International Journal of Science Engineering and Advance Technology

Performance Analysis of Wavelet Packet Based SPHIT Algorithm

Kakara Suneel Kumar, A.Pravin, T. Siva Sankar Phani, G.Sudeer Babu

M.TECH Scholar, Dept of ECE, Associative Professor, Dept. of ECE, Assistant Professor, Dept. of ECE
BVC Engineering college, Odalarevu, AP, India

Abstract:

The necessity in image compression continuously grows during the last decade. An image compression algorithm based on wavelet packet transform is introduced. This paper introduces an implementation of wavelet packet image compression which is combined with SPIHT (Set Partitioning in Hierarchical Trees) compression scheme. The image compression includes transform of image, quantization and encoding. This paper describes the new approach to construct the best tree on the basis of Huffman coding for further compression.

In this method the tree are known as zero trees and they are efficiently represented by separating the root from the tree to make compression more. Through Experiments (SPIHT Algorithm)we have shown that the image through the wavelet transforms, the wavelet coefficients are generally small value in high frequency region. A large number of experimental results are shown that this method saves a lot of bits in transmission, further enhanced the compression performance.

Key words - Image compression, Wavelet Packet Transforms, Zero trees.

1. Introduction-

Discrete Wavelet Transform (DWT) provides a multiresolution 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 [1].

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 basic 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 outperform 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. The main contribution of Shapiro's work is zero tree quantization of waveletCoefficients and introduction of special zero tree symbol indicating that all coefficients in a SOT are found to be insignificant with respect to a particular quantization threshold. An embedded zero trees quantize refines each input coefficient sequentially using a bitplane coding scheme, and it stops when the size of the encoded bit stream reaches the target bit-rate. SPIHT coder provides gain in PSNR over EZW due to introduction ofa special symbol that indicates significance of child nodes of a significant parent and separation of child nodes (direct descendants) from second-generation descendants.

To date, there have been numerous variants and extensions to SPIHT algorithm, for example: 3-D SPIHT for video coding, SPIHT for color image coding, and scalable SPIHT for network applications.

Since the SPIHT algorithm relies on Spatial Orientation Trees (SOT) defined on dyadic sub band structure, there are a few problems that arise from their adaptation to WP decomposition. First is the so-called parental conflict that happens when in the wavelet packet tree one or more of the child nodes are at the coarser scale than the parent node. It must be resolved in order that SOT structure with well-defined parent–child relationships for arbitrary wavelet decomposition can be created. Xiong et al. avoided the parental conflict by restricting the choice of the basis. In their work the Space-Frequency Quantization (SFQ) algorithm is used. SFQ algorithm employs a rate-distortion (R-D) optimization framework for selecting the best basis and to assign an optimal quantize to each of the wavelet packet sub bands. Wavelet packet geometry and offered a general structure for an arbitrary WP decomposition. In their work a Compatible Zero tree Quantization (CZQ) is utilized,

2. SET PARTITIONING IN HIERARCHICAL TREES (SPIHT) ALGORITHM

Image Codec Based on SPIHT

Set Partitioning in Hierarchical Trees (SPIHT) is an image compression algorithm that exploits the inherent similarities across subbands 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 copy of the original image is obtained with time.

The SPIHT is considered the premier state-of-the-art algorithm in image compression, and has excellent coding performance for 1-D signals. This algorithm has been modified gradually; some of its modified versions: ESPIHT, MSPIHT and etc.

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.

One of the advantages with SPIHT is that it produces an (optimal) embedded bitstream. This means that the bitstream can be truncated at any instant, and is then guaranteed to yield the best possible reconstruction.

Compression Algorithm:

A PostScript version of the paper describing SPIHT is available via anonymous ftp. Here we take the opportunity to comment how it is different from other approaches.

SPIHT represents a small "revolution" in image compression because it broke the trend to more complex (in both the theoretical and the computational senses) compression schemes. While researchers had been trying to improve previous schemes for image coding using very sophisticated vector quantization, SPIHT achieved superior results using the simplest method: uniform scalar quantization. Thus, it is much easier to design fast SPIHT codecs.

Actually, we do expect better compression results from vector quantizers in the future (someday, somewhere, as prophesied by Shannon), but it is unknown if their speed will justify the gains.

