ISSN:0975-9646

Improved Image Compression by Set Partitioning Block Coding by Modifying SPIHT

Somya Tripathi[1] ,Anamika Ahirwar[2]
[1] *Maharana Pratap College of Technology, Gwalior, Madhya Pradesh 474006*
[2]*Department of Computer Science & Engineerring.*

Abstract-- The image compression proposed to present a different implementation method based on set partitioning in hierarchical trees (SPIHT), which provides even better performance which is already reported extension of the wavelet based compressions that surpassed the performance of the conventional methods like DCT and run length like coding . The image compression results are calculated by original images for compression ratio (CR) and reconstructed by the decompression methods for peak signal noise ratio (PSNR). In addition of the developed compression and decompression methods are very fast, and the performance can be made faster, with only small loss in performance, by reducing the bitrates values.

Keywords--Image Compression, SPIHT, Wavelet, DSP, Quantization

I. INTRODUCTION:

Image Compression is the technique of reducing the number of bits needed to store or transmit information. The image compression technique involves encoding information using less bits than original information content. Compression can be either loss or lossless. The Lossless compression technique reduces the bits by identifying and removing redundancy. In the lossless compression technique there is no information is lost. However lossy compression technique reduces the bits by identifying unnecessary information and removing it. This process of reducing the size of a data is popularly known as data compression or coding. Coding or compression is very useful because it helps to reduce usage of resource, data storage space or transmission capability. Almost all data compression algorithms consist of at least one model and a code associated with it. In which model estimates the probability distribution and code assigns shorter codes to the more probable symbols. Image Compression or coding is very useful in most situations because the compressed data will save time and the space if it is compared to the un encoded information . There are various types of coding techniques available for compression of information. Out of them here we are using arithmetic coding technique because it will create code word for the entire data altogether, rather than creating code word for each data word.

Arithmetic coding technique is a well-known method for lossless data compression or source coding. The Arithmetic coding is mostly popular in image and video compression and its applications. The arithmetic coding bypasses the idea of replacing an input symbol with its specific code and also it replaces a stream of input symbols with a single floating point number. This arithmetic coding offers the better way to create a code that exactly represents the frequency of any character . This coding technique is the most efficient method to code symbols according to their occurrence probability. Unlike a binary Huffman code tree the arithmetic coding offers a better compression rate as it generates a single symbol rather than several different code words.Arithmetic coding involves a message that is encoded as real number in an interval from zero to one. Arithmetic encoder generates a unique identifier for the sequence of length 'm' to be coded and will assign a unique binary code to this sequence. Since main data structure of arithmetic coding is an interval which represents the string constructed so far as its initial value is either 0 or 1. At each and every stage, the current interval [min, max] is subdivided into sub-intervals corresponding to the probability model for the next given character. Such interval chosen will be the one representing the next character. The more frequently occurring the character, the larger the interval assigned. The output of the coder is a number in the last interval. Here we try to design an arithmetic coder which is to be used in Set Partitioning in Hierarchical Trees (SPIHT) encoder for further compression of the discrete wavelet based decomposed images as per requirement. After this SPIHT transformation some regularities will present in the file . These regularities can be allow us for further file compression.

The SPIHT algorithm is one of the best algorithm present for the compression. We are using Wavelet transform SPIHT algorithm to encode the coefficients of wavelet. The SPIHT algorithm involves comparison of two values at a time which results in yes/no condition. According to this sorting pass coefficients are categorizes into 3 lists: LIS (List of Insignificant Sets) are the set of coefficients having their magnitude smaller than the threshold taken .And LIP (List of Insignificant Pixels) are the pixels having magnitude smaller than the threshold considered. The other one is LSP (List of Significant Pixels) which are the pixels whose magnitude is greater than that of considered threshold. This coding method can obtain by optimal performance of its ability to generate codes with fractional bits and it is widely used in various image compression techniques. More importantly the set partitioning in hierarchical trees uses an Arithmetic coding method to improve its peak signal to noise ratio value. Here we trying to design an arithmetic coder for SPIHT to increase the performance and further result in good compression of the image.

II. SPATIAL ORIENTATION TREE:

Most of an images energy is concentrated in its low frequency components. Consequently, the variation decreases as we move from the highest levels to the lowest levels of the sub band pyramidal structure. Also, it has been observed that there is a spatial self-similarity between existing sub bands, and their coefficients are expected to be better magnitude if we move downward in the pyramid following the same orientation in space. For an instance, large low-activity areas are supposed to be identified in the highest levels of the pyramidal structure, and then they are replicated in the lower levels at the same spatial locations [6].

