

A High Performance Modified SPIHT for Scalable Image Compression

Bibhuprasad Mohanty

*Assistant professor,
Institute of Technical Education and Research
SoA University, Bhubaneswar*

bmohanty.iit07@gmail.com.

Abhishek Singh

*Assistant professor
Faculty of Science and Technology,
ICFAI University, Tripura*

abhishek.uor@gmail.com.

Dr. Sudipta Mahapatra

*Associate Professor,
Department of E &ECE,
Indian Institute of Technology, Kharagpur, India*

sudipta.mahapatra@gmail.com

Abstract

In this paper, we present a novel extension technique to the Set Partitioning in Hierarchical Trees (SPIHT) based image compression with spatial scalability. The present modification and the preprocessing techniques provide significantly better quality (both subjectively and objectively) reconstruction at the decoder with little additional computational complexity. There are two proposals for this paper. Firstly, we propose a pre-processing scheme, called Zero-Shifting, that brings the spatial values in signed integer range without changing the dynamic ranges, so that the transformed coefficient calculation becomes more consistent. For that reason, we have to modify the initialization step of the SPIHT algorithms. The experiments demonstrate a significant improvement in visual quality and faster encoding and decoding than the original one. Secondly, we incorporate the idea to facilitate resolution scalable decoding (not incorporated in original SPIHT) by rearranging the order of the encoded output bit stream. During the sorting pass of the SPIHT algorithm, we model the transformed coefficient based on the probability of significance, at a fixed threshold of the offspring. Calling it a fixed context model and generating a Huffman code for each context, we achieve comparable compression efficiency to that of arithmetic coder, but with much less computational complexity and processing time. As far as objective quality assessment of the reconstructed image is concerned, we have compared our results with popular Peak Signal to Noise Ratio (PSNR) and with Structural Similarity Index (SSIM). Both these metrics show that our proposed work is an improvement over the original one.

Key Words : Zero Shifting, 2D SPIHT, Lifting Wavelet Transform, Context Modeling, Resolution Scalability, SSIM

1. INTRODUCTION

Image compression algorithms manipulate images to dramatically reduce the storage and bandwidth required while maximizing perceived image quality. Image compression based on discrete wavelet transform (DWT) has emerged as one of the popular tool. The wavelet domain provides a natural setting for many applications which involve real world signals. The presence of blocking artifacts at low bit rate in the entropy coding schemes (e.g., Discrete Cosine Transform) is completely avoided in this wavelet based coding. This transform achieves better energy compaction than the DCT and hence can help in providing better compression for the same quality measurement, like peak-signal-to-noise-ratio (PSNR) [1] and structural similarity (SSIM) [2].

The widespread deployment of diversified and heterogeneous digital networks requires different bandwidth range for various user nodes. This heterogeneity property demands for scalability to optimally serve each user according to the available bandwidth and computing capabilities. The goal of the scalable compression is to enable clients to reconstruct the image from a single compressed bit stream, according to their specific capabilities. A scalable coder generates a bit stream which consists of a set of embedded parts that offer better SNR and/or greater spatial resolution. Different types of decoders with different complexity and various access bandwidths can be accommodated in the case of an entirely scalable bit stream,

Based on DWT, several competitive image compression coders with high coding efficiency have been developed. Among those coders, Shapiro's Embedded Zero tree Wavelet (EZW) compression [3], Said and Pearlman's set partitioning in Hierarchical Trees (SPIHT) [4], and Tubman's Embedded Block Coding with Optimized Truncation (EBCOT) [5] are the benchmark contribution. The 2-D SPIHT coder performs competitively with most of the other coders published in the literature [6]. The encoded and compressed SPIHT bit stream is further compressed by applying adaptive arithmetic encoder which provides an additional PSNR gain of 0.2 to 0.6 dB [4] over the original one. The low margin PSNR improvement shows the fact that SPIHT itself is exploiting the redundancies across the sub band so efficiently that it leaves little room for further compression by the arithmetic coder. However, use of arithmetic coder increases the computational complexity and hence the scheme sacrifices its speed for a meager gain.

Further improvements of SPIHT have been published in [7-12]. In [13], the authors proposed a novel modification in exploiting the correlation of coefficients by lifting based DWT with 1D addressing method instead of traditional 2D method. Also, for the purpose of hardware implementation, they modified the SPIHT algorithm by interchanging the refinement pass and the sorting pass. Although almost all of the state-of-the-art zero tree-based image compression methods are SNR scalable and provide bit streams for progressive (by quality) image compression, they do not explicitly support spatial scalability and do not provide a bitstream which can be adapted easily to a heterogeneous environment. PROGRES [12] brings the concept of resolution scalability as well as faster encoding and decoding into the SPIHT, but it loses desirable feature like embeddedness and SNR scalable decoding.

