Performance evaluation of wavelet filters for Image Compression using SPIHT

S. M. Patil and S. N. Talbar

Abstract--Embedded Zero tree Wavelet (EZW) coding is a very effective and computationally simple technique for image compression. This principle is based on partial ordering by magnitude with a set partitioning sorting algorithm, ordered bit plane transmission, and exploitation of self-similarity across different scales of an image wavelet transform. Moreover, a new and different implementation based on set partitioning in hierarchical trees (SPIHT) is presented, which provides even better performance than the original EZW. The image coding results, calculated from actual file sizes and images reconstructed by the decoding algorithm, are either comparable to or surpass previous results obtained through much more sophisticated and computationally complex methods. The results for various wavelets on three different images are compared.

Index Terms-- Compression, EZW, SPHIT and Wavelet.

I. INTRODUCTION

THE term data compression refers to the process of reducing the amount of data required to represent a given quantity of information. In digital image compression, three basic data redundancies can be identified and exploited: coding redundancy, interpixel redundancy, and psycho visual redundancy. Data compression is achieved when one or more of these redundancies are reduced or eliminated.

Over the past several years, the wavelet transform has gained widespread acceptance in signal processing in general, and in image compression research in particular. In many applications wavelet-based schemes (also referred as subband coding) outperform other coding schemes like the one based on DCT.

Since there is no need to block the input image and its basis functions have variable length, wavelet coding schemes at higher compression avoid blocking artifacts.

S. M. Patil is with Department of Electronics Engg. , Datta Meghe College of Engg., Airoli

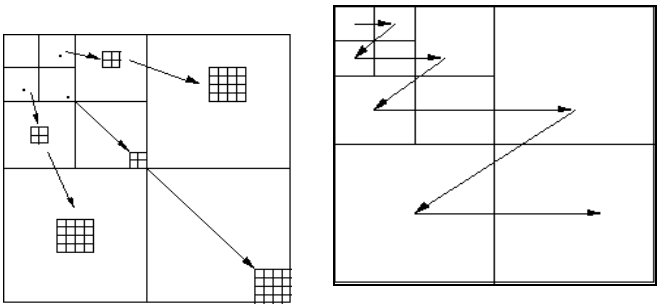
Dr. S. N. Talbar is with Department of EXTC, Shri Guru Govind Singh college of Engg., Nanded

Wavelet-based coding is more robust under transmission and decoding errors, and also facilitates progressive transmission of images. In addition, they are better matched to the HVS characteristics and are suitable for applications where *scalability* and *tolerable degradation* are important.

A. Embedded Zerotree Wavelet (EZW) Compression

In octave-band wavelet decomposition, shown in Fig. 11 (a), each coefficient in the high-pass bands of the wavelet transform has four coefficients corresponding to its spatial position in the octave band above in frequency. Because of this very structure of the decomposition, it probably needed a smarter way of encoding its coefficients to achieve better compression results. A tree-like data structure to represent the coefficients of the octave decomposition was first introduced in 1992.

In1993, Shapiro [1] called this structure zerotree of wavelet coefficients, and presented his elegant algorithm for entropy encoding called Embedded Zerotree Wavelet (EZW) algorithm. The zerotree is based on the hypothesis that if a wavelet coefficient at a coarse scale is insignificant with respect to a given threshold T, then all wavelet coefficients of the same orientation in the same spatial location at a finer scales are likely to be insignificant with respect to T. The idea is to define a tree of zero symbols which starts at a root which is also zero and labeled as end-of-block. Fig.1.1 (a) and 1.1(b) shows a similar zerotree structure. Many insignificant coefficients at higher frequency subbands (finer resolutions) can be discarded, because the tree grows as powers of four. The EZW algorithm encodes the tree structure so obtained. This results in bits that are generated in order of importance, yielding a fully embedded code. The main advantage of this encoding is that the encoder can terminate the encoding at any point, thereby allowing a target bit rate to be met exactly. Similarly, the decoder can also stop decoding at any point resulting in the image that would have been produced at the rate of the truncated bit stream. The algorithm produces excellent results without any pre-stored tables or codebooks, training, or prior knowledge of the image source.



