

以小波轉換為基礎，利用零樹及碎形編碼技術作影像壓縮之研究

A Wavelet-Based Zero Tree and Fractal Coding Approach to Image Compression

計畫編號：NSC87-2213-E-009-043

執行期間：86年8月1日 --- 87年7月31日

主持人：吳炳飛

執行機構及單位：國立交通大學 電機與控制工程學系

摘要：

As an extension of the embedded zerotree wavelet (EZW) algorithm, the Set Partitioning In Hierarchical Trees' (SPIHT) algorithm is an rapid and an efficient means of compressing an image. However, the original SPIHT uses three ordered lists to store the significant information during coding, which requires a significant amount of memory and leads to large cost on the hardware implementation. In this study, we present another implementation of SPIHT. Saving as much memory as possible is of the priority concern. In addition to the recursive programming, this novel implementation uses top bits of transformed coefficients to store a significant information. In the novel implementation, the memory required for the three lists can be discarded entirely. Experimental results show that the proposed method can save memory at least 300KB at bit rate 1 bits per pixel and preserve all merits of SPIHT including the property of embedded coding.

計畫緣由與目的：

To save transmission time or storage space of an image, the image compression technique is extensively applied to transmit or store an image. Among various compression techniques, zerotree coding has received the most interest recently owing to that it is computationally simple and quite effective on compression. In addition, its embedded coding property facilitates

progressive transmission.

Shapiro(1993) proposed the original zerotree algorithm, called embedded zerotree wavelet (EZW). The algorithm fully exploits the self-similarity among the wavelet transform coefficients located on the similar spatial orientation, but at different scales. That same investigation demonstrated that its performance at the peak signal-to-noise ratio (PSNR) markedly exceeds that of the JPEG standard. Said and Pearlman (1996) further enhanced the performance of EZW by presenting a more efficient and faster implementation called set partitioning in hierarchical tree (SPIHT). The SPIHT is among the best coding algorithms available. The SPIHT partitions a significant tree off and then uses three ordered lists to store the coordinates of the partitioning results. Owing to the excellent performance of EZW and SPIHT, many researchers have developed algorithms based on EZW or SPIHT, with notable examples including Effros (1997), Hontsch et al. (1997), Li and Jin (1997), Wang and Ghanbari (1997), and Rogers and Cosman (1998).

However, these EZW-based coders or SPIHT-based coders require an enormous amount of working memory to store the significant amount of information during coding, subsequently leading to high cost in terms of hardware realization. More specifically, there are two lists in the original EZW: dominant and subordinate. The dominant list contains the coordinates of those coefficients that have not yet been found to be significant. The subordinate list

contains the magnitudes of those coefficients that have been found to be significant. In the original SPIHT, three lists, i.e. insignificant sets (LIS), insignificant pixels (LIP), and significant pixels (LSP), are utilized to store the significant information, locations of partitioning sets and coefficients. These lists normally occupy at least 300K bytes of memory at the bit rate 1 bits per pixel (bpp). In addition, increasing the bit rate increases the demanded memory. Thus, the large number of required working memory can not be neglected, particularly when extending the EZW or SPIHT to the three-dimensional (3-D) video compression.

In this study, we present a novel implementation of SPIHT. The proposed technique used in SPIHT can also be applied to the EZW. By doing so, we can save as much working memory as possible. This novel implementation largely focuses on taking advantage of recursive programming to the repetitive tree structure and using three top bits of the transformed coefficients to store a significant amount of information instead of the lists. In this novel implementation, the three lists for SPIHT can be discarded entirely, leading to a low memory cost zerotree coding. For brevity, our method is referred to herein as low memory zerotree coding (LMZC). In addition, complexity analysis of LMZC is performed as well.

結果與討論：

A typical 3-scale spatial tree depicted in figure 1 is obtained by the multiscale pyramidal decomposition for an image (Mallat 1989, Antonini *et al.* 1992). Table 1 compares the EZW, SPIHT and our LMZC with respect to the required memory and computational complexity.

The results in table 2 are obtained by applying the proposed coder to the three, 512×512 and 8 bpp, test images:

Lena, Barbara and Goldhill. Herein, the 9/7 tap filter (Antonini *et al.* 1992). Symmetric extension is applied to the image edges. The total execution time of the encoder includes both the transform time and the encoding time. The total execution time deemed necessary for the decoder is the sum of the decoding time and the inverse transform time. All the results are obtained from an AMD K6-2 300 MHz CPU, RAM 64 MB personal computer, and the platform Win98.

