A Simple and Fast Algorithm for Image Coding Based on SPIHT

Modified SPIHT

Yiqun Zhang[1,2], Peijie Lin[1,2], Lijun Wu[1,2] and Shuying Cheng[1,2,*]
[1]College of Physics and Information Engineering, and Institute of Micro-Nano Devices and Solar Cells, Fuzhou University, Fuzhou, 350116 - China
[2]Jiangsu Collaborative Innovation Center of Photovolatic Science and Engineering, Changzhou, 213164 - China
*Corresponding author

Abstract—**The complicate calculation and frequent comparison based on spatial orientation tree (SOT) structure in SPIHT severely reduce the coding speed. To overcome these shortcomings, a modified image compression algorithm is proposed without SOT. The algorithm sorts the wavelet coefficients in descending order to eliminate the correlation among wavelet coefficients and obtain a one-dimensional coordinate list corresponding to the sorted wavelet coefficients. This algorithm stops the sorting pass immediately when a wavelet coefficient from the sorted coordinate list is not significant in a bit plane, which simplifies the process of scanning. Experimental results show that the proposed algorithm drastically improves the coding speed.**

Keywords-image compression; set partitioning in hierarchical tree; image coding; wavelet transform; zero tree coding

I. INTRODUCTION

Image compression is important in order to effectively store and transmit digital image. It has been proven that image compression algorithms based on the discrete wavelet transform (DWT) can provide high coding efficiency for natural images [1-4].

J.M.Shapiro [5] proposed an embedded zero tree wavelet (EZW) encoding that is a typical representative of the wavelet encoding method. Said and Pearlman [6] proposed an elaborate set partitioning in hierarchical trees (SPIHT) algorithm to improves the compression performance. SPIHT can be very useful for applications where the user can quickly inspect the image and decide if it should be really downloaded. Therefore, since a few years ago, there has been a growing interest in the SPIHT [4,7-10]. Even when SPIHT is not a standard, it is used in many video applications that use image compression with the embedded coding property [11-12] and medical imaging applications with SPIHT as a core [10,13-15].

However, One of the main drawbacks of SPIHT is the slow coding speed owing to the dynamic processing order that depends on the image contents [16]. In recent years, efforts to improve the performance have been mostly concentrated in modifying the lists used to store the sets coordinates. For example, Jagadeesh et al. [17] utilizes a pre-processing scheme, by 2-D arrangement for wavelet coefficients. Then, they rearrange the order of the encoded output bit stream by memory optimization at the algorithmic three continuously growing linked lists levels and a low bit rate image coder. Jin et al. [18] decomposes a wavelet-transformed image into 4×4-bit blocks and simultaneously encodes all the 4×4 bits in a bit-plane. Both an encoder and a decoder are implemented in the hardware. Xie et al. [19] scans the initial node subsequent N generation in the round, whereas it only scans the first component within the N generation offspring. This balances the performance of a code and improves the efficiency of coding. Domínguez et al. [20] combines the SPIHT lists and hierarchical subband scanning to save comparison operations during sorting passes. The sets inside a subband are scanned according to the magnitude of its subband threshold. The scanning of a SOT stops if the threshold of a subband containing descendants is less than the current threshold, reducing the number of comparisons and the time taken by the encoder per sorting pass.

All of these algorithms based on the SOT have improved the original algorithm in some extent. However, due to the complexity of the SOT, the coding speed is still limited. This work focuses on presenting a fast image compression method to meet real-time image transmission so that the SOT structure in SPIHT is not adopted. The adjustment of wavelet coefficients ordering process in this algorithm can simplify SPIHT to boost the coding speed.

II. MODIFIED SPIHT

In SPIHT, three lists are used to improve efficiency in SPIHT: LIP (list of insignificant pixel), LSP (list of significant pixel), and LIS (list of insignificant set) where each element contains four adjacent pixels. These lists make SPIHT computing complexity and frequently compared. The modified algorithm can significantly reduce the computational complexity and the comparison times.

A. Algorithm Depiction

(1) Sets and symbols definitions are as follows:

W: the coefficients matrix of an image after wavelet decomposition.

Copyright © 2016, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

LCDC: the list of coordinates corresponding to wavelet coefficients sorted in descending order about their absolute value in matrix W.

LSCC: the list of significant coefficients' coordinates obtained during the significance test.

NSC: the number of significant coefficients in the current bit plane.

LNSC: the list of the number of significant coefficients encoded before the current bit plane.