In this paper, we present a new scheme which provides significant improvement in the quality of the reconstructed image in terms of PSNR and SSIM without using the adaptive arithmetic encoder. The method, called zero shifting, is applied on the spatial domain of the image before it is wavelet transformed [18]. To accommodate the zero shifting schemes without sacrificing the inherent property of the SPIHT codec, we have modified the original algorithm in [4]. Further, we modeled the transformed coefficient according to its significance in a 2x2 adjacent offspring group as a fixed context. The systematic trend in probability distribution of such a context has been utilized for further compression of the image with the help of Huffman coding which is of lower complexity than the arithmetic coding [19]. We also propose an approach to resolution scalability by grouping together the bit streams generated for each resolution at one location. Both the sorting and refinement pass for SPIHT encoder is performed for each resolution and certain header information are generated which indicate the boundary condition for that resolution. The paper is organized as follows: Section 2 reviews the Lifting based wavelet transform and SPIHT coder briefly. The motivations for the proposed schemes are discussed in Section 3. The codec structure is described in Section 4 and we have discussed the results elaborately in Section 5. The conclusions of this paper are reported in Section 6.

2. BRIEF REVIEW

2.1 Lifting Based Wavelet Transform (WT)

Wavelet transforms provide both spatial and frequency-domain information about an image or a sequence of frames [14]. Such multiresolution representations of images/frames are often a more convenient format for image coding. This is particularly important since Human Vision System

(HVS) is less sensitive to high-frequency distortions than low-frequency ones. WT analyzes a signal at different frequency bands with different resolutions by decomposing it into a coarse approximation and detail information. The classical 2D transform is performed by two separate 1D transforms along the rows and the columns of the image data, resulting at each decomposition step in a low pass image or the coarse scale approximation, and three detail images. Because of their inherent multiresolution signal representation, wavelet-based coding schemes have the potential to support both SNR and spatial scalability [15].

The lifting scheme is a fast and efficient method to realize biorthogonal wavelet transforms [16]. It is simple, fast and reversible in computation due to the use of integer transformation. The algorithm has the advantages of using lifting, instead of convolution, which reduces the memory requirement. This influences the lossless compression algorithm profoundly because of its much wider dynamic range and hence JPEG 2000 uses the scheme. Even 3D wavelet decomposition is computed by applying three separate 1D transforms along the video data: rows and columns of the image as two directions and time as the third direction.

2.2 Set Partitioning In Hierarchical Trees (SPIHT)

SPIHT is an embedded coding technique. In embedded coding algorithms, encoding of the same signal at lower bit rate is embedded at the beginning of the bit stream for the target bit rate. Effectively, bits are ordered in importance. This type of coding is especially useful for progressive transmission using an embedded code; where an encoder can terminate the encoding process at any point. SPIHT algorithm is based on following concepts [4]:

1. Ordered bit plane progressive transmission.
2. Set partitioning sorting algorithm.
3. Spatial orientation trees.

SPIHT keeps three lists: LIP, LSP, and LIS. LIP stores insignificant pixels, LSP stores significant pixels and LIS stores insignificant sets. At the beginning, LSP is empty, LIP keeps all coefficients in the lowest sub band, and LIS keeps all tree roots which are at the lowest sub band. SPIHT starts coding by running two passes. The first pass is the sorting pass. It first browses the LIP and moves all significant coefficients to LSP and outputs its sign. Then it browses LIS executing the significance information and following the partitioning sorting algorithms.

The second pass is the refining pass. It browses the coefficients in LSP and outputs a single bit alone based on the current threshold. After the two passes are finished, the threshold is divided by 2 and the encoder executes the two passes again. This procedure is recursively applied until the number of output bits reaches the desired number.

3. THE PROPOSED SCHEME

3.1 The Zero Shifting

The method of zero shifting is a simple and easy-to-implement technique which preserves the embedded property of the SPIHT coding. By downward scaling of the pixel values by $2^{(N-1)}$, N being the number of bits representing the original pixel, the spatial domain magnitude becomes bipolar ranging from $(-2^{(N-1)})$ to $(+2^{(N-1)}-1)$, with almost half of the maximum absolute spatial value. In the transform domain, the wavelet lifting scheme produces the high pass sub band by taking the weighted differences of the pixel values after prediction. Similarly, the low pass sub band is produced by taking the weighted average of the pixel values after the updating step. This low pass sub band is further decomposed iteratively for each level of decomposition to finally provide, one lowest frequency band, the rest being high frequency bands.

SPIHT coding is a method of bit plane coding technique. The highest number bit plane is decided by the maximum absolute value of the transformed coefficients. As the maximum coefficient decreases by the method of zero shifting, SPIHT algorithm takes less time for encoding as well as for decoding. Further, the lower absolute image values in the spatial domain (which originally ranges from $0 \leq p_{i,j} \leq 2^{(N-1)}$) are shifted to higher absolute range in the bipolar sense. The lower

image values are detailed information pertaining to the boundaries, edges, etc. These coefficients become significant at a higher threshold value and accordingly are ordered earlier in the bit plane progressive transmission of the SPIHT bit stream. Since SPIHT algorithm always encodes a sign bit for significant coefficients, no extra bit budget is necessary for being bipolar. By terminating the decoding process even at a lower bit rate, the quality improves, both subjectively and objectively, because of the inclusion of some detail information at that rate.

