RESEARCH ARTICLE OPEN ACCESS

Effective Compression in Digital Images with SPIHT Algorithm and Entropy

G. Sreenivasulu, B. Srinivasa Rao
Assistant Professor Sree Vidyanikethan Engg.College (Autonomous) Tirupati, Andhra Pradesh
Assistant Professor Sree Vidyanikethan Engg.College (Autonomous) Tirupati, Andhra Pradesh.

Abstract
Image compression is achieved by removing data redundancy while preserving information content. 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 Entropy for further compression that saves a lot of bits in the image data transmission.
Keywords — *Encoding; DWT; SPIHT; Huffman*

I. Introduction

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 code for a (stationary) source approaches optimality in the limit of infinite computation (source length). Notwithstanding he 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. Following that significant work, we developed an alternative exposition of the underlying principles of the EZW technique.

EZW stands for 'Embedded Zero tree Wavelet', which is abbreviated from the title of Jerome Shapiro's 1993article [4],"Embedded Image Coding Using Zero trees of Wavelet Coefficients". EZW is a simple and effective image compression algorithm, its output bit-stream ordered by importance. Encoding was able to end at any location, so it allowed achieving accurate rate or distortion. This algorithm does not need to train and require pre-stored codebook. In a word, it does not require any prior knowledge of original image.

More improvements over EZW are achieved by SPIHT, 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 image through the wavelet transform, the generally small [6], so it will appear seriate "0" situation in quantify. 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

Set partitioning in hierarchical trees (SPIHT) is an image compression algorithm that exploits the inherent similarities across the subband in a wavelet decomposition of an image. The algorithm codes the most important wavelet transform coefficients first, and transmits the bits so that an increasingly refined copy of the original image can be obtained progressively.

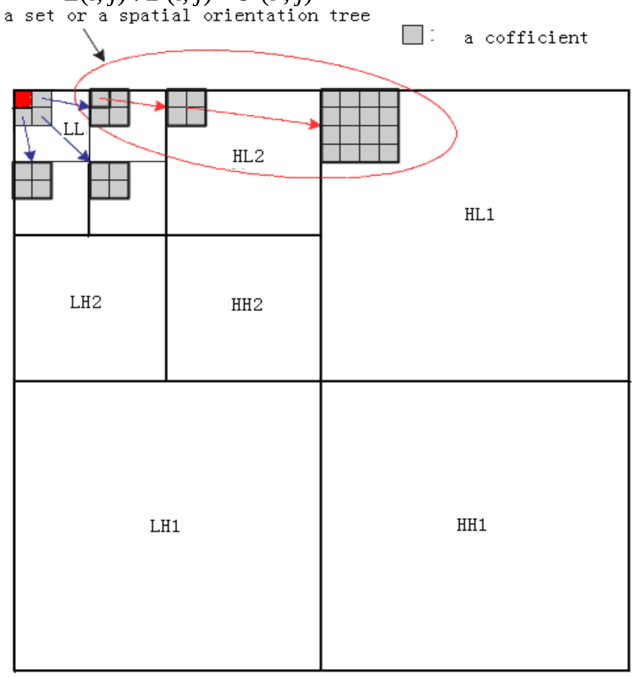
The powerful wavelet-based image compression method called *Set Partitioning in Hierarchical Trees* (SPIHT), The SPIHT method is not a simple extension of traditional methods for image compression, and represents an important advance in the field. Image data through the wavelet decomposition, the coefficient 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 coefficients has four children except the 'red' marked coefficients in the *LL* subband and the coefficients in the highest subband (*HL*1;*LH*1; *HH*1).

The following set of coordinates of coefficients is used to represent set partitioning method in SPIHT algorithm. The location of coefficient is notated by *(i, j)*, where *i* and *i*ndicate row and column indices, respectively

H: Roots of the all spatial orientation trees

$O(i, j)$:Set of offspring of the coefficient (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 coefficient (i, j), $L(i, j) : D(i, j) - O(i, j)$



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

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

where $c_{i,j}$ is the wavelet coefficients.

In the 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 coefficient if the set partitioning algorithm is applied to a wavelet transf - ormed image.

Algorithm: SPIHT
1) Initialization:

1. output $n = \lfloor \log_2 \max\{|c(i,j)|\} \rfloor$;
2. set $LSP = \varphi$;
3. set $LIP = (i, j) \in H$;
4. set $LIS = (i, j) \in H$, where $D(i; j) \neq \varphi$ and set each

Entry in *LIS* as type A ;

2) Sorting Pass:
1. for each$(i, j) \in LIP$ 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 $(i, j) \in$ LIS do:
   (a) If (i, j) is type A then i. output $S_n(D(i, j))$
   ii. If $S_n(D(i; j)) = 1$ then
   A. for each $(k, l) \in O(i, j)$
   output $Sn(k; l)$
   f $S_n(k, l) = 1$ then append (k, l) to *LSP*, output
   sign $(c_{k,l})$,and $c_{k,l} = c_{k,l} - 2^n \cdot \text{sign}(c_{k,l})$
   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 $(k, l) \in O(i, j)$ 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. go to step 2)