Tree structure, called spatial orientation tree,which is naturally defines the spatial relationship existing on the hierarchical pyramid. Fig. 1 clearly shows that how our spatial orientation tree is defined in a pyramid constructed with recursive splitting of four-subband. Every node of the tree corresponds to a pixel, and it is identified by the pixel coordinate than the direct descendants (offspring) correspond to the pixels of the same spatial orientation in the next level of the considered pyramid. The tree that defined in such a way that each node has either no offspring (the leaves) or have four offsprings, which always form a group of 2X2 neighbouring pixels. In Fig. 1 we see the arrows are oriented from the parent node to its four obtained offspring. The pixels that are in the highest level of the pyramid are the tree roots and are also grouped in 2X2 adjacent pixels as shown. However, their relative offspring branching rule is different, and hence in each group one of them (indicated by the star in Fig. 1) which has no descendants. The sets of coordinates are used to present the new coding method are as follows:

- O(i;j): is the set of coordinates of all the offspring node (i; j);
- D(i; j):is the set of coordinates of all the descendants of nodes (i; j);
- H:is the set of coordinates of all the spatial orientation tree roots (nodes in the highest pyramid level);
- L(i,j) = D(i, j) - O(i, j)

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

O(i,j) = {(2i.2j) , (2i . 2j+10 , (2i + 1 , 2j) , (2i +1 , 2j +1)}

We use some parts of the spatial orientation trees as the partitioning subsets in the sorting SPIHT algorithm. The set partitioning rules are simple as given below:

1. The initial partition is performed with the sets {(i; j)} and D(i; j), for all (i; j) Є H;
2. if D(i; j) is significant then it is again partitioned into L(i; j) plus the four single-element sets with (k; l) 2 O(i; j).
3. if L(i; j) is significant then it is again checked and partitioned into the four sets D(k; l), with (k; l) Є O(i; j).

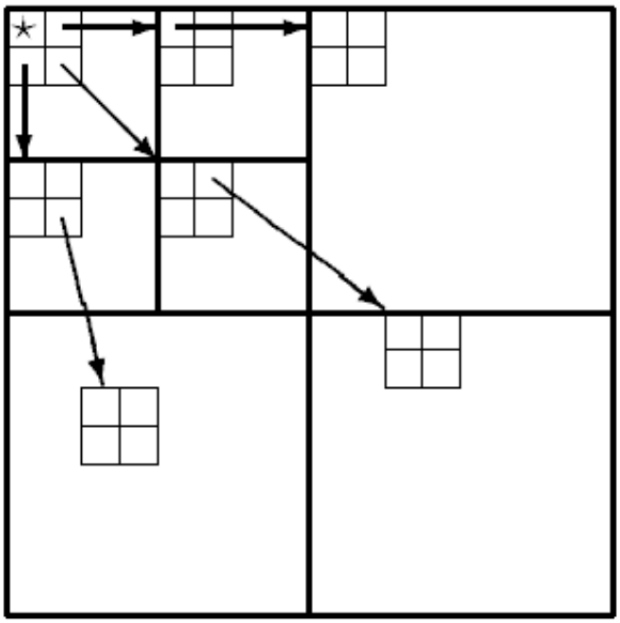

Fig 1: spatial orientation tree.

III. CODING ALGORITHM:

The order in which the subsets are checked for significance is very important, in every practical implementation the significance information is stored in three ordered lists, generally known as the list of insignificant sets (LIS), list of insignificant pixels (LIP), and the last is list of significant pixels (LSP). In all these lists each entry is identified by its coordinate i.e (i; j), which is in the LIP and LSP individual pixels, and for the LIS it is represented by the set D(i; j) or by L(i; j).By doing differentiate between them we say that a LIS entry is of type A, if it is represented as D(i; j), is of type B if it is represented as L(i; j).

While the sorting pass the pixels present in the LIP set which were insignificant in the previous pass are tested accordingly, and those pixels are found significant than moved to the LSP set.

Similarly, the sets which are sequentially evaluated the following LIS order, and whenever is found a significant set it is removed from the list and is partitioned again and than all the new subsets formed with more than one element are added to back of LIS, whereas the single-coordinate sets are added to the end of the LIP or at the LSP, depending on whether they are found insignificant or are significant respectively. So formed LSP contains the coordinates of the pixels that are visited in each refinement pass.

Here we propose the new encoding algorithm.