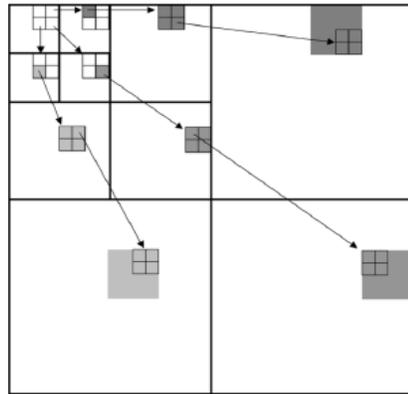


FIGURE 1. Spatial Orientation tree in 2D SPIHT (source [8])

3.2 Modified SPIHT

The parent offspring dependency and corresponding spatial orientation trees for the SPIHT algorithm is as shown in Fig. 1 for three levels of decomposition. Each node of the tree is represented by its coordinates in the algorithm. The tree is defined in such a way that each node has no offspring (leaves) or four off springs at the same spatial location in the next finer level [4].

In original SPIHT, the pixels in the coarsest level e.g., LL sub band of the 3rd level for Fig. 1, are the tree roots and they are grouped into blocks of 2x2 adjacent pixels. In each block one (the left top corner) pixel has no descendant. In the initialization stage, the coordinates of all coefficients in the coarsest level are put into LIP list and the coordinates of coefficients having descendants only into the LIS list as D-type sets. We have modified the initialization stage of the SPIHT algorithm to accommodate the principle of zero shifting as follows: if the decomposition level is such that the coarsest level LL subband consists of a single pixel (e.g., for an image of 512x512 size decomposed into 9 level), then we set the LIS list with the HL, LH, and HH coordinates of the coarsest level (each of which is also a single pixel). Otherwise, we initialize the LIS list with all the coordinates of the LL sub band as of the LIP list. Accordingly, in the sorting pass for LIS, the SPIHT first tests the significance in the LH, HL, and HH band of the lowest subband. For each root in LL, it compares the maximum value between the other three bands at the same spatial orientation and sorts it in the output bit stream. Once this test is over, the set partition rule of SPIHT [6] divides the remaining descendants into either D-type set or L-type set. The decoder duplicates the path of execution of the encoder. At the encoder we are at a liberty to encode the entire bit plane without any bit budget constraint. Experiments carried out over different type of images with spatially different features show that the proposed modification provides a faster codec than the original one while resulting almost the same PSNR values.

3.3: Fixed Context Modeling

SPIHT algorithm generates significant information during encoding of LIS and LIP. LIP is not supposed to generate any redundancies as they are distinctly decorrelated. When LIS of type A becomes significant at certain threshold, the significance information of offspring of 2x2 blocks are generated. We exploited the possibility of such four offspring being significant. Assigning 0 to insignificant and 1 to significant coefficient, we created 16 possible combination of such configuration, starting from 0000 to 1111. We called each configuration a context and found the probability of occurrence of each context model (Fig. 6A and 6B). The systematic trend found in

probability distribution of such context has been utilized for further compression of the image with the help of Huffman coding which has a lower complexity than the arithmetic coding.

3.4 Fixed Context Resolution Scalability (FCRS)

SPIHT algorithm provides SNR (quality) scalability due to its bit plane coding strategy but fails to provide resolution scalability. Each bit plane is coded in two passes – Significant pass and then Refinement pass. Each pass generates the bitstream as a whole in an interleaved fashion without distinguishing between any resolutions so needed (Fig 2(a)). If the decoder wants to decode an image at a given resolution then all the information carrying bits of that resolution should be available at one place. In addition, the knowledge of boundary of separation of different resolution should be available. We have modified the bitstream structure in a way so as to gather all the information pertaining to a particular resolution at a place as shown in Fig.2 (b), where we have shown only three resolutions as per our simulation. It is to be noted that we have incorporated the context modeling into the SPIHT algorithm. Here, the bitstream for both the passes are adjacent to each other corresponding to different individual resolution. Thus, coding of each bit plane should be split in parts depending upon the specific resolution required and each part should contain the corresponding information generated by both the significance and refinement passes for that resolution.

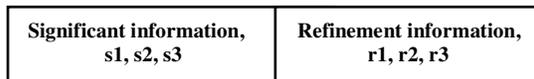


FIGURE 2(a). Structure of output bit stream

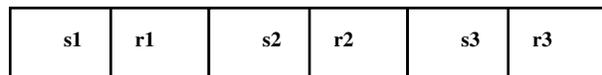


FIGURE 2(b). Modified Structure

For each resolution, the sorting pass is carried out following the refinement pass for all the threshold values and kept in the output bitstream as per the progressive bit plane coding. Starting from the resolution 1, the algorithm encodes the entire image and generates a header which carries the information regarding the boundary for a particular resolution. Say, for resolution 1, we process those LIS of type A whose offspring are also lying in the resolution level 1 only. Similarly, while processing LIP, we are processing only those coefficients which belong to the same resolution. The header, which stores all the boundary information, is sent to the decoder. The reconstruction at the decoder at a particular resolution level is done using the header information.