(a) (b)

Fig. 1.1(a) Structure of zerotrees, and (b) Scanning order of subbands for encoding

Many enhancements have been made to make the EZW algorithm more robust and efficient. One very popular and improved variation of the EZW is the SPIHT algorithm.

B. Set Partitioning in Hierarchical Trees (SPIHT) Algorithm

Said and Pearlman [9], offered an alternative explanation of the principles of operation of the EZW algorithm to better understand the reasons for its excellent performance. According to them, partial ordering by magnitude of the transformed coefficients with a set partitioning sorting algorithm, ordered bitplane transmission of refinement bits, and exploitation of self-similarity of the image wavelet transform across different scales of an image are the three key concepts in EZW. In addition, they offer a new and more effective implementation of the modified EZW algorithm based on set partitioning in hierarchical trees, and call it the SPIHT algorithm. They also presented a scheme for progressive transmission of the coefficient values that incorporates the concepts of ordering the coefficients by magnitude and transmitting the most significant bits first. They use a uniform scalar quantizer and claim that the ordering information made this simple quantization method more efficient than expected. An efficient way to code the ordering information is also proposed. According to them, results from the SPIHT coding algorithm in most cases surpass those obtained from EZQ algorithm.

II. WAVELET TRANSFORM (WT)

A. Basics of Wavelet Transform

The wavelet transform circumvents some of the disadvantages of the Short Time Fourier Transform (STFT). By using wavelet transform it is possible to achieve windows of variable sizes. A better time resolution is obtain with smaller window and a better frequency resolution is obtained with larger window. The time resolution directly varies with window size. In case of STFT frequency resolution varies linearly with the reciprocal of window size. However, the wavelet transform is not Fourier transform and consequently the relationship between window size and frequency is not simple.

For example

$$s(t) = \sin 2\pi f_0 t + \sin 2\pi (f_0 + \Delta f)t \tag{2.1}$$

For given frequency separation Δf the wavelet transform has a better chance of resolving the two component if f_o is smaller rather than large. That is rather than Δf alone, it is $\Delta f / f_0$ that matters for resolution of frequency. This is called, "constant Q" or "constant relative bandwidth" property. Generally real world signal encodes meaningful information in frequency band whose bandwidth is proportional to its absolute centre frequency, which is constant Q distribution energy. Thus, WT is particularly well suited for detecting meaningful frequency variation in such real world signals.

B. Continuous wavelet transform (CWT)

Like STFT the WT can be expressed as the time integrated product of a signal *s(t)* with a set of analyzing basis functions. However, the basis functions for the WT are dilated and shifted version of the same "mother wavelet". The mother wavelet can have many different forms, subject to certain mathematical constraints described below. This permits the WT basis function to take on a greater variety of shapes where as the basis functions of the STFT which are restricted to windowed sinusoidal oscillations.

The CWT is given by

$$W(s,\tau) = \int s(t)\psi^*_{s,\tau}(t)dt \tag{2.2}$$

$$\psi(s,\tau) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right) \tag{2.3}$$

Where $\Psi(s,\tau)$ is mother wavelet with most of the energy in concentrated in localized region. The WT $W(s,\tau)$ is closely related to time – frequency representations. The parameter τ is identical to τ in STFT. The parameter *S* is called the dilation or scale parameter and its reciprocal is similar to frequency.

Certain restriction on the mother wavelet is required so that the original signal can be recovered by the inverse WT.

WT removes the DC component of the signal. This DC component can be added back to the signal after performing inverse WT.

The duration of the wavelet defines its effective window width. As the duration of window the scale "s" also varies. The WT uses windows of varying size therefore overcome the key disadvantages of STFT, which uses a fixed window size. In WT variable window size provides different frequency resolutions where as STFT gives fixed frequency resolution. Information in the low frequency end of the signal is captured by basis of large "s" On the other hand for smaller values of "s" the basis function are narrower and their energy spectrum densities peak at higher frequency with broader lobes and so

information of high frequency is capture. Hence the form of time- frequency representation provided by WT is more accurately referred to as a "time - scale" representation.