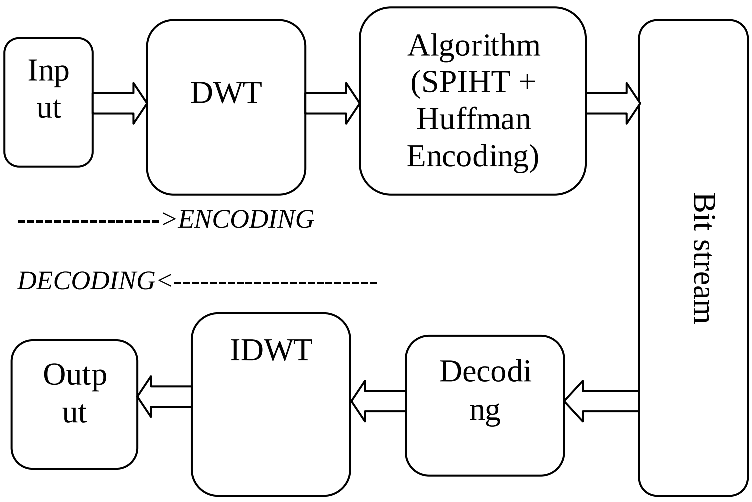


Fig1: block diagram of compression and decompression process

B. Analyse of SPIHT algorithm

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

26	6	13	10	7	13	-12	7
-7	7	6	4	3	4	6	-1
4	-4	4	-3	5	-7	3	9
2	-2	-2	0	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

$n = [\log_2 max\{|c(i,j)|\}] = 4$, so, the initial threshold Value: $T_0=2^4$, for T_0, the output binary stream: 111000000000111100111011111111100011101110100,

Total 42 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: 110 000 000 001 111 001 110 111 111 111 000 111 011 101 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.

Figure2 is shown the output bit stream structure of Huffman encoding.

Number of remain bits	Bits stream	Remain Bits

Figure2: the bit stream structure of Huffman encoding

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

P (‘000’) = 0.2144 P (‘001’) =0.14
P (‘010’) = 0 P (‘011’) =0.07
P (‘100’) = 0 P (‘101’) = 0.07
P (‘110’) = 0.14 P (‘111’) = 0.35

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’→’00’	‘100’→’011’
‘001’→’01’	‘101’→’0011’
‘010’→’11’	‘110’→’01111’
‘011’→’010’	‘111’→’111111’

Through the above code book we can get the corresponding output stream: 00 01 11 010 011 0011 01111 111111, a total of 27 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 15 bits.

Decoding is inverse process of the above-mentioned process.

III. ANALYSIS OF EXPERIMENTAL RESULTS

In order to verify the validity of this algorithm, images usually using all are analyzed, we use 5-level pyramids constructed with the 9/7-tap filters.Table2 is shown the experiment results of two standard 512×512 grayscale image Lena, Goldhill at different rate. Average code length which is calculated as follows:

$$\overline{L} = \sum_{i=1}^{8} p(i)L_i$$

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

From the experimental results, we can see that values of $\overline{L}$ are less than 3, so we can achieve the compression effect. For each image in the same rate always the probability of each symbol appear flat, and only small fluctuations, so saving the number of bits are also pretty much the same thing. With the rate increase word code length in average ($\overline{L}$) will be an increasing trend, but after the rate greater than 0.3bpp the trending will be become very slowly, and more value of rate more bits will be save.

Image	Lena			Goldhill		
Rate (bpp)	0.1	0.3	0.5	0.1	0.3	0.5
P('000')	0.2636	0.2362	0.2352	0.2521	0.2457	0.2205
P('001')	0.1364	0.1393	0.1383	0.1468	0.1441	0.1454
P('010')	0.1201	0.1229	0.1212	0.1271	0.1276	0.1282
P('011')	0.1001	0.1022	0.1028	0.0974	0.0974	0.1008
P('100')	0.1362	0.1397	0.1407	0.1403	0.1419	0.1461
P('101')	0.0759	0.0840	0.0832	0.0746	0.0767	0.0816
P('110')	0.0961	0.1021	0.1024	0.0938	0.0954	0.0992
P('111')	0.0716	0.0738	0.0762	0.0680	0.0713	0.0782
average code length	2.8839	2.9216	2.9242	2.8905	2.9023	2.9393
the number of output bits by SPIHT encoding	26225	78650	131080	26225	78647	131080
the number of output bits by the new algorithm encoding	25430	76802	127986	25478	76287	128642
the number of saving bits	795	1848	3094	747	2360	2438

IV. CONCLUSION

Proposing a simple and effective method 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 ranges of practical value for today that have a large number of image data to be transmitted.

REFERENCES

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

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

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

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

[5] J. M. SHAPIRO. Embedded image coding using zerotree of wavelets coefficients [J]. IEEE Trans. Signal Processing 1993 41(12) 3445-3462.

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

[7] FAN Qi-bin. Wavelet analysis. Wuhan: Wuhan University Press, 2008.J. Breckling, Ed., *the Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.