ALGORITHM :

1. Initialization:

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

2. Shorting Pass:

2.1. for each entry (i; j) in the LIP do the following:
2.1.1.for output Sn(i; j);
2.1.2. if Sn (i; j) = 1 then move (i; j) to LSP and the output sign of ci;j ;
2.2. for each entry (i; j) in the LIS do the following:
2.2.1. if the entry is of type A then
For output Sn(D(i; j));
if Sn (D(i; j)) = 1
for each (k; l) Є O(i; j) do:
The output Sn(k; l);
if Sn(k; l) = 1 then (k; l) is added to the LSP and output the sign of ck;l;
if Sn(k; l) = 0 then (k; l) is added to the end of the LIP;
if L(i; j) 6= ; then (i; j) is moved 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; In
2.2.2. if the entry is found to be of type B then
output Sn(L(i; j));
And if Sn (L(i; j)) = 1 then
each (k; l) 2 O(i; j) is added to the end of the LIS as an entry of type A;
and (i; j) is removed from the LIS
.

3. **Refinement pass:**

for every entry (i; j) in LSP, except those which are included in the last sorting pass (i.e., with same n), the n-th most significant bit of $|c_{i;j}|$ is presented as output;

4. **Quantization-step update:**

n is decremented by 1 and then go to Step 2.

One of the most important characteristic of the algorithm is that the entries added to the end of the LIS in Step 2.2 are evaluated before that same sorting pass finished. So, whenever we say that for each entry in the LIS" we also mean that those are being added to its end. In this algorithm the rate can be controlled because the transmitted information consists of single bits. The encoder can also use the property in equation number (4) to estimate the reduction in progressive distortio and stop at some desired distortion level.

Note that in the algorithm all branching conditions based on the significance data S_n which can only be calculated with the knowledge of (ci;j) are given at the output by encoder. Hence, to obtain the desired algorithm at decoder, which only duplicates the execution path of the encoder as it sorts the significant coefficients only, we just simply have to replace the words output by input in Algorithm . However note that whenever the decoder inputs the data, the three control lists of decoder (LIS, LIP, and LSP) are identical to the ones used by the encoder at the moment it gives that data at its output, it means that the decoder indeed recovers the ordering from execution path. Now it becomes easy to see that with this scheme coding and decoding both have the same computational complexity.

Another additional task done by decoder is to update the reconstructed image. For every value of n when a coordinate is moved to LSP, and it is known that $2^n \leq |ci;j| < 2^{n+1}$. So,when the decoder uses that information, and the sign bit i.e input just after the insertion in the LSP,for $c_{i;j}$ = $\pm 1{:}5X2^n$. Similarly, during the refinement pass the decoder adds or subtracts 2^{n-1} from $c_{i;j}$ when it inputs the bits of the binary representation of value $|c_{i;j}|$. In this way the distortion gradually decreases during both of the sorting and the refinement passes.

As in any other coding method, efficiency of Algorithm can be improved by entropy-coding of its output. Practical experiments shows that normally there is little to be gained by entropy-coding the coefficient signs or the bits that are put out during refinement pass. On the other side, the significant values are not having equal probablity, and hence there exists a statistical dependence between $S_n(i; j)$ and $S_n(D(i; j))$ value, and this is also present between the significance of adjacent pixels.

We have done this dependence using the adaptive arithmetic coding algorithm.In order to increase the coding efficiency,the groups of 2X2 coordinates were kept together in the mentioned lists, and their significance values were coded as a single symbol by the arithmetic coding algorithm as discussed. As the decoder only needs to know the transition from insignificant to significant (the inverse is impossible)values, the amount of information which is needed to be code, changes according to a number m of insignificant pixels present in that group, and for each case it can be delivered by an entropy-coding alphabet with max of 2m symbols. By arithmetic coding it is cleared to use several adaptive models, each with 2^m symbols, where m Є {1; 2; 3; 4}, in order to code the information in a group of four pixels.

By arithmetic coding the significant pixels information together gives the average bit rate corresponds to a m-th order entropy. At the same time, by using different models for the different number of insignificant pixels, each adaptive model contains probabilities conditioned to the fact that a certain number of adjacent pixels are either significant or insignificant .In this way dependence between magnitudes of adjacent pixels is completely exploited. The above scheme was also used to code the significance of trees rooted in groups of pixels of size 2X2.

With the arithmetic entropy-coding it is still possible to produce a coded file with almost the exact code rate, and probably a few unused bits to pad the file to desired size.

IV. RESULT AND DISCUSSION:

In this paper we have shown that the results are obtained by our ASPIHT compression method over the coloured images. We have taken five different images and tested our algorithm at different bit rates for our image compression . The images which we have tested is Figure 4.1(a) “heart.jpg” and the bit rates that we have considered are:

1).0625
2)0.125
3)0.25
4)0.5
5)1

At every time our algorithm generates the reconstructed image along with the PSNR and CR values. These results are condensed and shown in the next sections of this chapter one by one.

