Gray-Scale Image Compression Using DWT-SPIHT Algorithm

S.Narasimhulu[1], M.Tech,
Dept of ECE,
S .V.U College of Engineering,
Tirupati, India.

Dr.T.Ramashri[2],
Associate Professor, Dept of ECE,
S.V.U College of Engineering,
Tirupati, India.

Abstract—

SPIHT is computationally very fast and among the best image compression algorithms known today. According to statistic analysis of the output binary stream of SPIHT encoding, propose a simple and effective method combined with Huffman encode for further compression. In this paper the results from the SPHIT algorithm are compared with the existing methods for compression like discrete cosine transform (DCT) and discrete wavelet transform (DWT).

Keywords- Encoding; Decoding; DCT; DWT; SPIHT; Huffman coding

I. INTRODUCTION

The discrete cosine transforms (DCT) [1] is a technique for converting a signal into elementary frequency components. It is widely used in image compression. Here we develop some simple functions to compute the DCT and to compress images [2]. These functions illustrate the power of Mathematical in the prototyping of image processing algorithms.

In recent years, wavelet transform [3][4] as a branch of mathematics developed rapidly, which has a good localization property[5] in the time domain and frequency domain, can analyze the details of any scale and frequency. So, it superior to Fourier and DCT. It has been widely applied and developed in image processing and compression.

Wavelet Transform (WT) has received more and more significant attention in signal compression. However, many differences lie in the performance of different wavelets. There is a need to select the optimal matched wavelet bases to analyze the signal and the signal needs to be expressed with the fewest coefficients, i.e. sparse coefficients. The signal compression with wavelet is a procedure in which the input signal is expressed with a sum of a few of power terms for wavelet function. The more similar the bases function is to input signal, the higher the compression ratio is. But, at higher compression ratios we may experience more errors, i.e. mean square error will be high at the receiving end and hence PSNR will be very low.

More improvements over DWT are achieved by SPIHT [6][7], by Amir Said and William Pearlman, in 1996 article, "Set Partitioning in Hierarchical Trees"[5]. In this method, more (wide-sense) zero-trees are efficiently found and represented by separating the tree root from the tree, so, making compression more efficient. Experiments are shown that the images through the wavelet transform, the wavelet coefficients' value in high frequency region are generally Small , so it will appear seriate "0" situation in quantify [8]. SPIHT does not adopt a special method to treat with it, but direct output. In this paper, focus on this point, propose a simple and effective method combined with Huffman encode for further compression. A large number of experimental results are shown that this method saves a lot of bits in transmission, further enhanced the compression performance.

II. SPIHT ALGORITHM

A. Description Of The Algorithm

Image data through the wavelet decomposition, the coefficient of the distribution turn into a tree. According to this feature, defining a data structure: spatial orientation tree. 4-level wavelet decomposition of the spatial orientation trees structure are shown in Figure1.We can see that each coefficient has four children except the 'red' marked coeffcients in the LL subband and the coeffcients in the highest subbands ($HL1;LH1; HH1$).

The following sets of coordinates of coeffcients are used to represent set partitioning method in SPIHT algorithm. The location of coeffcient is notated by (i,j),where i and j indicate row and column indices, respectively.

H: Roots of the all spatial orientation trees

$O(i, j)$:Set of offspring of the coeffcient (i, j), $O(i, j) = \{(2i, 2j), (2i, 2j + 1),(2i + 1, 2j), (2i + 1, 2j + 1)\}$, except (i, j) is in LL; When (i,j) is in LL subband, $O(i; j)$ is defined as: $O(i, j) = \{(i, j + w_{LL}), (i + h_{LL}, j), (i +h_{LL} , j + w_{LL})\}$, where w_{LL} *and* h_{LL} is the width and height of the LL subband, respectively.

$D (i, j)$: Set of all descendants of the coeffcient (i, j),
$L (i, j)$: $D (i, j) - O (i, j)$

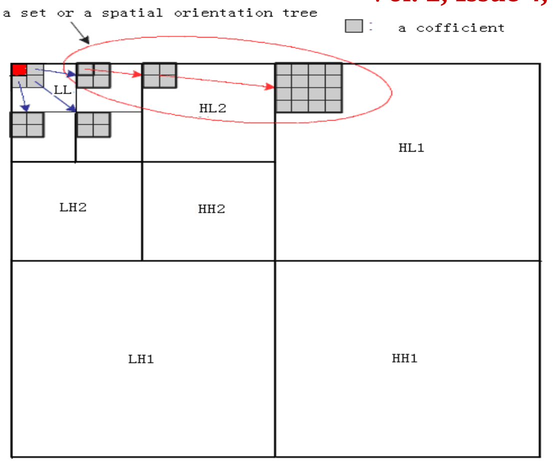


Figure1 Parent-child relationship in SPIHT

A significance function $S_n(\tau)$ which decides the Significance of the set of coordinates, τ, with respect to the threshold 2^n is defined by:

$$\boldsymbol{S_n(\tau)} = \begin{cases} \boldsymbol{1, if\ \ max_{(i,j)\in\tau}\{|c_{i,j}|\} \geq 2^n} \\ \boldsymbol{0, else} \end{cases}$$

Where $c_{i,j}$ is the wavelet coefficient.