LOBS: list of output bit stream completed by encoding.

n_{max}: maximum bit-plane level required for coding, the initial n_{max} is calculated as:

$$n_{max} = \lfloor \log_2(|\mathrm{W(LDPC(1))}|) \rfloor \quad (1)$$

(2) Two major modifications are proposed:

1) In allusion to the frequently add and delete nodes to dynamic lists, using LCDC to avoid. During the initialization, obtain LCDC. While the memory space of coefficients sorted in descending order can be released, it is no longer used in the next algorithm. Because 1 is the minimum threshold in the significance test, the coordinates, corresponding to the wavelet coefficients with less than 1, are not included in LCDC to save the memory space. To perform the significance test, it only discriminates by order whether the wavelet coefficient is significant or not. Therefore, this eliminates the discrimination for its descendants' significance, which tremendously simplifies the process of scanning.

2) In terms of redundant encoding of large insignificant coefficients, in each bit plane, discriminating by order whether the coordinate in LCDC corresponding to the wavelet coefficient in matrix W is significant or not. If the wavelet coefficient is significant, then add the coordinate to LSCC, and remove it from LCDC. If not significant, then stop the significant test in current bit plane. As a result, the modified algorithm only needs to encode the significant wavelet coefficients without any redundant encoding.

However, LNSC and LSCC need to be transmitted with LOBS for decoding which reduces the peak signal to noise ratio (PSNR) of an image for a given bit rate.

B. Encoding Steps

Step 1. Initialization: set $n = n_{max}$. Get LCDC. Set $\mathrm{LSCC} = \{\phi\}$. Set $\mathrm{LNSC} = \{\phi\}$. Set NSC = 0.

Step 2. Significance test (sorting pass) in the n-th bit plane: Set $i = 1$.

If $|\mathrm{W(LCDC}(i))| \geq 2^n$, output a bit "1" and output the sign of $\mathrm{W(LCDC}(i))$, then set NSC = NSC + 1, update LSCC = [LSCC, LCDC(i)], set $i = i + 1$ and go to Step 2.

Else update LNSC = [LNSC, NSC], set $i = 1$, then remove the first NSC elements in LCDC and go to Step 3.

Step 3. Refinement pass:

If $n = n_{max}$, then go to Step 4.

Else if $1 \leq i \leq (sum(\mathrm{LNSC}) - \mathrm{NSC})$, for the elements in LSCC, output the n-th most significant bit of $|\mathrm{W(LCDC}(i))|$, then set $i = i + 1$ and go to Step 3.

Else go to Step 4.

Step 4. Quantization-step update: decrement n by 1, set NSC = 0 and go to Step 2 until the bits of LOBS reach the specified number.

C. Decoding

Set a zero matrix with the size of W. Then, restructure the significant coefficients according to LNSC and LSCC using the following equation:

$$value = \begin{cases} 2^n + 2^{n\text{-}1}, & \text{sign bit} = 1 \\ -2^n - 2^{n\text{-}1}, & \text{sign bit} = 0 \end{cases} \quad (2)$$

where n represents the bit plane, value is the value of a wavelet coefficient.

The refinement pass method difference between this algorithm and the original SPIHT is that it needs to be based on the LNSC and LSCC.

This paper uses an image block of 8×8 size as an example to briefly introduce the improved encoding method. Assuming that the image block is decomposed by wavelet transform to a wavelet coefficients matrix shown in Figure 1.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	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

FIGURE I. THREE-LEVEL WAVELET DECOMPOSITION COEFFICIENTS

Initialize LCDC = [1, 17, 29, 9, 2, 10, 3, ⋯] according to Figure I and obtain n = 5. In this bit plane, the first four coordinates meet the significance test in LCDC. Therefore, output LOBS = 11111110, and update LCDC = [2, 10, 3, ⋯], LSCC = [1, 17, 29, 9], LNSC = [4]. Then decrement n by 1. In this bit plane, the first two coordinates meet the significance test in LCDC. Update LCDC = [3, ⋯], LSCC = [1, 17, 29, 9, 2, 10], LNSC = [4, 2], output of bit stream in this significance test is "1011". And four significant coefficients (63, 49, 47, 34) have refinement pass, the output of refinement bit stream is "1100". Therefore in this bit plane (n = 4), update LOBS =

1111111010111100. The following significance test and refinement pass are deduced as above until the bits of LOBS reach the specified number to stop encoding.