4. THE CODEC STRUCTURE

The well known SPIHT algorithm encodes the wavelet transformed coefficient in a bit plane manner and outputs the significant information with respect to a specified threshold and its sign. We pre processed the pixel values by shifting downwards by 2^{N-1} , N being the no of bits in the original image, and then applied the 2D wavelet on it. The transformed coefficient is then encoded with 2D SPIHT coder (Fig.3).

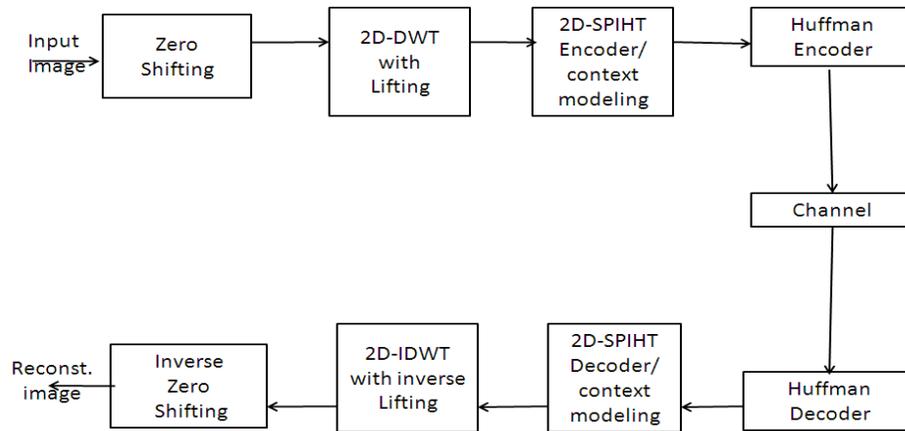


FIGURE 3. Proposed image codec with zero shifting

In the decoder, the execution path of the encoder is duplicated and the values of inverse transformed coefficients are shifted upwards by $2^{(N-1)}$ to get back the original image.

5. RESULTS AND DISCUSSION

5.1 Quality Measurement Metrics

During the process of said compression, the reconstructed image is subject to a wide variety of distortion. Subjective evaluations emphasizes on the visual image quality, which is too inconvenient, time consuming, and complex. The objective image quality metrics like Peak Signal to Noise Ratio (PSNR), or Mean Squared Error (MSE) and structural Similarity (SSIM) [2] are thought to be the best for the image processing application.

The MSE metric is most widely used for it is simple to calculate, having clear physical interpretation and mathematically convenient. MSE is computed by averaging the squared intensity difference of reconstructed image, \hat{x} and the original image, x . Then from it the PSNR is calculated. Mathematically,

$$MSE = \frac{1}{M \times N} \sum [x(i,j) - \hat{x}(i,j)]^2, \quad (1)$$

Where, $M \times N$ is the size of the image and assuming the grey scale image of 8 bits per pixel (bpp), the PSNR is defined as,

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (2)$$

But, they are not very well matched to human visual perception. The authors in [2] have shown that even for the same PSNR values, the visual quality for two reconstructed images is different at different bit rate. It is because the pixels of the natural images are highly structured and exhibit strong dependencies. The SSIM system separates the task of similarity measurement into three comparisons: luminance, contrast, and structure. The overall similarity measure is defined as [2,17]

$$S(x, \hat{x}) = f(l(x, \hat{x})) \cdot c(x, \hat{x}) \cdot s(x, \hat{x}) \quad (3)$$

Where, x is the original assuming to be completely perfect and \hat{x} is the reconstructed image such that both are non negative images. $S=1$ implies a perfect similarity between the images.

$l(x, \hat{x})$ = luminance comparison function, which in turn the function of the mean values of x and \hat{x} ;

$c(x, \hat{x})$ = contrast comparison function and they are function of the standard deviation of x and \hat{x} ; and

$s(x, y)$ = structure comparison function

5.2 Algorithm Complexity in Terms of Encoding/decoding Time

We will present some evidence of our implementation with Matlab in this section. Experiments are performed with grey scale 'Lena' and 'Ship' image for their established popularity. Ship image contains more details than Lena image. All experiments are done with Pentium IV, 1.76 GHz, 512 MB RAM. We have considered two type of filters for their establish popularity in the image compression application. They are 'bior4.4' and 'cdf9/7' with half point symmetry at the boundary.

