

Performance Evaluation of Wavelet-Based Image Compression Techniques

Nishat Bano, Monauwer Alam and Shish Ahmad

Abstract Wavelet transform provides excellent energy compaction and exploits redundancy for better image compression. This paper analyses the performance of wavelet-based image compression techniques, namely set partitioning in hierarchical trees (SPIHT), set partitioned embedded block (SPECK) and JPEG2000. Image quality measured as peak signal-to-noise ratio and computational complexity, i.e. encoding and decoding times, is calculated for these techniques. Simulation results show that JPEG2000 outperforms other state-of-the-art wavelet image codecs (SPIHT and SPECK) in coding efficiency and computational complexity. Simulation is carried out on various images of different dimensions at different bit rates using MATLAB software.

Keywords JPEG2000 · Set partitioning in hierarchical trees (SPIHT) · Set partitioned embedded block coder (SPECK) · Wavelet transform

1 Introduction

For computer applications before transmission, the redundant data needs to be compressed in order to reduce storage. In image compression, the highly correlated neighbouring pixels are discarded by reducing the redundant information in an image. Image compression techniques are becoming more complex with the improvement in efficiency. Image can be transformed using discrete cosine transform (DCT) [1] and discrete wavelet transform (DWT) [2, 3]. Joint Photographic

N. Bano (✉) · M. Alam

Department of Electronics Engineering, Integral University, Lucknow, UP, India
e-mail: nishatrizvi9@gmail.com

M. Alam

e-mail: malam@iul.ac.in

S. Ahmad

Department of Computer Science, Integral University, Lucknow, UP, India
e-mail: shish_parv@rediffmail.com

Experts Group developed (JPEG) coder [4, 5, 6] uses DCT and generates energy-compacted spectral components. Wavelet transform offers excellent energy compaction and eliminates redundant information present in an image. Shapiro [7] introduced EZW which is extremely fast, very effective and produces embedded bit stream. The performance of EZW is further enhanced by presenting a more efficient and faster technique known as set partitioning in hierarchical trees (SPIHT) developed by Said and Pearlman which is simple technique and offers excellent rate–distortion performance [8]. A. Islam and W. Pearlman proposed block-based wavelet transform coding technique with low complexity known as set partitioning embedded block (SPECK) [9]. JPEG2000 [10] provides higher compression efficiency which is faster than zerotree coders but it needs extra memory allocation. However, for high resolution images, SPIHT and SPECK need large memory bank which in turn increases the cost of hardware. The paper is organized as follows: Sect. 2 contains overview of the wavelet-based compression techniques, namely SPIHT, SPECK and JPEG2000. In Sect. 3, results are given. We conclude the paper in Sect. 4.

2 Overview of Wavelet-Based Compression Techniques

2.1 SPIHT

Set partition coding (SPC) has high energy compaction and zerotree properties of wavelet decomposition. This gives bitstream which is embedded in nature. This means that coding algorithm can be stopped at any time. SPIHT transmits decision by assigning bits. The pixel values are compared with the threshold using the following function:

$$\Gamma_n(X) = \begin{cases} 1, & \max_{(i,j) \in X} \{|c_{ij}|\} \geq 2^n, \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where 1 and 0 indicates the significance and insignificance of sets with respect to threshold.

The set partitioning rules are defined as:

$\hat{Q}(i, j)$ = A set of the co-ordinates of offsprings of node (i, j) ; $\hat{D}(i, j)$ = A set of the co-ordinates of descendants of node (i, j) ; $\hat{L}_r(i, j)$ = A set of co-ordinates of descendants excluding $\hat{Q}(i, j)$; $\hat{L}_r(i, j) = \hat{D}(i, j) - \hat{Q}(i, j)$.

SPIHT comprises two passes: sorting and refinement and consists of three lists to keep track of pixel values [8]. LIP is the list of insignificant pixels which has all the pixels of LL band (highest energy band). LIS is the list of insignificant sets that consist of sets with descendants only excluding tree root. LSP is the list of significant pixels that contains significant coefficients of all passes. During sorting pass or significance map testing, all the pixels in LIP are tested for significance. If they are significant, an output bit is generated and a sign bit is given. The significant

pixels are now sent to LSP. Now, tree sets with descendants are checked for significance. If they are found significant, they are detached from the list and then partitioned into four offsprings. These offsprings are tested and if found significant, they are sent to LSP otherwise they are sent to LIP. During refinement pass, those pixels which were significant in the previous sorting pass are refined and output the most significant bit of the pixel value [11, 12, 13].