To elevate the coding results in the PSNR, all the generated bit streams in the coding algorithm are further encoded by using the adaptive arithmetic codes (Witten *et al.* 1987). Table 3 summarizes the results at various bit rates of our coder. Notably, the reported bit rates are calculated from the actual compressed files. In addition, the PSNRs can be expressed as

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

where the MSE is calculated from the original image and the reconstructed image produced by the decoding algorithm. For comparison, the original image of Lena and its reconstructed one at bit rate 0.5 bpp (compression ratio 16:1) are shown in figure 3. According to this figure, these two images appear to have no perceptible difference. Thus, the performance of LMZC is very good.

Tables 2 and 3 confirm that the algorithm proposed herein is as efficient and rapid as other high performance coders. In addition, the algorithm proposed herein saves a lot of memory, thereby making it more appropriate for applications involving required less memory.

計畫成果自評：

本計畫主要研究小波轉換及編碼技術對影像壓縮的成效。本研究中，我們提出一個相當省記憶體空間的零樹編碼法，使得實作上，小波轉換變得非

常可行，研究成果目前正準備整理成論文發表。個人覺得成果相當不錯，可提供有心研究小波轉換應用的學者專家進一步落實理論成果。

參考文獻：

- [1] Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I., 1992, Image coding using wavelet transform. *IEEE Trans. Image Processing*, **1**, pp. 205-220.
- [2] Effros, M., 1997, Zerotree design for image compression: toward weighted universal zerotree coding, *Proceedings of International Conference on Image Processing*, pp. 616-619.
- [3] Hontsch, I., Karam, L. J., and Safranek, R. J., 1997, A perceptually tuned embedded zerotree image coder. *Proceedings of International Conference on Image Processing*, pp. 41-44.
- [4] Li, J. and Jin, J. S., 1997, Structure-related perceptual weighting: a way to improve embedded zerotree wavelet image coding. *IEE Electronics Letters*, **33**, pp. 1305-1306.
- [5] Lewis, A. S. and Knowles, G., 1990, Video compression using 3D wavelet transforms. *Electronics Letters*, **26**, pp. 396-398.
- [6] Lewis, A. S. and Knowles, G., 1992, Image compression using the 2-D wavelet transform. *IEEE Trans. Image Processing*, **1**, pp. 244-250.
- [7] Mallat, S. G., 1989, A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Trans. Pattern Analysis and Machine Intelligenc*, **11**, pp. 674-693.
- [8] Mohd-Yusof, Z. and Fischer, T. R., 1996, An entropy-coded lattice vector quantizer for transform and subband image coding. *IEEE Trans. Image Processing*, **5**, pp. 289-298.
- [9] Rogers, J. K. and Cosman, P. C., 1998, Wavelet zerotree image compression with packetization. *IEEE Signal Processing Letters*, **5**, pp. 105-107.
- [10] Said, A. and Pearlman, W. A., 1996, A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems for Video Technology*, **6**, pp.243-250.
- [11] Shapiro, J. M., 1993, Embedded image coding using zerotrees of wavelets coefficients. *IEEE Trans. Signal Processing*, **41**, pp. 3445-3462.
- [12] Shapiro, J. M., 1996, A fast technique for identifying zerotrees in the EZW algorithm. *ICASSP'96, 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, pp. 1455-1458.
- [13] Wang, Q. and Ghanbari, M., 1997, Scalable coding of very high resolution video using the virtual zerotree. *IEEE Trans. Circuits and Systems for Video Technology*, **7**, pp. 719-727.
- [14] Witten, I. H., Neal, R. M., and Cleary, J. G., 1987, Arithmetic coding for data compression. *Commun. ACM*, **30**, pp. 520-540.
- [15] Xiong, Z., Ramchandran, K., and Orchard, M. T., 1997, Space-frequency quantization for wavelet image coding. *IEEE Trans. Image Processing*, **6**, pp. 677-693.