Of course the multiresolution analysis can be extended to two dimensions. The straightforward way to do this is to use a tensor product of two 1-D multiresolution analyses.

III. SET PARTITIONING IN HIERARCHICAL TREES (SPIHT) ALGORITHM

Image compression techniques, especially nonreversible or lossy ones, have been known to grow computationally more complex as they grow more efficient, confirming the tenets of source coding theorems in information theory that a code for a (stationary) source approaches optimality in the limit of infinite computation (source length). Notwithstanding, the image coding technique called Embedded Zero tree Wavelet (EZW), introduced by Shapiro, interrupted the simultaneous progression of efficiency and complexity. This technique not only was competitive in performance with the most complex techniques, but was extremely fast in execution and produced an embedded bit stream. With an embedded bit stream, the reception of code bits can be stopped at any point and the image can be decompressed and reconstructed. An alternative exposition of the underlying principles of the EZW technique was developed, and presented an extension that achieved even better results.

The EZW technique is based on three concepts:

1) Partial ordering of the transformed image elements by magnitude, with transmission of order by a subset partitioning algorithm that is duplicated at the decoder, 2) ordered bit plane transmission of refinement bits, and 3) exploitation of the self-similarity of the image wavelet transform across different scales. As to be explained, the partial ordering is a result of comparison of transform element (coefficient) magnitudes to a set of octavely decreasing thresholds. An element is significant or insignificant with respect to a given threshold, depending on whether or not it exceeds that threshold.. The subset partitioning is so effective and the significance information so compact that even binary uncoded transmission achieves about the same or better performance than in these previous works. Moreover, the utilization of arithmetic coding increases the peak signal-to-noise ratio (PSNR) by 0.3-0.6 dB for the same rate. Execution times are also reported to indicate the rapid speed of the encoding and decoding algorithms.

A. Set partitioning sorting algorithm

One of the main features of the proposed coding method is that the ordering data is not explicitly transmitted. Instead, it is based on the fact that the execution path of any algorithm is defined by the results of the comparisons on its branching points. So, if the encoder and decoder have the same sorting algorithm, then the decoder can duplicate the encoder's execution path if it receives the results of the magnitude comparisons, and the ordering information can be recovered from the execution path.

One important fact used in the design of the sorting algorithm is that we do not need to sort all coefficients. Actually, we need an algorithm that simply selects the coefficients such that $2n \leq |c_{i,j}| < 2n+1$, with n decremented in each pass.

Given n, if $|c_{i,j}| \geq 2^n$ then we say that a coefficient is ***significant;*** otherwise it is called insignificant The sorting algorithm divides the set of pixels into partitioning subsets τ_m, and performs the magnitude test

$$\max_{(i,j)\in\tau_m}\{|c_{i,j}|\} \geq 2^n \qquad (3.1)$$

If the decoder receives a "no'' to that answer (the subset is insignificant), then it knows that all coefficients in τ_m, are insignificant. If the answer is "yes" (the subset is significant), then a certain rule shared by the encoder and the decoder is used to partition τ_m, into new subsets $\tau_{m,l}$, and the significance test is then applied to the new subsets. This set division process continues until the magnitude test is done to all single coordinate significant subsets in order to identify each significant coefficient. To reduce the number of magnitude comparisons (message bits) we define a set-partitioning rule that uses an expected ordering in the hierarchy defined by the sub band pyramid. The objective is to create new partitions such that subsets expected to be insignificant contain a large number of elements, and subsets expected to be significant contain only one element. To make clear the relationship between magnitude comparisons and message bits, we use the function

$$s_n(\tau) = \begin{cases} 1, \max_{(i,j)\in\tau_m}\{|c_{i,j}|\} \geq 2^n \\ 0 \quad otherwise \end{cases} \qquad (3.2)$$

to indicate the significance of a set of coordinates τ. To simplify the notation of single pixel sets, we write $S_n(\{(i,j)\})$ as $S_n(i,j)$.