Level of decomposition	With BM	With CM	With BMZ	With CMZ	With BO	With CO	With BOZ	With COZ
3	3.4735	3.2595	3.1445	2.9640	35.6672	34.6788	40.9529	41.2126
4	2.1118	2.0357	1.8985	1.9531	68.7433	739693	75.9793	86.8388
5	1.6166	1.5989	1.5829	1.6727	87.8300	88.0677	102.1387	88.9418
6	1.4674	1.5062	1.4425	1.4883	90.3934	92.3333	89.8692	96.6292
7	1.4110	1.5085	1.4085	1.4189	87.7200	93.4404	94.7028	93.2454
8	1.3989	1.5181	1.3745	1.4009	95.7797	93.2782	97.9520	94.9666
9	1.8483	1.4252	1.4139	1.4035	98.4183	104.0956	95.5076	110.1410

B (C) M (O) : With Bior4.4 (CDF 9/7) filter for Modified (Original) SPIHT algorithm with 1bpp,
Z : with method of Zero-shifting

TABLE 1: Complexity Comparison in terms of Encoding Time (sec)

It is evident from the Table 1 that, with modification of the algorithm, the time of encoding decreases significantly, i.e., the codec becomes faster irrespective of the type of the filter used. The modified algorithm takes 10 to 90 times less time than that for the original one. Interestingly, the encoding time goes on decreasing as the number of decomposition increases from 3 to 9 (for square image of resolution size 512x512) in steps of 1, whereas the time increases almost linearly with the original one. By the method of zero shifting, the original algorithm takes even more time for encoding than that for without the said pre processing steps. It is found that for the modified algorithm, the encoding time remains almost constant for 6th level of decomposition or more. So for other experiments with the modified algorithm, we have spatially decomposed the image for 8 levels, unless otherwise stated. After verifying the complexity in terms encoding time, we are conducting the following experiment to verify the quality of the reconstructed image with the proposed scheme of zero shifting. The following Table 2 and Table 3 provide some detail of decoding time, the PSNR value and SSIM for the reconstructed LENA image at different bit rate for both the algorithms with 'bior4.4 filter.

Bit rate in (bpp)	Decoding time (sec) without Zero shifting	Decoding time (sec) with Zero shifting	SSIM without Zero shifting	SSIM with Zero shifting	PSNR without Zero shifting	PSNR with Zero shifting
0.01	0.06	0.03	0.4574	0.5298	17.77	20.89
0.11	0.14	0.05	0.8425	0.8464	26.98	28.67
0.21	0.22	0.07	0.9077	0.9125	29.95	31.59
0.31	0.26	0.10	0.9340	0.9435	32.46	32.87
0.41	0.33	0.13	0.9550	0.9532	33.69	34.78
0.51	0.35	0.13	0.9638	0.9671	35.31	35.51
0.61	0.44	0.16	0.9674	0.9716	35.85	36.07
0.71	0.53	0.20	0.9705	0.9744	36.35	36.53
0.81	0.60	0.25	0.9790	0.9784	37.31	37.98
0.91	0.62	0.24	0.9818	0.9826	38.28	38.44

TABLE 2 : Decoding time, PSNR and SSIM for reconstructed LENA with Modified SPIHT

It is to be noted that the modified algorithm encodes the image for all the bitplane; whereas the original is restricted to the specified bit budget. However, it is possible to encode with the modified algorithm with the same bit budget constraints. It is evident from the observations from both these tables that the modification in the algorithm makes the decoding faster. The objective metric like PSNR also improves due to modification in the initialization of the image pixel coordinate and hence in the scanning process. However, the SSIM metric is better for the original one at some bit rate. In the following we are showing the comparison curves for LENA and SHIP images. The experiments are performed with 6 level of spatial decomposition with bior4.4 filter.

5.3 Results with Proposed Image Codec

We have found that 6 or more level of spatial decomposition provides almost constant PSNR for any bit rate. Fig. 4 and 5 shows the performances of PSNR of the Lena and Ship image at different bpp. It is quite evident from the curves that the modified algorithm significantly improves the performances for all bit rates. The improvement is more pronounced in Lena than in Ship, because Ship contains more detail information which are not properly exploited by the SPIHT algorithm. At the lowest possible decoding rate (i.e., 0.1 to 0.2), where the perceptual quality is significantly good, our zero shifting scheme provides almost 0.5 to 1dB higher PSNR than the original one. Experiments show that at 0.15 bpp the visual quality is quite good. Further, on wavelet transforming the shifted pixel values, coefficients are generated which are mostly decorrelated in sign. Experimental observations reveal that these correlations are more pronounced in the lowest frequency sub band.