SPIHT algorithm

In the SPIHT algorithm, the image is first decomposed into a number of subbands by means of hierarchical wavelet decomposition. The subband coefficients are then grouped into sets known as spatial-orientation trees, which efficiently exploit the correlation between the frequency bands. The coefficients in each spatial orientation tree are then progressively coded from the most significant bit-planes (MSB) to the least significant bit-planes (LSB), starting with the coefficients with the highest magnitude and at the lowest pyramid levels. The SPIHT multistage encoding process employs three lists and sets:

- SPIHT has 3 lists
  - LIP: list of insignificant pixels (individual insignificant coefficients)
  - LIS: list of insignificant lists (insignificant trees)
  - LSP: list of significant pixels (significant coefficients)
- SPIHT defines 2 types of trees

- Type D: check all descendants for significance
- Type L: check all descendants except immediate children

- Other features
  - Root node is checked independently of the rest of the tree
  - SPIHT sorting pass checks significance of LIP & LIS elements, then moves significant coefficients to LSP

SPIHT Basic Algorithm
Step 1: **Initialization**
Compute initial threshold
LIP: all root nodes (in lowpass subband)
LIS: all trees (type D)
LSP: empty
Step 2: Sorting Pass
Check significance of all coefficients in LIP :
If significant, output 1 followed by a sign bit & move it to LSP.
If insignificant, output 0.
Step 3: Check significance of all trees in LIS For type-D tree :
If significant, output 1 & proceed to code its children
If a child is significant, output 1, sign bit, & add it to LSP.
If a child is insignificant, output 0 and add it to the end of LIP.
If the child has descendants, move the tree to the end of LIS as type L, otherwise remove it from LIS.
If insignificant, output 0.
For type-L tree :
If significant, output 1, add each of the children to the end of LIS as type D and remove the parent tree from LIS.
If insignificant, output 0.
Step 4: Refinement pass
For each element in LSP – except those just added above, output the nth most significant bit of coefficient.
End loop over LSP.
Step 5: Decrease the threshold by a factor of 2. Go to Step 2.

3. PROPOSED WAVELET PACKETS TREE

A. Wavelet Packets

The wavelet packet method is a generalization of wavelet decomposition that offers a richer signal analysis. Wavelet packet atoms are waveforms indexed by three naturally interpreted parameters i.e. position, scale (as in wavelet decomposition), and frequency. For a given orthogonal wavelet function, we generate a library of bases called wavelet packet bases. Each of these bases offers a particular way of coding signals, preserving global energy, and reconstructing exact features. The wavelet packets can be used for numerous expansions of a given signal.

Zerotree quantization is an effective way of exploiting the self-similarities among high frequency subbands at various res-olutions. The main thrust of this quantization strategy is in the prediction of the significance of corresponding wavelet coef-ficients in higher frequency sub bands at the finer resolutions by exploiting the parent-offspring relationship. This prediction works well, in terms of efficiently encoding the wavelet coefficients, due to the statistical characteristics of sub bands at various resolutions and is related to the scale-invariance of edges in high frequency sub bands of similar orientation. Moreover, embedded (progressive) transmission and reconstruction, which is required in some applications, using zero trees with successive approximation quantization (SAQ), is quite straightforward.

B. Wavelets to wavelet packets decomposing.

The orthogonal wavelet decomposition procedure splits the approximation coefficients into two parts. After splitting we obtain a vector of approximation coefficients and a vector of detail coefficients both at a coarser scale. The information lost between two successive approximations is captured in the detail coefficients. Then the new approximation coefficient vector is split again. In the wavelet packet approach, each detail coefficient vector also decomposed into two parts as in approximation vector splitting.

C. Building Wavelet Packets

The computation scheme for wavelet packets generation is easy when using an orthogonal wavelet. We start with the two filters of length 2N.

$W_n(x)$, n=0,1,2,3)

$$w_{2n}(x) = \sqrt{2}\sum_{k=0}^{2N-1} h(k).w_n(2-k)$$

$$w_{2n+1}(x) = \sqrt{2}\sum_{k=0}^{2N-1} g(k).w_n(2x-k)$$

h(n) and g(n), corresponding to the

wavelet. $W_0(x) = \Phi(x)$ is the scaling function and $W_1(x) = \Psi(x)$ is the wavelet function. An idea of wavelet packet is the same as wavelet. Only difference is that wavelet packet offers a more complex and flexible analysis. In wavelet packet analysis the details as well as the approximation are split. The wavelet packet tree for 3-level decomposition is shown in Figure 2.