4.1 Compression of image "heart.jpg"

The heart.jpg image is resized as image of size 512X512 using the MATLAB command thereafter we have applied the image compression algorithm ASPIHT and reconstructed image is shown at different bit rates. The original image is shown in fig. 2(a) for reference and comparison purpose. The reconstructed image at different bit rates are shown in fig 2(b) to fig.2(f).

Fig. 2(a) Original image 'heart.jpg'

Fig. 2(b) Reconstructed Image at bit rate =0.0625

Fig. 2(c) Reconstructed Image at bit rate =0.125.

Fig. 2(d) Reconstructed Image at bit rate =0.25

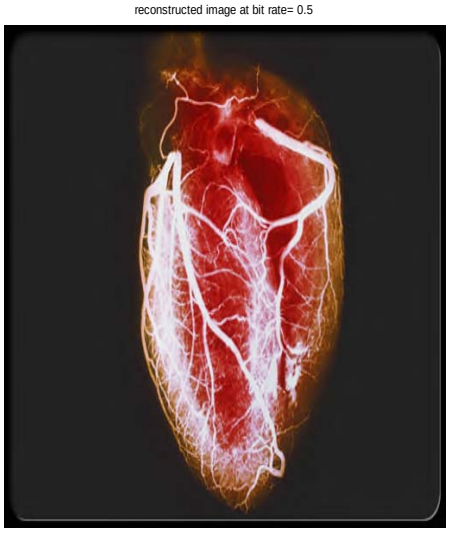

Fig. 2(e) Reconstructed Image at bit rate =0.5.

Fig. 2(f) Reconstructed Image at bit rate =1

Table 1. Analytical Results for Image compression for image heart.jpg.

Bit Rate	R	G	B	Average value
.0625	PSNR = 29.0021 CR = 128	PSNR = 26.8883 CR = 127.98	PSNR = 28.4326 CR = 127.99	PSNR = 28.1076 CR = 127.99
.125	PSNR = 31.1587 CR = 64	PSNR = 29.8257 CR = 64	PSNR = 29.4107 CR = 64	PSNR = 30.1317 CR = 64
.25	PSNR = 34.7756 CR = 32	PSNR = 33.8756 CR = 32	PSNR = 33.5586 CR = 32	PSNR = 34.069 CR = 32
.5	PSNR = 40.1346 CR = 16	PSNR = 39.5307 CR = 16	PSNR = 39.2756 CR = 16	PSNR = 39.649 CR = 16
1	PSNR = 48.197 CR = 8	PSNR = 47.6962 CR = 8	PSNR = 47.3008 CR = 8	PSNR = 47.731 CR = 8

We can see that for bit rate 0.0625 reconstruction of the image is blurred. However the blurring has been less significantly as we move from bit rate 0.0625 to bit rate 1 that is, from fig. 2(b) to fig.2(f). Although we observed that the image reconstructed from bit rate 0.25 is very much similar to that of the original image fig, 2(a).To analyze the similarity between the reconstructed image and the original image we have considered two parameters that are PSNR and CR obtained for three different planes viz. R, G, B for all the bit rates given in table 1:

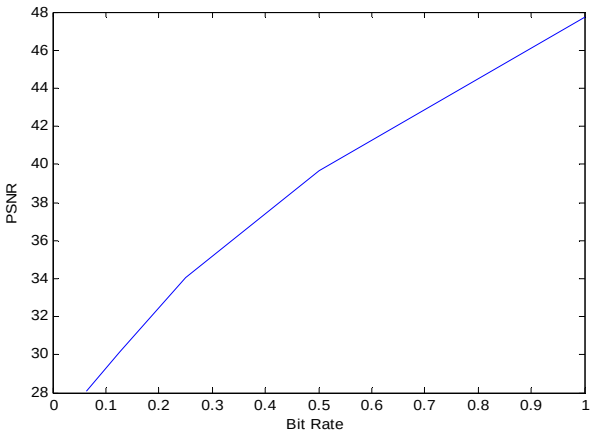

Figure 3: PSNR vs. Bit rate plot using table 1.