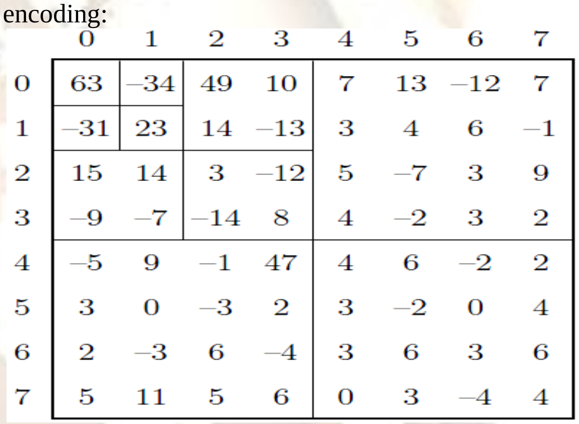
In this algorithm, three ordered lists are used to store the significance information during set partitioning. List of insignificant sets (*LIS*), list of insignificant pixels (*LIP*), and list of significant pixels (*LSP*) are those three lists. Note that the term 'pixel' is actually indicating wavelet coeffcient if the set partitioning algorithm is applied to a wavelet transformed image.

Algorithm: SPIHT

```
1) Initialization:
        1. output n= [log$_2$ max{|($c_{i,j}$)|}]
        2. set LSP =∅;
        3. set LIP = (i,j) ∈ H;
        4. set LIS = (i,j) ∈ H, where D(i; j) ≠ ∅ and
        set each entry in LIS as type A ;
2) Sorting Pass:
        1. for each(i, j) ∈□ □LIP do:
        (a) output $S_n(i,j)$
        (b) if $S_n(i,j)$ = 1 then move (i, j) to LSP and
        output Sign ( $c_{i,j}$ )
        2. for each (i, j) ∈ LIS do:
        (a) if (i, j) is type A then
                i. output $S_n(D(i,j))$
                ii. if then$S_n(D(i,j))$ = 1 then
        A. for each (k, l) ∈ O(i, j)
        . output $S_n(k,l)$
        . if $S_n(k,l)$ = 1 then append (k, l) to LSP,
        output Sign($c_{k,l}$),and $c_{k,l} = c_{k,l} - 2^n$
sign($c_{k,l}$)
                .else append (k; l) to LIP B. move (i, j) to the
                end of LIS as type B
                (b) if (i, j) is type B then
                        i. output $S_n(L(i,j))$
                        ii. if $S_n(L(i,j))$ = 1 then
                . append each (k, l) ∈ O(i, j) to the end of LIS
                as  type A
                . remove (i,j) from LSP
3) Refinement Pass:
                1. for each (i,j) in LSP, except those included
                in the last sorting pass
                 . output the n-th MSB of |$c_{i,j}$|
4) Quantization Pass:
                1. decrement n by 1
                2. goto step 2)
```

B. Analyses of SPIHT Algorithm

Here a concrete example to analyze the output binary stream of SPIHT encoding. The following is 3-level wavelet decomposition coefficients of SPIHT encoding:



	0	1	2	3	4	5	6	7
0	63	−34	49	10	7	13	−12	7
1	−31	23	14	−13	3	4	6	−1
2	15	14	3	−12	5	−7	3	9
3	−9	−7	−14	8	4	−2	3	2
4	−5	9	−1	47	4	6	−2	2
5	3	0	−3	2	3	−2	0	4
6	2	−3	6	−4	3	6	3	6
7	5	11	5	6	0	3	−4	4

n = [log$_2$ max {|c(i,j)|}] = 5, so, The initial threshold value:$T_o = 2^5$, for T_o, the output binary stream: 11100011100010000001010110000, 29 bits in all.

By the SPIHT encoding results, we can see that the output bit stream with a large number of seriate "0" situation, and along with the gradual deepening of quantification, the situation will become much more severity, so there will have a great of redundancy when we direct output.

C. Modified SPIHT Algorithm

For the output bit stream of SPIHT encoding with a large number of seriate "0" situation, we obtain a conclusion by a lot of statistical analysis: '000' appears with the greatest probability value, usually will be about 1/4.Therefore, divide the binary output stream of SPIHT every 3 bits as a group, every group recorded as a symbol, a total of eight kinds of symbols, statistical probability that they appear, and then encoded using variable-length encoding naturally reached the further compressed, in this paper, variable-length encoding is Huffman encoding.

Using the output bit stream of above example to introduce the new encoding method process.

1) First, divide the binary output stream every 3 bits as a group: 111 000 111 000 100 000 010 101 100 00. In this
Process, there will be remain 0, 1, 2 bits can not participate. So, in order to unity, in the head of the output bit stream of Huffman encoding cost two bits to record the number of bits that do not participate in group and those remainder bits direct output in end. Figure 2 is shown the output bit stream structure of Huffman encoding.