B. Spatial orientation trees

Normally, most of images energy is concentrated in the low frequency components. Consequently, the variance decreases as we move from the highest to the lowest levels of the subband pyramid. Furthermore, it has been observed that there is a spatial self-similarity between subbands, and the coefficients are expected to be better magnitude-ordered if we move downward in the pyramid following the same spatial orientation [4]. For instance, large low-activity areas are expected to be identified in the highest levels of the pyramid, and they are replicated in the lower levels at the same spatial locations.

A tree structure, called ***spatial orientation tree,*** naturally defines the spatial relationship on the hierarchical pyramid. Fig.3. 1 shows how our spatial orientation tree is defined in a pyramid constructed with recursive four-sub band splitting. Each node of the tree corresponds to a pixel and is identified by the pixel coordinate. Its direct descendants (offspring) correspond to the pixels of the same spatial orientation in the next finer level of the pyramid. The tree is defined in such a way that each node has either no offspring (the leaves) or four offspring, which always form a group of 2 x 2 adjacent pixels. In Fig.3.1, the arrows are oriented from the parent node to its four offspring. The pixels in the highest level of the pyramid are the tree roots and are also grouped in 2 x 2 adjacent pixels. However, their offspring branching rule is different, and in each group, one of them (indicated by the star in Fig.3.1) has no descendants.

The following sets of coordinates are used to present the new coding method:

$O(i, j)$: Set of coordinates of all off springs of node (i, j);

$D(i, j)$: Set of coordinates of all descendents of the node (i, j);

H : Set of coordinates of all spatial orientation tree roots (Nodes in the highest pyramid level);

$L(i, j) = D(i, j) - O(i, j)$

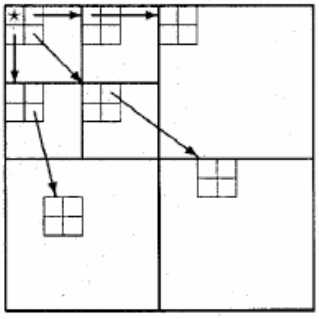


Fig.3. 1 Examples of parent-offspring dependencies in the spatial-orientation tree.

For instance, except at the highest and lowest pyramid levels, we have

$$O(i,j)=\{(2i,2j),(2i,2j+1),(2i+1,2j),(2i+1,2j+1)\} \quad (3.3)$$

We use parts of the spatial orientation trees as the partitioning subsets in the sorting algorithm. The set partitionmg rules are simply the following.

1. The initial partition is formed with the sets $\{(i, j)\}$ and $D(i, j)$ for all $(i, j) \in H$

2. If $D(i, j)$ is significant, then it is partitioned into $L(i, j)$ plus the four single-element sets with $(k,l) \in O(i, j)$

3. If $L(i, j)$ is significant, then it is partitioned into the four sets $D(k,l)$, *with* $(k,l) \in O(i, j)$.

C. Coding algorithm

Since the order in which the subsets are tested for significance is important, in a practical implementation the significance information is stored in three ordered lists, called list of insignificant sets (LIS), list of insignificant pixels (LIP), and list of significant pixels (LSP). In all lists each entry is identified by a coordinate (i, *j),* which in the LIP and LSP represents individual pixels, and in the LIS represents either the set $D(i, j)$ or $L(i, j)$. To differentiate between them, we say that a LIS entry is of type A if it represents $D(i, j)$, and of type *B* if it represents $L(i, j)$.

During the sorting pass [9], the pixels in the LIP-which were insignificant in the previous pass-are tested, and those that become significant are moved to the LSP. Similarly, sets are sequentially evaluated following the LIS order, and when a set is found to be significant it is removed from the list and partitioned. The new subsets with more than one element are added back to the LIS, while the Single-coordinate sets are added to the end of the LIP or the LSP, depending whether they are insignificant or significant, respectively. The LSP contains the coordinates of the pixels that are visited in the refinement pass. Below we present the new encoding algorithm in its entirety. It is essentially equal to Algorithm I [9], but uses the set partitioning approach in its sorting pass.