III. Experimental Results and Analysis

The performance of the proposed modified SPIHT is examined and compared with SPIHT and Domínguez et al. [20] (with good coding speed, the source code is provided by the author) methods using the images of Lena, Peppers, and Goldhill at monochrome 512 × 512 size. In the experiment, each image is decomposed by using CDF 9/7 filter bank with 6 level (L = 6). The bit rate interval tested was from 0.01 to 4 bpp (bit per pixel). The methods used in this paper were implemented in R2014a Matlab under the windows7 operating system, running on the Lenovo personal computer, with an Intel Core i5 clocked at 2.70 GHz and 4 GB of memory. And there is no any optimization for special hardware. The time performance and PSNR were compared. The execution time was taken including the initialization stage up to the end of the encoding process.

A. Execution Time

In SPIHT algorithm, if a coefficient is insignificant and it will be compared with the next current thresholds in further sorting passes until it is found significant or the encoding process ends. Domínguez at al. [20] combines the SPIHT lists and hierarchical subband scanning to save comparison operations during sorting passes, which drastically reducing the time taken by the encoder per sorting pass. However, this algorithm is based on complex SOT structure which must discriminate if a set contains significant pixels, consuming time. The improved algorithm in this paper only encodes the significant wavelet coefficients and eliminates the discrimination for its descendants' significance, saving time greatly.

TABLE I. EXECUTION TIME COMPARISON (IN S)

Bit rate (bpp)	Image	Modified SPIHT	SPIHT	Domínguez at al.[20]
0.01	Lena	**0.0156**	0.1404	0.1248
	Peppers	**0.0156**	0.1716	0.1092
	Goldhill	**0.0156**	0.1560	0.1546
0.1	Lena	**0.0312**	0.8112	0.3120
	Peppers	**0.0312**	0.8736	0.3744
	Goldhill	**0.0312**	0.8580	0.3588
0.25	Lena	**0.0468**	1.8564	0.7488
	Peppers	**0.0468**	1.8720	0.8112
	Goldhill	**0.0468**	2.3556	0.8424
0.5	Lena	**0.0780**	4.2120	1.2480
	Peppers	**0.0780**	4.4928	1.4196
	Goldhill	**0.0780**	5.2884	1.5132
0.75	Lena	**0.1248**	9.0325	2.1372
	Peppers	**0.1248**	9.6877	2.2464
	Goldhill	**0.1092**	9.5161	1.8252
1	Lena	**0.1404**	13.5409	2.3556
	Peppers	**0.1404**	18.4549	3.0576
	Goldhill	**0.1404**	16.8169	2.5584
2	Lena	**0.2808**	74.8961	4.2744
	Peppers	**0.3120**	87.2826	4.7268
	Goldhill	**0.2964**	67.7824	4.1184
4	Lena	**0.5772**	241.9108	5.7564
	Peppers	**0.5148**	295.0135	5.7876
	Goldhill	**0.4212**	242.5036	5.7720

Note: **Black** fonts indicate the best performance.

Table 1 shows the numerical results of the execution time of the modified SPIHT, compared with SPIHT and Domínguez et al. [20] methods for the Lena, Peppers, and Goldhill images. The time performance of the modified SPIHT at each bit rate listed in table 1 is the most superior.

Figure II shows the plots of the time ratios Domínguez et al./modified SPIHT at bit rates of 0.01, 0.1, 0.25, 0.5, 0.75, 1, 2 and 4 bpp for the Lena, Peppers, and Goldhill images. For the Lena, the improved performance is optimal at 0.75 bpp. For the Peppers, the improved performance is optimal at 1 bpp. For the Goldhill, the improved performance is optimal at 0.5 bpp.

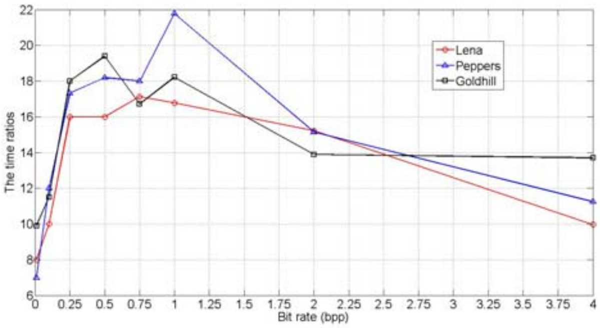

FIGURE II. THE EXECUTION TIME RATIOS DOMÍNGUEZ ET AL/MODIFIED SPIHT

B. PSNR Results

The distortion between the original and recovered images was measured by using the PSNR formula:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{255^2}{\mathrm{MSE}}\right)\mathrm{dB} \qquad (3)$$

where MSE is the mean squared error between the original and the reconstructed image.