Number of remain bits	bits stream	remain bits

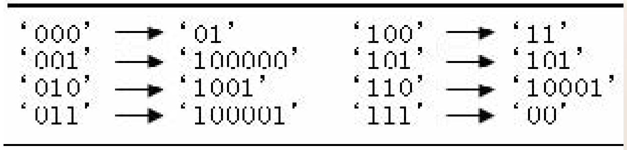
Figure 2: The bit stream structure of Huffman encoding

2) The emergence of statistical probability of each symbol grouping results are as follows:

P('000')= 0.3333 P('001')= 0
P('010')= 0.1111 P('011')= 0
P('100')= 0.2222 P('101')= 0.1111
P('110')= 0 P('111')= 0.2222

3) According to the probability of the above results, using Huffman encoding, then obtain code word book, as follow table1.

Table 1: Code word comparison table



'000'	→	'01'	'100'	→	'11'
'001'	→	'100000'	'101'	→	'101'
'010'	→	'1001'	'110'	→	'10001'
'011'	→	'100001'	'111'	→	'00'

Through the above code book we can get the corresponding output stream: 10 00 01 00 01 11 01 1001 101 11 00, a total of 25 bits, the '10' in the head is binary of remainder bits' number. The last two bits '00' are the result of directly outputting remainder bits. Compared with the original bit stream save four bits.

Decoding is inverse process of the above-mentioned process.

III. RESULTS

The experimental results show the standard Lena image 128×128 grayscale image compression with different comparison parameters. Average code length which is calculated as follows:

$$\bar{\mathbf{L}} = \sum_{\mathbf{i=0}}^{\mathbf{8}} \mathbf{p(i)\ L_i}$$

Where p is the probability of symbols appeared, L_i is the length of word code.

The Performance of this algorithm is verified with respect to the existing algorithms like Discrete Cosine Transform and Discrete Wavelet Transform. For this performance analysis we have considered three parameters (Mean Square Error, Peak Signal to Noise Ratio, Compression Ratio). Mean square error, PSNR and compression ratio are calculated as follows.

$$\text{MSE} = \sigma_q^2 = \frac{1}{N}\sum_{j,k}(f[j,k]-g[j,k])^2$$

$$\text{PSNR} = 10\log_{10}\left(\frac{255^2}{MSE}\right)$$

CR = (Number of bits in the original image) / (Number of bits in the compressed image)

By using the above formulae in the proposed algorithm the following parameters are calculated for the Lena image and given in the following table.

Table 2: Result analysis for Lena image of size 128×128

Algorithm	Parameters			
	CR	MSE	PSNR	Compressed image size
DCT	2.4059	3.4401	36.1236	6810
DWT	1.4080	0.7479	48.1308	11636
SPIHT	1.3507	0.0874	58.7156	12130

IV. CONCLUSION

The proposed lossy image compression algorithm is simple and effective method for gray-scale image compression and is combined with Huffman encoding for further compression in this paper that saves a lot of bits in the image data transmission. There are very wide range of practical value for today that have a have number of image data is to be transmitted.

V. FUTURE SCOPE OF THE WORK

In future this work may extend for the color image and video compression. With existing better algorithms and techniques for image compression.

REFERENCES

[1] Ahmed,N.; Natarajan,T.; Rao,K.R.“ Discrete Cosine Transform,” IEEE Trans. on Computers, vol. C-32, pp. 90-93, Jan. 1974.

[2] Liu,C.P.; Poularikas,A.D.” A New Subband Coding Technique Using (JPEG) DCT for Image Compression,” IEEE Trans. on Image processing, pp.317-321,1996.

[3] Antonini,M.; Barlaud,M.; Mathieu, P.; Daubechies,I.“ Image Coding Using Wavelet Transform, ” IEEE Trans. on Image Processing, Vol. 1, No. 2, pp.205-220.1992.

[4] Ronald A.DeVore; Bjorn Jawerth; Bradley J. Lucier " Image Compression Through Wavelet Transform Coding, " IEEE Trans. on Information Theory, Vol.38.NO.2,pp.719-746, MARCH 1992.

[5] Lewis,A.S.; Knowles,G. " Image Compression Using the 2-D Wavelet Transform, " IEEE Trans. on Image Processing, Vol. I . NO. 2, PP. 244 - 250, APRIL 1992.

[6] Shapiro,J.M. “ Embedded image coding using zerotree of wavelets coefficients,” IEEE Trans. on Signal Processing,pp.3445-3462, 1993.

[7] Said,A.; Pearlman,W.A. “ 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. 243-250,1996.

[8] Wei Li, Zhen Peng Pang and Zhi Jie Liu,” SPIHT algorithm combined with Huffman encoding,” 3rd International Symposium on Intelligent Information Technology and Security Informatics, IEEE Trans. Page No 341- 343,2010.