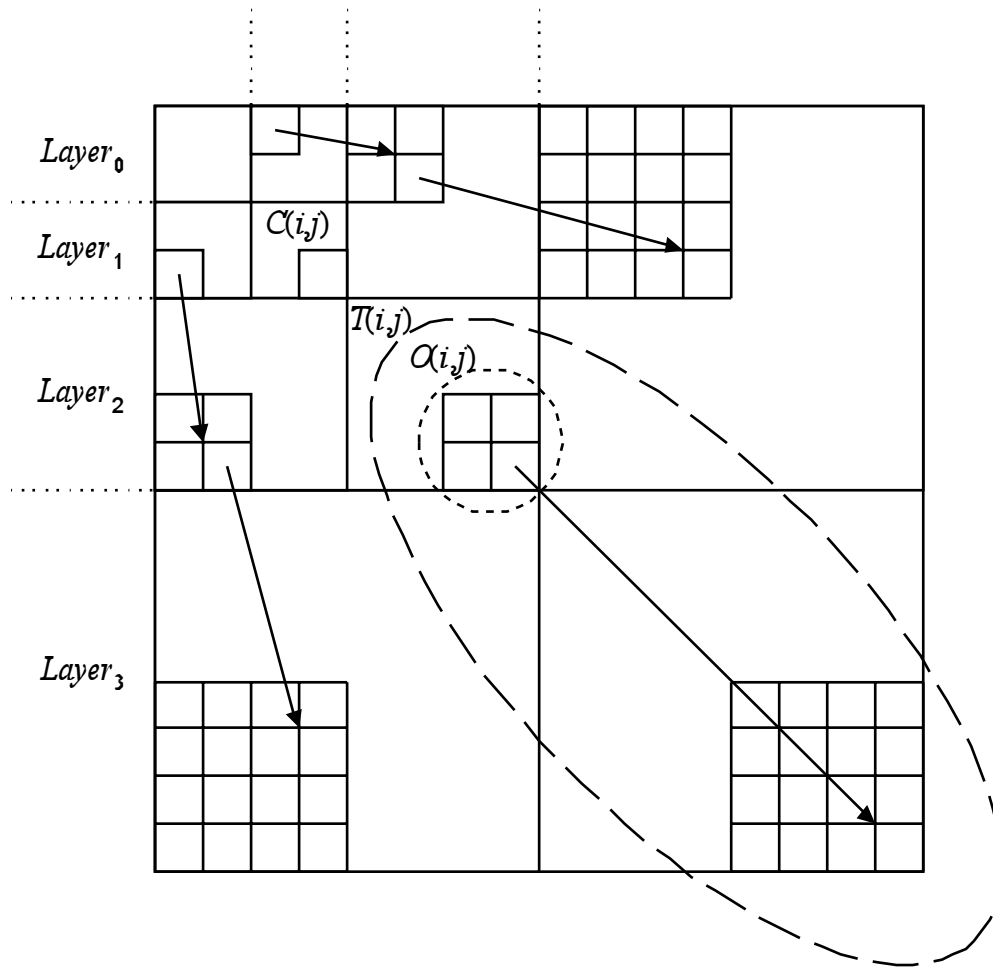


Figure 1. Definition of a 3-scale spatial tree.