Bit rate in (bpp)	Decoding time (sec) without Zero shifting	Decoding time (sec) with Zero shifting	SSIM without Zero shifting	SSIM with Zero shifting	PSNR without Zero shifting	PSNR with Zero shifting
0.01	0.039	0.043	0.4697	0.4093	16.72	20.59
0.11	1.005	0.953	0.8590	0.8622	28.30	29.4
0.21	2.501	2.64	0.9272	0.9221	31.61	32.38
0.31	4.483	5.739	0.9536	0.9517	33.69	34.11
0.41	9.178	9.002	0.9639	0.9629	35.03	35.60
0.51	12.995	17.53	0.9729	0.9728	36.37	36.62
0.61	21.854	24.955	0.9776	0.9765	37.12	37.39
0.71	33.991	38.809	0.9804	0.9795	37.83	38.05
0.81	47.440	49.256	0.9830	0.9828	38.46	38.74
0.91	58.591	65.62	0.9855	0.9857	39.18	39.37

TABLE 3: Decoding time, PSNR and SSIM for reconstructed LENA with Original SPIHT

Since in the process of SPIHT encoding we are at a liberty to invest an extra bit always for testing any significance, the reconstruction at the lower bit rate is better. Also, due to efficient grouping of the insignificant information in LIS with our proposed scheme, the no of bits to be transmitted to decoder also reduced to a factor of 4. Hence truncating the bit stream at lower bit rate provides much of the significant information to the decoder and the reconstruction quality also improves.

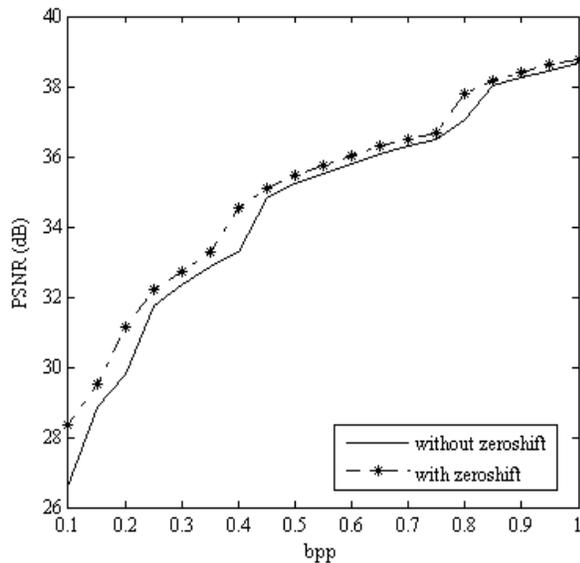


FIGURE 4. Comparison of Lena image

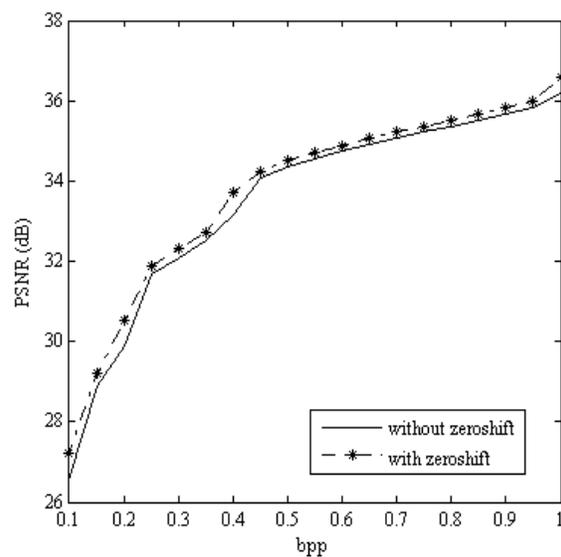


FIGURE 5. Comparison of Ship image

5.4 Result With Context Modeling and Scalability

Extensive experiments are carried out with lot many images to find the pattern of the probability distribution of the significant coefficient in the 2x2 blocks. We found that the distribution of almost all the images of varying characteristics show a fixed pattern. The distribution for two characteristically different grey scale images are shown; Lena image with much less detail within the sub band (Fig. 6A), and Ship image with much high details (Fig. 6B). By taking the average of this probability distribution we model that context.