2.2 *SPECK*

Set partitioned embedded block coding (SPECK) [9] is the block-based efficient state-of-the-art algorithm which has only two lists, namely LIS and LSP. There are three coding steps in SPECK algorithm: initialization, sorting and refinement. Initially, a threshold is set. The image is transformed using DWT and partitioned into a set \hat{S} which consists of root node of the pyramid and a set \hat{I} which consists of remaining coefficients. Initially, \hat{S} block coefficients are kept in LIS and LSP is empty. During the sorting pass, significance of \hat{S} and \hat{I} sets is checked. If \hat{S} is found to be significant, it is partitioned into four equal sized subsets added to LIS. All subsets of 1×1 are checked for significance and if found significant, the sign bit is coded. These subsets are then added to LSP. If the set \hat{I} becomes significant, it is divided into octave band partitioning. A total of four sets are generated: three \hat{S} sets and one \hat{I} set. When all sets are partitioned and made significant, the refinements pass starts which refines the quantized pixel values of LSP that are found significant in the earlier passes. Now, threshold is reduced in a step of two and the process is repeated until the desired bit rate is attained. Because of the linked lists of SPECK, huge memory is required that increases its complexity [14, 15].

2.3 *JPEG2000*

JPEG2000 [10] was developed by Joint Photographic Experts Group (JPEG) committee. It exploits the features of discrete wavelet transform (DWT), a benchmark in image compression and introduced several new features. These new features such as better low bit rate performance, grey and colour image compression, lossless and lossy compression, large images and image components, large dynamic range of pixels, progressive transmission, error resiliency, image security are effective for vast applications. The block diagram of JPEG2000 encoder is given in Fig. 1.

JPEG2000 encoder performs three steps in compression: discrete wavelet transform (DWT), quantization and entropy encoding. Each component after multi-component transformation is decomposed by DWT in different subbands which are then independently quantized. These quantized subbands are broken into equal sized small code blocks, generally 32×32 , 64×64 . These code blocks produce compressed bitstreams after independently entropy encoded by Tier-1 and Tier-2 coding

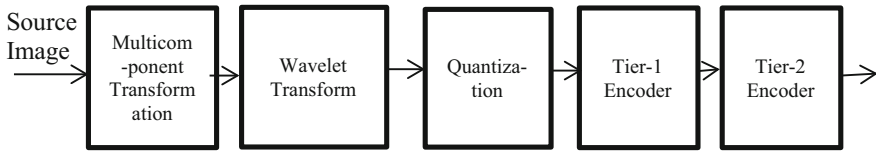


Fig. 1 JPEG2000 encoding process

engines. In Tier-1 coding process, each code block is divided into n bit planes, where n is the precision of number of elements in code blocks. They are encoded from n th bit plane to first bit plane by fractional bit plane coding using embedded block truncation optimization technique (EBCOT) [16] algorithm in three coding passes such that if a part of one bit plane is being encoded, other bit planes do not overlap. The three coding passes of EBCOT are significance propagation, magnitude refinement and clean-up pass. EBCOT generates context data and binary decision values which are coded by binary arithmetic coding (known as MQ-coder) and produce compressed bits for all code blocks. In Tier-2 coding, for each code block, a layer has all the coding passes from all consecutive bit planes and block summary information has length of compressed bitstreams, zero bit plane data and the number of coding passes data. Tag trees (quadtrees) are used to represent the layer and block summary data for all code blocks as compressed file header.

3 Simulation Results

The wavelet-based image compression techniques SPIHT, SPECK and JPEG2000 are simulated on MATLAB software. The results are tabulated as follows, and PSNR, encoding and decoding times are plotted against different bit rates for Lena and Barbara images (Tables 1 and 2).

Table 1 Image quality in terms of PSNR (dB) for image Lena

Image Size	Bit rate	SPIHT	SPECK	JPEG2000
256 × 256	0.03125	18.9583	18.9076	18.8
	0.0625	20.8117	20.8093	21.9
	0.125	22.837	22.7617	24.4
	0.25	24.9053	24.8546	26.8
	0.5	28.229	28.2168	30.4
512 × 512	0.01	19.9878	20.316	21.3
	0.03125	23.4453	23.5408	25.3
	0.0625	25.2964	25.3614	27.9
	0.125	28.0685	28.0842	30.9
	0.25	31.2954	31.2811	34.0
	0.5	34.8531	34.821	37.2

Table 2 Image quality in terms of PSNR (dB) for image Barbara

Image Size	Bit rate	SPIHT	SPECK	JPEG2000
256 × 256	0.03125	19.5478	19.558	20.2
	0.0625	21.1388	21.2202	22.2
	0.125	22.6964	22.7187	24.1
	0.25	24.4466	24.6391	27.1
	0.5	28.1264	28.4063	31.4
512 × 512	0.03125	21.4355	21.6194	22.4
	0.0625	22.4363	22.4943	23.8
	0.125	22.8963	24.1632	25.7
	0.25	26.4289	26.8417	28.8
	0.5	29.9908	30.2937	32.2

Figures 2 and 3 compare the coding performance (PSNR) (in dB) of SPIHT, SPECK and JPEG2000 algorithms for Lena image of dimension 256 × 256 and 512 × 512 against different bit rates. Figures 4 and 5 compare the PSNR (in dB) values of SPIHT, SPECK and JPEG2000 for Barbara image of 256 × 256 and 512 × 512 size, respectively. PSNR values (in dB) for JPEG2000 are best at all bit rates for these dimensions (Tables 3, 4, 5 and 6).