Table 2 shows the comparison of PSNRs. The modified SPIHT yields lowest PSNR for each bit rate because LNSC and LSCC take a number of bits for a given bit rate. It can be observed that the PSNR of the modified SPIHT for the Lena image at a bit rate of 0.25 bpp has 30.27 dB that is between 29.82 dB at 0.1 bpp and 33.65 dB at 0.25 bpp in Domínguez at al. method. The PSNR of the modified SPIHT for the Lena image at the bit rate of 0.5 bpp has 32.74 dB that is between 29.82 dB at 0.1bpp and 33.65 dB at 0.25bpp in Domínguez at al. method. The PSNR of the modified SPIHT for the Lena image at the bit rate of 0.75 bpp has 34.62 dB that is between 33.65 dB at 0.25 bpp and 36.77 dB at 0.5 bpp in Domínguez at al. method. The modified SPIHT for the Lena image at the bit rate of 0.75 bpp has the execution time of 0.1248s, which is equal to one-sixth of Domínguez et al. at bit rates of 0.25 bpp with 0.7488s.

Furthermore, the PSNRs of Modified SPIHT for the images of Lena, Peppers, and Goldhill at 0.25 bpp are similar to those of SPIHT at 0.1 bpp, which means the sacrifice of 2.5 times compression space. And the PSNRs of Modified SPIHT for the images of Lena, Peppers, and Goldhill at 0.5 bpp are similar to those of SPIHT at 0.25 bpp, which means the sacrifice of 2 times compression space. Under the circumstances, table 3 shows the time ratio modified SPIHT/SPIHT. The coding speed is greatly improved. Therefore, the modified SPIHT boosts coding speed at the sacrifice of the compression space or image quality under conditions permitting.

TABLE II. COMPARISON OF PEAK SIGNAL-TO-NOISE RATIO (IN DB)

Bit rate (bpp)	Image	Modified SPIHT	SPIHT	Domínguez at al.[20]
0.01	Lena	20.95	22.33	**22.44**
	Peppers	19.41	20.88	**21.07**
	Goldhill	22.32	22.32	**22.95**
0.1	Lena	27.15	29.32	**29.82**
	Peppers	25.93	28.65	**28.99**
	Goldhill	26.36	27.11	**27.64**
0.25	Lena	30.27	33.32	**33.65**
	Peppers	29.42	32.79	**32.91**
	Goldhill	28.38	29.77	**30.20**
0.5	Lena	32.74	36.59	**36.77**
	Peppers	32.35	35.44	**35.46**
	Goldhill	30.08	32.35	**32.57**
0.75	Lena	34.62	38.30	**38.34**
	Peppers	34.14	**36.64**	36.63
	Goldhill	31.30	34.33	**34.49**
1	Lena	35.71	39.85	**39.88**
	Peppers	35.04	**37.81**	37.72
	Goldhill	32.19	35.73	**35.85**
2	Lena	38.77	44.05	**44.11**
	Peppers	37.54	**42.12**	41.89
	Goldhill	34.84	**40.84**	40.72
4	Lena	42.17	**55.07**	52.29
	Peppers	40.50	**53.44**	50.78
	Goldhill	37.88	**51.68**	50.07

Note: **Black** fonts indicate the best performance.

TABLE III. COMPARISON OF THE RELATIONSHIP BETWEEN TIME AND SPACE WITH SIMILAR PSNR

Space ratio	Image	Time ratio
2.5	Lena	0.0577
	Peppers	0.0535
	Goldhill	0.0545
2	Lena	0.0420
	Peppers	0.0417
	Goldhill	0.0331

IV. CONCLUSIONS

In this paper, a simplified and fast SPIHT algorithm that abandons SOT is proposed to boost the coding speed for image compression. This algorithm is easily implemented in the software and hardware. Standard images like Lena, Barbara and Mandrill are taken for experimentation. It is observed from the experimental results that the proposed method without SOT outperforms the SOT-based image compression methods in terms of execution time, but reducing the image quality. However, the more sparse an image, the higher quality a restored image. It should be balanced between coding speed and image quality. Therefore, this work focuses on proposing a simplified and fast image compression method to meet real-time image transmission with less demanding about image quality.