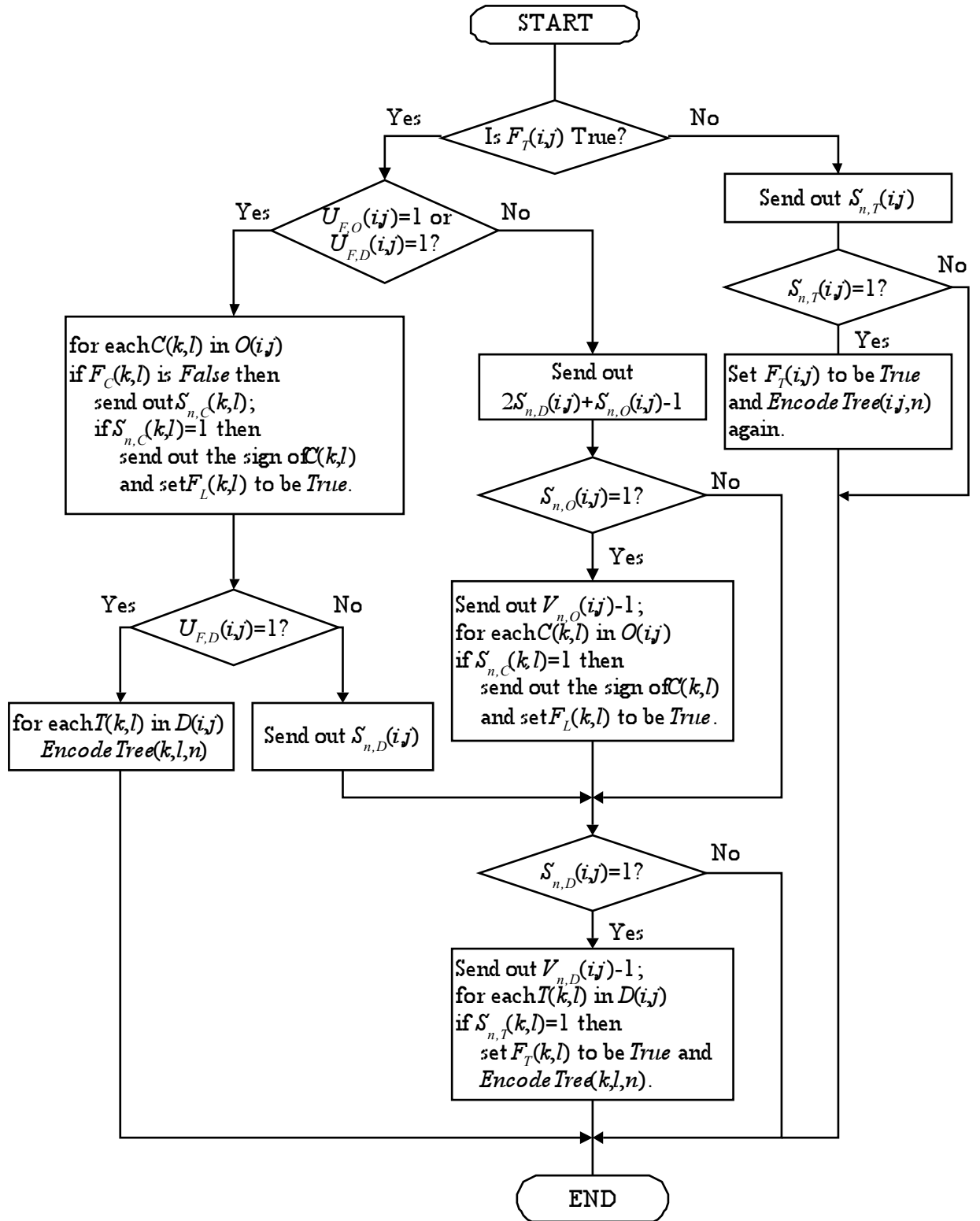


Figure 2. The flowchart of the recursive procedure $EncodeTree(i,j,n)$.



(a)



(b)

Figure 3. Comparison of the original and a reconstructed Lena images. The reconstructed image is decoded at bit rate 0.5 bpp and PSNR is 36.8 dB.

Table 1

Comparisons of the EZW, SPIHT and our coder LMZC on the required memory and the computational complexity.

	EZW	SPIHT	LMZC
Memory required for storing the significant information	at least 300K bytes at bit rate 1 bpp	at least 300K bytes at bit rate 1 bpp	None
The method to store the significant information	Store the coordinates of the coefficients to the lists.	Store the coordinates of the coefficients to the lists.	Bit reversal

Table 2

The results of average execution time of our coder LMZC for Lena, Barbara, and Goldhill 512 by 512 standard images.

Image	Lena			Barbara			Goldhill		
Bit rate (bpp)	0.2	0.5	1	0.2	0.5	1	0.2	0.5	1
Transform time (sec)	0.44								
Inverse transform time (sec)	0.44								
Encoding time (sec)	0.71	0.88	1.1	0.61	0.82	0.99	0.66	0.83	1.4
Decoding time (sec)	0.11	0.17	0.38	0.11	0.16	0.38	0.11	0.16	0.33

Table 3

The coding results in the PSNR (dB) of our coder LMZC for Lena, Barbara and Goldhill 512 by 512 standard images.

Bit rate \ Image	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
Lena	32.88	34.64	35.96	36.8	37.7	38.66	39.09	39.54	39.98
Barbara	26.89	28.97	30.65	31.93	33.29	34.58	35.3	36.06	37
Goldhill	29.68	30.77	31.77	32.99	33.73	34.23	34.81	35.52	36.25