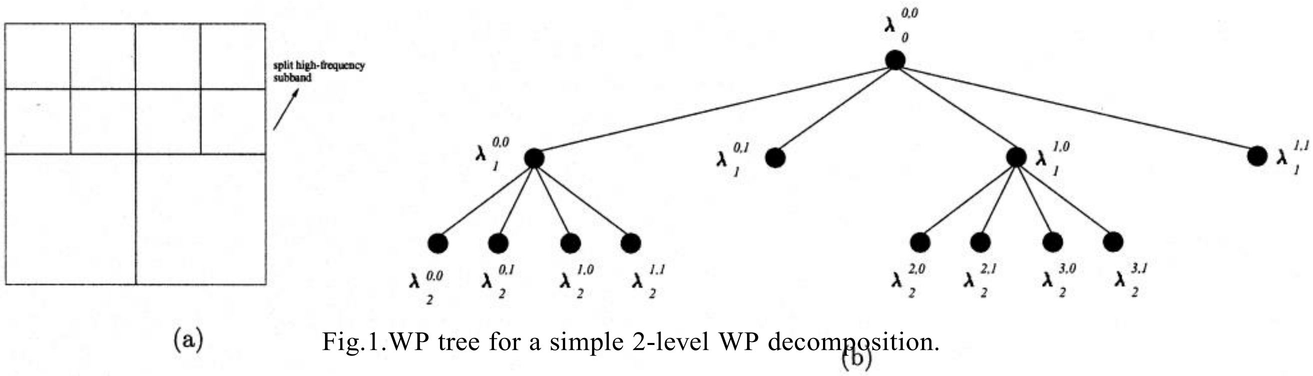

Fig.1.WP tree for a simple 2-level WP decomposition.

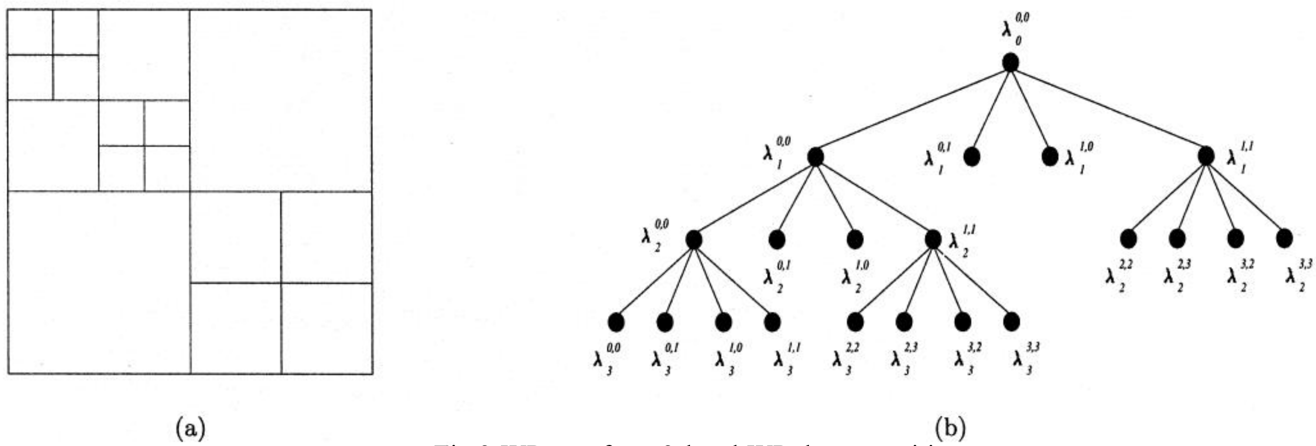

Fig.2.WP tree for a 3-level WP decomposition.

In this paper Shannon entropy criteria is used to construct the best tree. Shannon entropy criteria find the information content of signal 'S'.

$$\text{Entropy}(S) = \sum_i S_i^2 \log S_i^2 \qquad (2)$$

The value of threshold is calculated based on nature of image and type of wavelet used for decomposition.

Threshold =K–Sqrt(mean (w_energy(T)×100))

D. The Proposed Algorithm

The algorithm is described as follows:

1. Level counter = 1
2. The current node = input image.
3. Decompose current node using wavelet packet tree. 4. Find the entropy of the current node.
5. Find the entropy of decomposed components, CA1, CH1, CV1, CD1.
6. Compare the entropy of parent node with the sum of the entropy of child node. If the sum of the entropy of child nodes is less than that of parent node, then child node will be considered as a leaf node of a tree and repeat the steps 3, 4, 5, 6 for each child nodes considering it as current node. Otherwise parent acts as a leaf node of a tree.
7. Stop

5.9. Encoding/Decoding Speed:

The SPIHT process represents a very effective form of entropy-coding. This is shown by the demo programs using two forms of coding: binary-uncoded (extremely simple) and context-based adaptive arithmetic coded (sophisticated). Surprisingly, the difference in compression is small, showing that it is not necessary to use slow methods (and also pay royalties for them!). A fast version using Huffman codes was also successfully tested, but it is not publicly available.