Algorithm:

1. Initialization:

output $n = \left\lfloor \log_2 \left(\max_{(i,j)} \{| c_{i,j} |\} \right) \right\rfloor$; set the LSP as an empty list, and add the coordinates $(i, j) \in H$ to the LIP, and only those with descendants also to the LIS, as type A entries.

2. Sorting Pass:

2.1) for each entry *(i, j)* in the LIP do:

2.1.1) Output $S_n\ (i, j)$;

2.1.2) If $S_n\ (i, j) = 1$ then move (i, *j)* to the LSP and output the sign of $C_{i,j}$;

2.2): for each entry (i, j) in the LIS do:

2.2.1) if the entry is of type A then

- Output $S_n(D(i,j))$:
- If $S_n(D(i,j)) = 1$ then

  * for each $(k,l) \in O(i,j)$ do:
- Output $S_n(k,l)$;
- If $S_n(k,l) = 1$ then add (k,l) to the LSP and output the sign $C_{k,l}$;
- If $S_n(k,l) = 0$ then add (k,l) to the LIP;

* If $L(i,j) \neq 0$ then move (i,j) to the end of the LIS, as an entry of type B, and go to step 2.2.2); otherwise, remove entry (i,j) from the LIS;

2.2.2) If the entry of type B, then

- Output $S_n(L(i,j))$;
- If $S_n(L(i,j))$ =1 then

  * Add each $(k,l) \in O(i,j)$ to the end of the LIS as an entry of type A;

  * Remove (i,j) from the LIS.

3) **Refinement Pass:** for each entry (i,j) in the LSP, except those included in the last sorting pass (i.e., with same n), output the nth most significant bit of $|c_{i,j}|$;

4) **Quantization-Step Update:** decrement n by 1 and go to Step ***2.***

One important characteristic of this algorithm is that the entries added to the end of the LIS in Step 2.2) are evaluated before that same sorting pass ends. So, when we say "for each ntry in the LIS" we also mean those that are being added to its end. With this Algorithm, the rate can be precisely controlled because the transmitted information is formed of single bits. Note that in this Algorithm, all branching conditions based on the significance data S_n-which can only be calculated with the knowledge of $c_{i,j}$ are output by the encoder. Thus, to obtain the desired decoder's algorithm, which duplicates the encoder's execution path as it sorts the significant coefficients; we simply have to replace the words *output* by *input* in this Algorithm. Comparing the algorithm above to Algorithm I, we can see that the ordering information $\eta(k)$ recovered when the coordinates of the significant coefficients are added to the end of the LSP; that is, the coefficients pointed by the coordinates in the LSP are sorted as in [4]. But note that whenever the decoder inputs data, its three control lists (LIS, LIP, and LSP) are identical to the ones used by the encoder at the moment it outputs that data, which means that the decoder indeed recovers the ordering from the execution path. It is easy to see that with this scheme, coding and decoding have the same computational complexity.

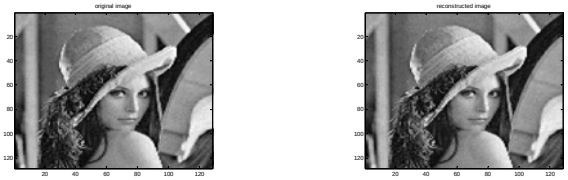
An additional task done by decoder is to update the reconstructed image. For the value of ***n*** when a coordinate is moved to the LSP, it is known that

$$2^n \leq |c_{i,j}| < 2^{n+1}$$

So, the decoder uses that information, plus the sign bit that is input just after the insertion in the LSP, to set $\hat{c}_{i,j} = \pm 1.5 * 2^n$. Similarly, during the refinement pass, the decoder adds or subtracts 2^{n-1} to $\hat{c}_{i,j}$ when it inputs the bits of the binary representation of $|c_{i,j}|$. In this manner, the distortion gradually decreases during both the sorting and refinement passes.