ACKNOWLEDGMENT

This research is supported by the grant (No.JK2014003) from the Educational Department of Fujian Province, China, the grant (No. 2015H0021) from Fujian Provincial Department of Science & Technology, China, and the grant (No.51508105) from National Natural Science Foundation of China.

REFERENCES

[1] Grgic, S., M. Grgic, B. Zovko-Cihlar. “Performance analysis of image compression using wavelets,” IEEE Transactions on Industrial Electronics, vol. 48, no. 3, pp. 682-695, 2001.

[2] Yang, S., R. Wu, M. Wang, et al. “Evolutionary clustering based vector quantization and SPIHT coding for image compression,” Pattern Recognition Letters, vol. 31, no. 13, pp. 1773-1780, 2010.

[3] Priya, R.A., S. PNarote, A. Patil. “Dual Tree Wavelet Transforms in Image Compression,” IJST, vol. 3, no. 1, pp. 83-95, 2011.

[4] Lee, T.-Y.H.-W. Shen. “Efficient local statistical analysis via integral histograms with discrete wavelet transform,” IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 12, pp. 2693-2702, 2013.

[5] Shapiro, J.M. “Embedded image coding using zerotrees of wavelet coefficients,” IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3445-3462, 1993.

[6] Said, A.W. 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, pp. 243-250, 1996.

[7] Christophe, E., C. Mailhes, P. Duhamel. “Hyperspectral image compression: adapting SPIHT and EZW to anisotropic 3-D wavelet coding,” IEEE Transactions on Image Processing, vol. 17, no. 12, pp. 2334-2346, 2008.

[8] Manikandan, M.S.S. Dandapat. “Effective quality-controlled SPIHT-based ECG coding strategy under noise environments,” Electronics Letters, vol. 44, no. 20, pp. 1182-1184, 2008.

[9] Ma, J., J. Fei, D. Chen. “Rate-distortion weighted SPIHT algorithm for interferometer data processing,” Journal of Systems Engineering and Electronics, vol. 22, no. 4, pp. 547-556, 2011.

[10] Cavero, E., A. Alesanco, L. Castro, et al. “SPIHT-based echocardiogram compression: clinical evaluation and recommendations of use,” IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 1, pp. 103-112, 2013.

[11] Zhu, J.R.M. Dansereau. “Error-resilient and error concealment 3-D SPIHT for multiple description video coding with added redundancy,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 6, pp. 855-868, 2012.

[12] Xiang, T., J. Qu, D. Xiao. “Joint SPIHT compression and selective encryption,” Applied Soft Computing, vol. 21, pp. 159-170, 2014.

[13] Higgins, G., B. McGinley, N. Walsh, et al. “Lossy compression of EEG signals using SPIHT,” Electronics letters, vol. 47, no. 18, pp. 1017-1018, 2011.

[14] Rubio, Ó.J., Á. Alesanco, J. García. “Secure information embedding into 1D biomedical signals based on SPIHT,” Journal of biomedical informatics, vol. 46, no. 4, pp. 653-664, 2013.

[15] Isa, S.M., A. Noviyanto, W. Jatmiko, et al. “The effect of electrocardiogram signal compression using beat reordering and SPIHT on automatic sleep stage classification,” Procedia Engineering, vol. 41, pp. 888-896.

[16] Zhang, N., L. xu Jin, H. jiang Tao, et al. “Image compression algorithm of high-speed SPIHT for aerial applications,” in Proc. of IEEE Conf. on Communication Software and Networks, pp.536-540, May 27-29, 2011.

[17] Jagadeesh, S.E. Nagabhooshanam. “Linear adaptive global node-tree filters based SPIHT MR image codec,” in Proc. of IEEE Conf. on Electrical, Computer and Communication Technologies, pp.1-8, March 5-7, 2015.

[18] Jin, Y.H.-J. Lee. “A block-based pass-parallel SPIHT algorithm,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, no. 7, pp. 1064-1075, 2012.

[19] Xie, K., Z. Bai, W. Yu. "Fast seismic data compression based on high-efficiency SPIHT," Electronics Letters, vol. 50, no. 5, pp. 365-367, 2014.

[20] Domínguez, H.d.J.O., O.O.V. Villegas, V.G.C. Sanchez. "Modified set partitioning in hierarchical trees algorithm based on hierarchical subbands," Journal of Electronic Imaging, vol. 24, no. 3, pp. 1-12, 2015