarrow S \equiv \bigcup_{k=1}^{N}(HL_k, LH_k, HH_k), LL_N$

Table.2. Pseudo code for greedy tree growth algorithm

function WPanalysis(S, $depth$, treeinf) $\rightarrow$ {S, $cost_S$, treeinf}

- $DWT(S) \rightarrow [[LL\ HL]^T[LH\ HH]^T] \rightarrow S$
- $cost(S) \rightarrow cost_S$
- $cost(LL) \rightarrow cost_{LL}$
- $cost(HL) \rightarrow cost_{HL}$
- $cost(LH) \rightarrow cost_{LH}$
- $cost(HH) \rightarrow cost_{HH}$
- if $cost_{LL} + cost_{HL} + cost_{LH} + cost_{HH} < cost_S$
    - if $depth > 1$
        - WPanalysis(LL, $depth - 1$, treeinf) $\rightarrow$ {LL, $cost_{LL}$, treeinf}
        - WPanalysis(HL, $depth - 1$, treeinf) $\rightarrow$ {HL, $cost_{HL}$, treeinf}
        - WPanalysis(LH, $depth - 1$, treeinf) $\rightarrow$ {LH, $cost_{LH}$, treeinf}
        - WPanalysis(HH, $depth - 1$, treeinf) $\rightarrow$ {HH, $cost_{HH}$, treeinf}
    - [1 treeinf] $\rightarrow$ treeinf
    - $cost_{LL} + cost_{HL} + cost_{LH} + cost_{HH} \rightarrow cost_S$
    - $[[LL\ HL]^T[LH\ HH]^T] \rightarrow S$
- else [0 treeinf] $\rightarrow$ treeinf

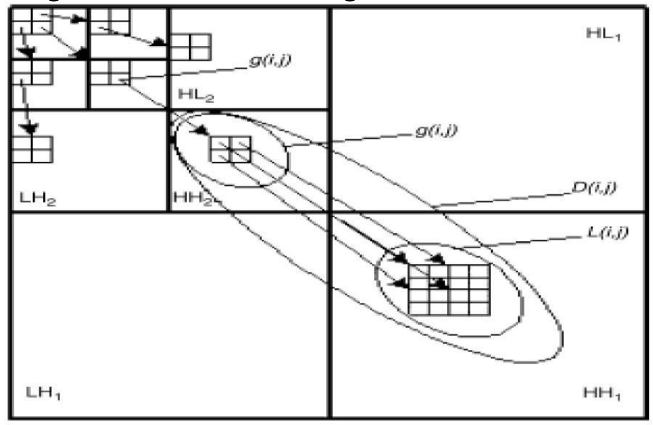

Fig.7. Parent-children relations across sub bands.

V. EXPERIMENTAL RESULTS

WP-SPIHT algorithm has been tested on 256 X 256 all images compressions different levels with below wave lets names “Haar ”, ” Biorthogonal ”, ” Symlets ”, ” Daubechies ”, ReverseBior ”, ” D a Meyer ”, “Coif lets “ We present WP-SPIHT coding results both visually and in the terms of PSNR. Results only one level decomposition in the documentation and we show complete Results with MATLAB 7.2 VERSION.

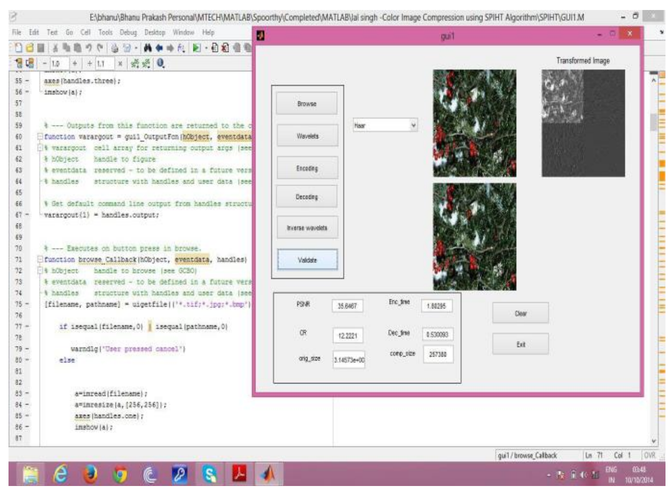

Fig.8. WP-SPIHT coding results both visually and in the terms of PSNR.

Present WP-SPIHT coding results with “haar” wave let image compression level one below figures.

Fig.9. Original image.

Fig.10. Compressed image.

Fig.11. Reconstructed image.

We present WP-SPIHT coding results with " Biorthogonal" wave let image compression level one below figures.

Fig.12. Original image.

Fig.13. Compressed image.

Fig.14. Reconstructed image.

We present WP-SPIHT coding results with "Symlets" wave let image compression level one below figures

Fig.15. Original image.

Fig.16. Compressed image.

Fig.17. Reconstructed image.

VI. CONCLUSION & FUTURE SCOPE

This paper finally concluded and future scope will be enhanced below.

A. Conclusion

Efficient set of rules for establishing zerotree structures when used with WP decomposition is presented. The proposed solution enables the modification of popular SPIHT scheme, called WP-SPIHT—a combination of WP as a decomposition method with SPIHT as an image compression scheme. The compression performance of WP-SPIHT has been compared to SPIHT both visually and in terms of PSN R. WP-SPIHT significantly outperforms SPIHT for textures. For natural images, which consist ofboth smooth and textured areas, the chosen cost function used in WP is not capable ofestimat ing correctly the true cost for a case when the subband is encoded with SPIHT. For those images the PSNR performance of WP-SPIHT is usually slightly worse. As opposed to wavelets, wavelet packet basis might not necessarily produce coefficients that help towards success ofthe zerotree quantization. Using the entropy as a criterion for the best basis selection can result into many coarse scale high frequency subbands. If a coefficient somewhere at the bottom ofthe zerotree is found to be significant, the parent nodes need to be encoded even ifsome ofthem are insignificant. One way to improve those results would be in designing an advanced cost metric that would take into account the characteristics of SPIHT algorithm and provide optimal distortion value for a given bitrate.

B. Future Scope

Compressing color images efficiently are one of the main problems in multimedia applications. So we have tested the efficiency of color image compression using SPIHT algorithm. The SPIHT algorithm is applied for luminance (Y) and chrominance (Cb,Cr) part of RGB to YCbCr transformed image. Reconstructed image is verified using human vision and PSNR. Huffman and arithmetic coding can be added to increase the compression. We can test the channel behavior by sending compressed image between two computer and check the reconstructed image.

VII. REFERENCES

[1] An extensive study of compressed mammography image quality using SPIHT compression was performed at Stanford University. The paper can be downloaded in PostScript or portable document format (pdf) format.

[2] SPIHT can also be efficiently used with DCTs, as shown by Z. Xiong, O. Guleryuz and M. T. Orchard, in their paper "A DCT-based embedded image coder." KITE ECE Page 71

[3]A new application for image compression proposed by Wm. Randolph Franklin. is described in Compressing Elevation Data (HTML version).

[4] The importance of trees in the EZW algorithm was studied by V. Ralph Algazi and Robert R. Estes Jr.. You can read a HTML summary or download a postscript paper.

[5] Antonini M, Barland M, Mathieu P, Daubechies I. Image coding using the wavelet transform. IEEE Transactions on Image Processing 1992;2:205–20.