IV. RESULTS AND DISCUSSION

The SPHIT algorithm has been tested on various images. The results are shown for three images of size128 X 128. Figures 4.1, 4.3 and 4.5 shows the original and reconstructed images of Lena, Barbara and Baboon respectively. Figures 4.2,4.4 and 4.6 shows the Bit rate Vs. PSNR (db) curves for Lena, Barbara and Baboon images using haar, bior2.2, bior4.4, db7 and db10 respectively.



(a) (b)

Fig.4.1The original (a) and reconstructed (b) image of Lena

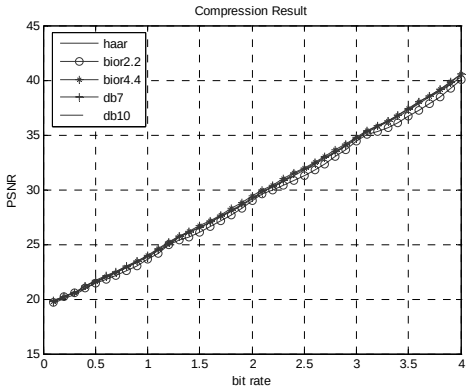
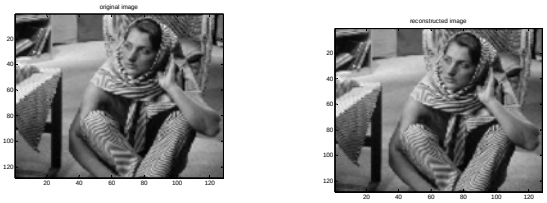


Fig.4.2 Bit rate Vs PSNR (db) for Lena



(a) (b)

Fig.4.3 The original (a) and reconstructed image (b) of Barbara

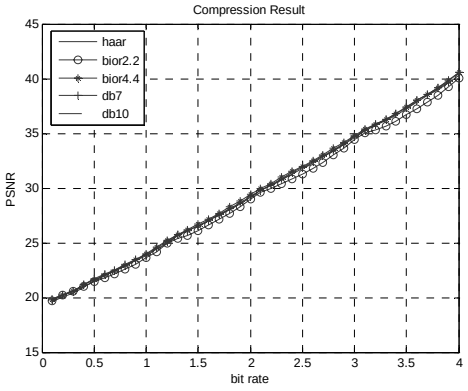
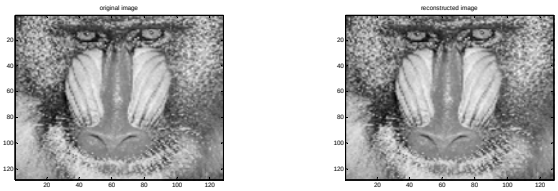


Fig.4.4 Bit rate Vs PSNR (db) for Barbara



(a) (b)

Fig.4.5 The original (a) and reconstructed image (b) of Baboon

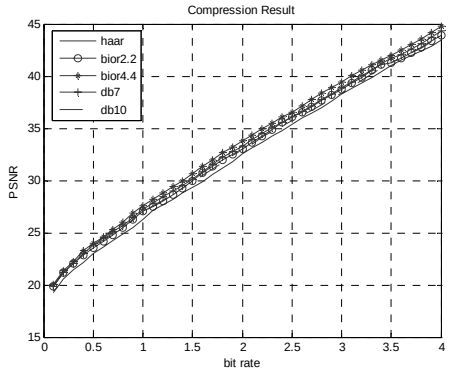


Fig 4.6 Bit rate PSNR vs. bit-rate for Baboon image

TABLE I. COMPARISON OF PSNR VS BIT-RATE FOR BABOON USING VARIOUS FILTERS

Rate	PSNR (db)				
bpp	haar	Bior2.2	Bior4.4	Db7	Db10
3	34.57	34.40	34.79	34.69	34.62
2	29.19	29.01	29.46	29.27	29.20

Above results show that using SPHIT we get better PSNR for low detail images for all bit rates. Comparing results for various filters show that bior4.4 filter gives higher PSNR.