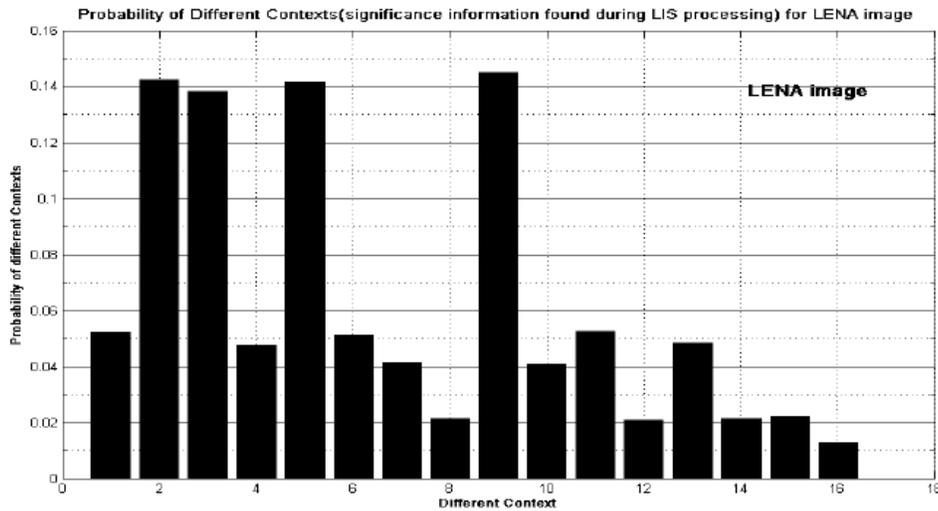


FIGURE 6A. Context Count for Lena

We prepared a table to store the generated Huffman codes corresponding to each model and at the time of encoding, we utilized this table directly. We compared the result of this context modeling with that of the arithmetic encoding. We define 128x128 sizes as resolution level 1, 256x256 as resolution level 2 and 512x512 as resolution level 3. All results are obtained by applying 6 level of spatial wavelet decomposition with 'bior 4.4' filter. The modified SPIHT encoder with the method of zero shifting is set to produce a bit stream that supports three number of spatial scalability levels as mentioned above. The PSNR calculation is carried out with the following formula:

$$PSNR = \frac{10 \log_{10} (255 \times 2^{(s-resolution)})^2}{MSE}, db \tag{4}$$

Where, MSE is the Mean Squared Error calculated between reconstructed image at a particular resolution and the original image of the same size as in eqn (2).

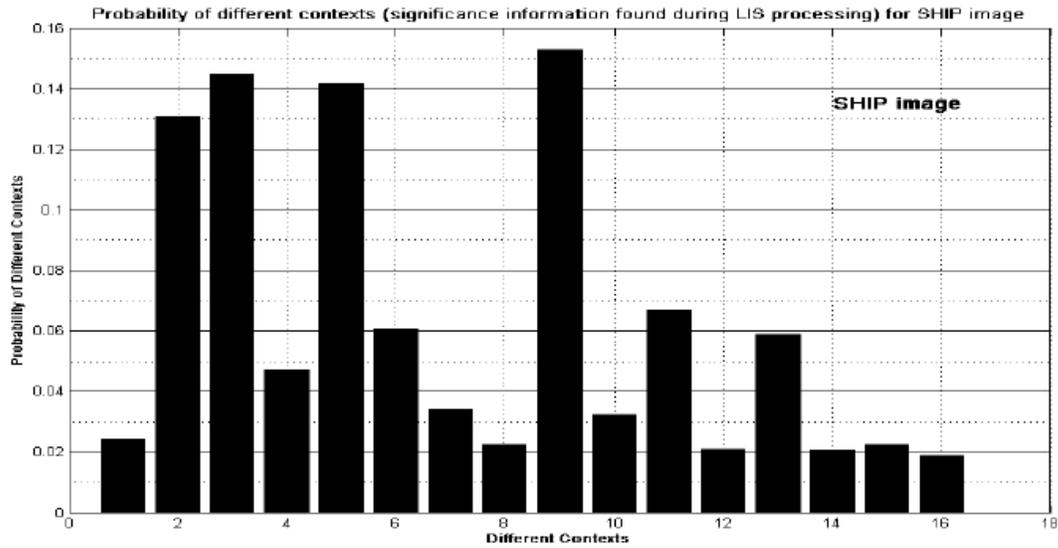


FIGURE 6B. Context Count for Ship

The rate distortion curves for each resolution in Fig.7 and the corresponding reconstructed LENA images in Fig.8 clearly shows that our proposed encoder completely keeps the progressiveness (by SNR) property of the original SPIHT algorithm. For resolution level 1 and 2, our resolution scalable decoder obtained the proper bit stream tailored by the encoder for that resolution level.

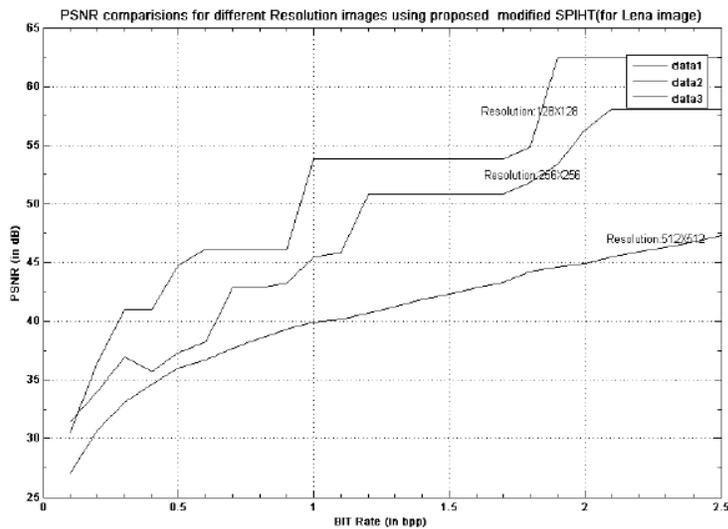


FIGURE 7. Performance comparison of FCRS SPIHT for different resolution



(a) Resolution 1 (128x128)

(b) Resolution 2 (256x256)



(c) Resolution 3(512x512)

FIGURE 8. FCRS reconstructed Lena at 0.5 bpp for 3 different level

As expected, the performance of scalable decoder is much better than for SPIHT and as the resolution level becomes smaller (hence the scale becomes larger), the difference becomes more significant. All the results are obtained with the context modeling of the encoded bit stream.