A straightforward consequence of the compression simplicity is the greater coding/decoding speed. The SPIHT algorithm is nearly symmetric, i.e., the time to encode is nearly equal to the time to decode. (Complex compression algorithms tend to have encoding times much larger than the decoding times.)

Some programs use floating-point operations extensively, and can be slower in some CPUs (floating points are better when people want to test you programs with strange 16 bpp images). However, this problem can be easily solved: try the lossless version to see an example. Similarly, the use for progressive transmission requires a somewhat more complex and slower algorithm.

When measuring speed please remember that the demo programs were written for academic studies only, and were not fully optimized.

Experimental Result

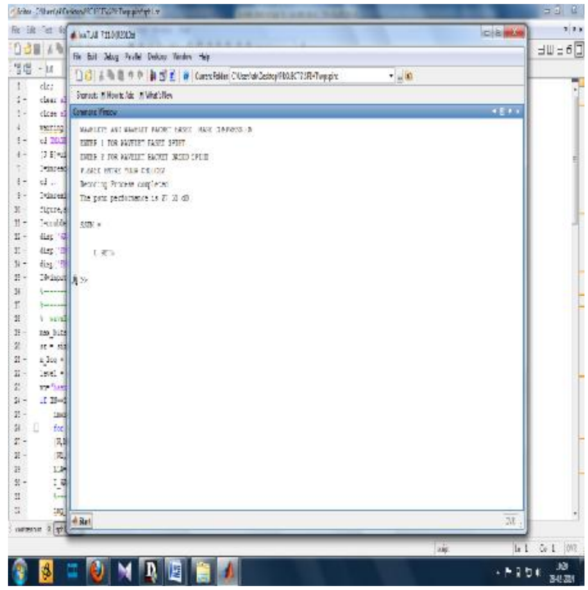

Fig-3 PSNR performance

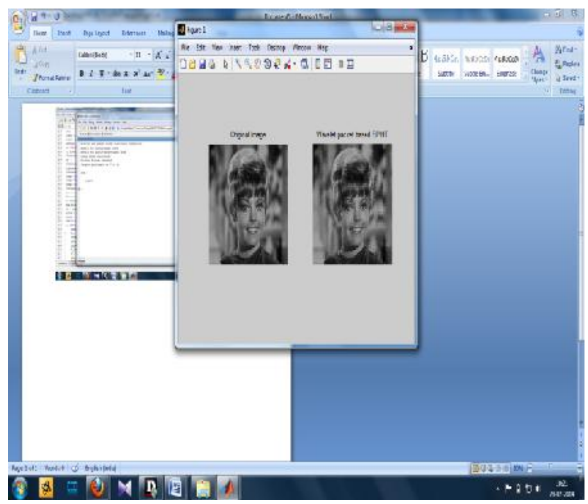

Fig-4 , result incorporated by wavelet method

Conclusion

This work Proposing a simple and effective method combined with Huffman encoding for further compression in this paper that saves a lot log bits in the image data transmission. There is very wide range of practical value for today that have a large number of image data to be transmitted.

References

[1] Rafael C. GONZALEZ Richard E. WOODS. Digital image processing: second ed [M]. Beijing Publishing House of Electronics Industry 2002。

[2] Marc ANTONINI Michel BARLAUD Pierre MATHIEU et al. Image coding using wavelet transform [J]. IEEE Trans. Image Processing 1992 **1**(2) 205-220。

[3] Cheng Li-chi, Wang Hong-xia, Luo Yong. Wavelet theory and applications. Beijing: Science Press,2004(chinese)

[4] J. M. SHAPIRO. Embedded image coding using zerotree of wavelets coefficients [J]. IEEE Trans. Signal Processing 1993 41(12) 3445-346 2。

[5] Amir SAID William A.PEARLMAN . A new fast and efficient image codec based on set partitioning in hierarchical trees [J]. IEEE Transactions On Circuits and Systems for Video Technology 1996 6(3) 243-250。

[6] FAN Qi-bin. Wavelet analysis. Wuhan: Wuhan University Press, 2008.

[7] Sian-Jheng Lin and Wei-Ho Chung "Effective compression using SPIHT with Entropy encoder" IEEE TRANSACTIONS ON IMAGE PROCESSING, VOL. 21, NO. 1, JANUARY 2012