IV. CONCLUSION

Researchers have invested a substantial amount of effort in studying Image compression in the context of data storage, transmission capacity and scalability. This seemingly most important application in terms of communication, therefore, received considerable attention. In spite of this attention, the problem of data compression continues to harbor plenty of challenges, especially for low bit rate requirement.

In this paper we analyzed the SPIHT algorithm for image compression .We have used various types of standard images for illustration purpose. We used 'Lena' a Low detail Image, 'Barbara' a Medium detail Images and 'Baboon' a High detail Image.

We analyzed the performance of SPIHT algorithm for various bit rates and various levels of decompositions.

Results show that, SPIHT gives high PSNR for low detail images for all bit rates. In case of level of wavelet decomposition it can be observed that increase in level of decomposition gives higher PSNR values for all bit rates.

Based on these results it can be stated that SPIHT algorithm is completely based on the statistical contain of the image, which is to be compressed and the order of wavelet approximations.

V. REFERENCES

[1] J. M. Shapiro, "Embedded image coding using zerotrees of wavelets coefficients," IEEE Trans.Signal Processing, vol.41,pp. 3445-3462,Dec. 1993.

[2] M.Rabbani and P. W. Jones, Digital *Image* Compression Techniques. Bellingham, WA: SPIE Opt. Eng. Press, 1991.

[3] R. A. DeVore, B. Jawerth, and B. J. Lucier, "Image compression through wavelet transform coding," IEEE Trans. *I*nform. Theory, vol. 38, pp, 719-746, Mar. 1992.

[4] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. Imam Processing**,** vol. 1, _p_p. 205- 220, Apr. 1992.

[5] Said and W. A. Pearlman, "Image compression using the spatial orientation tree," in IEEE lnt. Symp. Circuits-and Systems,-Chicago, IL, pp. 279-282, May 1993.

[6] H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," Commun. ACM, vol. 30, pp. 520-540, June 1987.

[7] Said and W. A. Pearlman, "Reversible image compression via multiresolution representation and predictive coding," in Proc. SPIE Con$ Visual Communications and Image Processing '93, Cambridge, MA, Proc. SPIE 2094, pp. 664-674, Nov. 1993.

[8] Raghuveer Rao and Ajit Bopardikar, "Wavelet Transform-Introduction to Theory and Applications", Addison Wesley Pearson Edu. Edition, 2000.

[9] Amir Said, and William *A.* Pearlman "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees" IEEE Transactions on circuits and systems for video technology, vol. 6, no. *3,* June 1996.

[10] Rafael C. Gonzalez and Richard E.Woods "Digital image processing" Pearson Education, 2002.

[11] Khalid Sayood " Introduction to Data Compression" Morgan Kaufmann Publishers, 2005

VI. BIOGRAPHIES

Sanjay M. Patil graduated from Shri Sant Gajanan Maharaj College of Eegg. Shegaon in 1991. He completed M.E. (Electronics) from Shri Guru Govind Singh college of Engg. and Tech.Nanded.. He joined as lecturer in Datta Meghe college of Engg. Airoli in 1993, joined as Assistant Professor in 2007. His field of interest is Image Processing.

Dr.Sanjay N Talbar completed his graduation and post graduation in Electronics Engg. in 1985 and 1990 resp. from Shri Guru Govind Singh college of Engg. and Tech.Nanded. He completed Ph.d from Swami Ramanand Terth Marathwada University, Nanded in 2000. Presently he is working as Professor in Electronics Engg. in Shri Guru Govind Singh college of Engg. and Tech.Nanded and has guided about fifty postgraduate students and also students to Ph.d. He has published four papers in journals and thirty-five papers in national and international conference. His field of interest is Image Processing and Embedded systems.