6. CONCLUSION

In this paper, we presented a novel extension of wavelet based SPIHT coding scheme. The scheme pre processed the pixel values around zero before any wavelet transform takes place. Higher PSNR and more compression are obtained with the scheme without using the popular arithmetic coding. Experimental results show that it performs better measurably and also at lower bit rate its performance is well suited for the purpose of surveillance and monitoring system. We have presented an extension to the original SPIHT by incorporating spatial scalability feature to the encoded bit stream. The bit stream can be stopped at any point to meet a bit budget during the process of coding and decoding for a particular resolution level. To lessen the burden of computation on the codec, we proposed a Huffman coding based context modeling to achieve further compression at each resolution level. This technique can be extended for combined SNR, spatial and frame rate scalable video coding.

7. REFERENCES

- [1] G. Wallace, "The JPEG still picture compression standard," *Communications of ACM*, vol. 34, no. 4, 1991.
- [2] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, Apr., 2004.
- [3] J.M. Shapiro, "Embedded image coding using Zero trees of wavelet coefficients," *IEEE Trans. on Signal Processing*, vol.41, pp. 3445-3462, Dec.,1993.
- [4] A.Said and W.A.Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. on Circuits and Systems for Video Technology*, vol.6, pp. 243-250, June, 1996.

- [5] D. Taubman, "High Performance Scalable Image Compression with EBCOT," *IEEE Trans. on Image Processing*, vol. 9, pp. 1158-1170, July, 2000.
- [6] B.-J. Kim and W. A. Pearlman, "An embedded video coder using three-dimensional set partitioning in hierarchical trees (SPIHT)," *in proc. IEEE Data Compression Conf.*, pp. 251-260, Mar., 1997.
- [7] J. Karlenkar and U. B. Desai, "SPIHT video coder," *in Proc. IEEE Region 10 International Conference on Global Connectivity in Energy, Computer, Communication and Control, TENCON'98*, vol. 1, pp. 45-48, 1998.
- [8] B.-J. Kim, Z. Xiong, and W. A. Pearlman, "Low bit-rate scalable video coding with 3-d set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circ. and Syst. for Video Technology*, vol. 10, no. 8, pp. 1374-1387, Dec. 2000.
- [9] J. Zho and S. Lawson, "Improvements of the SPIHT for image coding by wavelet transform," *Proc. IEEE Seminar on Time-scale and Time-Frequency Analysis and Applications (Ref. No. 2000/019)*, pp. 24/1 -24/5, 2000.
- [10] E. Khan and M. Ghanbari, "Very low bit rate video coding using virtual spiht," *IEE Electronics Letters*, vol. 37, no. 1, pp. 40-42, Jan. 2001.
- [11] H. Cai and B. Zeng, "A new SPIHT algorithm based on variable sorting thresholds," *in Proc. IEEE Int. Symp. Circuits and Systems*, vol. 5, pp. 231-234, May 2001.
- [12] Yushin Cho, W.A.Pearlman and A. Said, "Low complexity resolution progressive image coding algorithm: progres (progressive resolution decompression)," *IEEE International Conference on Image processing*, Volume 3, pp. 49-52, 2005.
- [13] J.Jytheswar and S. Mahapatra, "Efficient FPGA implementation of DWT and modified SPIHT for lossless image compression," *Journal of System Architecture*, vol. 53, pp.369-378, 2007.
- [14] S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Patt. Rec. and Mach. Int.*, Vol. PAMI-11, No. 7, pp. 2091-2110, July 1989.
- [15] J.R.Ohm, "Three dimensional subband coding with motion compensation," *IEEE Trans. Image Processing*, vol.3, no. 5, pp. 559-571, Sept. 1994.
- [16] W. Sweldens," The Lifting Scheme: A Custom-design Construction of Biothogonal Wavelets," *Journal of Applied Computation Haron Anal*, vol.3, issue 2, pp. 186-200, 1996.
- [17] <http://www.ece.uwaterloo.ca/~z70wang/research/ssim>
- [18] B.Mohanty, P.Verma and S.Mahapatra, "A High Performance SPIHT based Image and Video codec for survelliance," *Int.journal of Signal and Imaging System Engg.*(in press).
- [19] A. Singh, B. Mohanty, P. Verma and S. Mahapatra, "Fixed context resolution scalable SPIHT: a novel extension," *Proc. of International conference on Communication, Computation, Control and Nanotechnology (ICN 2010)*, pp. 28-32, Oct.2010.