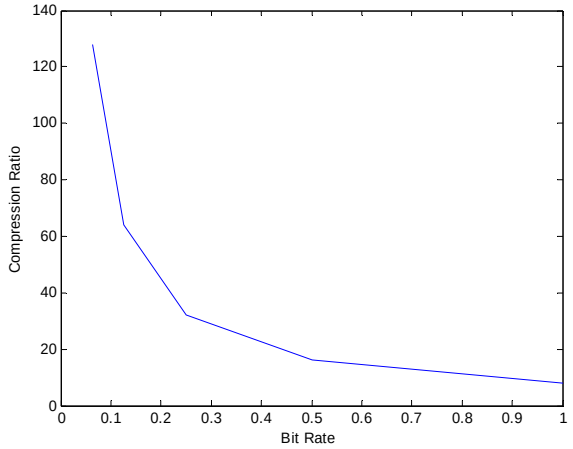

Figure 4: Compression Ratio vs. Bit rate plot using table 1.

We have plotted PSNR and Compression ratio with respect to the Bit rate in figure 3 and 4.it can be observed that PSNR above than 30 can be obtained at bitrate above than 0.1.since PSNR=30 is acceptable value for decompressed quality hence if we see figure 4 it can be observed that the compression ratio of 100 times can be obtained with image acceptable image quality.

However our main objective is to provide high degree of image compression. For justifying this we have also shown the CR for the reconstructed images as shown in fig.2(b to f).

Since the compression is performed for R, G, B planes to analyse the result we have calculated the average values of PSNR and Cr. We can see that the average PSNR varies from 28 to 47 and the corresponding CR varies from 8 to 128 times.

However the acceptable PSNR values are nearly 31 to35 and we can see that the image at bit rate 0.25 is giving PSNR of 34.069 with corresponding CR of 32 which is appropriate value for our reconstructed image after compression.

V. CONCLUSION:

In our work we have done that the application of an advance wavelet transform based on image compression technique using SPIHT coding algorithm and is tested for several color images belonging to different file format. The performance of our algorithm is tested at different bit rates.

And finally results are tabulated in terms of PSNR and CR for justifying the performance of our proposed image compression. It has been observed that the best suitable PSNR is in the range of 28 to 47 and CR is in the range of b 8 to 128,which is appropriate value to reconstruct the original image as per the human perception and to retrieve the max.

Hence it can be concluded that our algorithm is capable of reconstructing the color image successfully at different bit rates .

In future we can explore the performance of our algorithm, at various decomposition level of the wavelet transform.

There are advanced versions of DWT known as fractal based wavelets and LWT, that can also be applied to enhance the performance of our proposed algorithm.

REFERENCES:

[1] Keun-hyeong Park and HyunWook Park, "Region-of-Interest Coding Based on Set Partitioning in Hierarchical Trees" IEEE transactions on circuits and systems for video technology, VOL. 12, ISSUE 2 , February 2002, PP.106-113,ISSN:1051-8215.

[2] William A. Pearlman et al, "Efficient, Low-Complexity Image Coding with a Set-Partitioning Embedded Block Coder", VOL.14,ISSUE 11,2004,PP.1219-1235,ISSN:1051-8215 - ieeexplore.ieee.org.

[3] Hala H. Zayed et al, "3D Wavelets with SPIHT Coding for Integral Imaging Compression" IJCSNS International Journal of Computer Science and Network Security, VOL.12,ISSUE 1, January 2012,PP.126-133,ISSN:1110-6409.

[4] Monauwer Alam and Ekram Khan, "listless highly scalable set partitioning in hierarchical trees coding for transmission of image over heterogenous networks" International Journal of Computer Networking, Wireless and Mobile Communications (IJCNWMC), VOL.2, ISSUE 3, Sep 2012,PP. 36-48,ISSN:2250-1568.

[5] Mahesh Chandra et al, "A Multiple Description Coding Method Based On Set Partitioning In Hierarchical Tree Algorithm For High Definition Image" International Journal of Electrical and Electronics Engineering Research (IJEEER) ,VOL. 3, ISSUE 1, Mar 2013,PP. 55-60,ISSN:2250-155X.

[6] Venkatesh VC et al, "SPHIT Algorithm combined with Variable length encoder to enhance the performance of the image compression technique" International Journal Of Engineering Sciences & Management, VOL.3,ISSUE 2, April-June, 2013,ISSN:2277-5528.

[7] A. Sreenivasa Murthy et al, "Performance Analysis Of Set Partitioning In Hierarchical Trees (Spiht) Algorithm For A Family Of Wavelets Used In Color Image Compression" ICTACT Journal On Image And Video Processing, VOL.05,ISSUE02, November 2014, PP.932-936.

[8] Meenu Roy and N.Kirthika, "Design of Coder Architecture for Set Partitioning in Hierarchical Trees Encoder" International Journal of Current Engineering and Technology, VOL.4,ISSUE 3,June 2014,PP.1200-1205,ISSN:2270-4106:PISSN:2347-5161.