The encoding time is defined as the total time elapsed in wavelet transforming an image and then encoding the transformed coefficients. The decoding time is defined as the time elapsed in decoding the bitstream to reconstruct the image at a given bit rate. From Figs. 6, 7, 8 and 9, it is seen that at low bit rate, SPIHT and SPECK have

Fig. 2 PSNR (in dB) versus bit rate

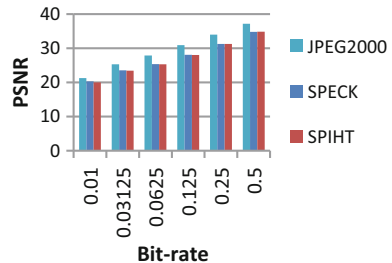


Fig. 3 PSNR (in dB) versus bit rate

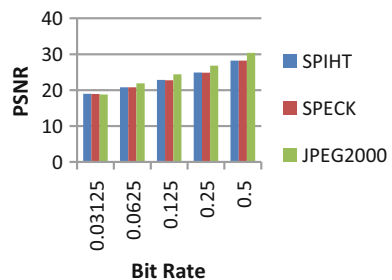


Fig. 4 PSNR (in dB) versus bit rate

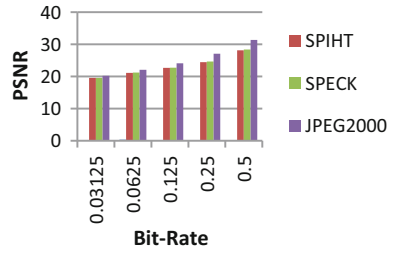


Fig. 5 PSNR (in dB) versus bit rate

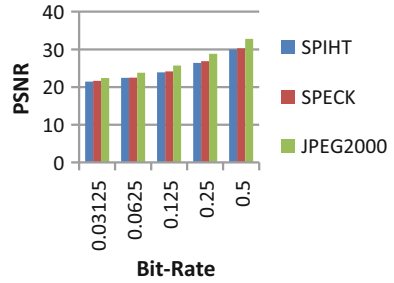


Table 3 Encoding and decoding times (in sec) for Lena image (256 × 256)

Bit rate	Encoding times			Decoding times		
	SPIHT	SPECK	JPEG2000	SPIHT	SPECK	JPEG2000
0.03125	0.371826	0.237291	0.545	0.041339	0.284069	0.265
0.0625	0.501402	0.452506	0.467	0.117489	0.388409	0.203
0.125	0.67561	0.891356	0.546	0.104909	0.631575	0.234
0.25	1.117004	1.778107	0.469	0.204936	0.896541	0.313
0.5	1.746806	3.577435	0.484	0.417989	2.010663	0.28

Table 4 Encoding and decoding times (in sec) for Lena image (512 × 512)

Bit rate	Encoding times			Decoding times		
	SPIHT	SPECK	JPEG2000	SPIHT	SPECK	JPEG2000
0.03125	0.986418	0.854543	0.67	0.11139	1.298324	0.546
0.0625	1.502201	1.717778	0.671	0.21293	2.567204	0.39
0.125	2.393825	3.384598	1.015	0.45045	5.192314	0.609
0.25	4.286269	6.804369	0.689	1.14808	10.47092	0.531
0.5	8.209132	13.54116	0.668	3.24535	20.70019	0.388

Table 5 Encoding and decoding times (in sec) for Barbara image (256 × 256)

Bit rate	Encoding times			Decoding times		
	SPIHT	SPECK	JPEG2000	SPIHT	SPECK	JPEG2000
0.03125	0.378639	0.215362	0.373	0.084359	0.44197	0.234
0.0625	0.513724	0.439732	0.514	0.06422	0.35901	0.202
0.125	0.655711	0.854807	0.404	0.110582	0.64911	0.264
0.25	1.18267	1.896267	0.545	0.211162	0.85683	0.234
0.5	2.932433	3.802933	0.594	0.447299	2.54111	0.329

Table 6 Encoding and decoding times (in sec) for Barbara image (512 × 512)

Bit rate	Encoding times			Decoding times		
	SPIHT	SPECK	JPEG2000	SPIHT	SPECK	JPEG2000
0.03125	1.03194	0.819432	0.716	0.099328	0.763043	0.404
0.0625	1.599716	1.651234	0.689	0.200636	1.003149	0.344
0.125	2.246898	3.326122	0.716	0.445806	2.163501	0.312
0.25	3.821626	7.586121	0.656	1.193864	5.805479	0.593
0.25	7.682409	13.91948	0.685	3.0199	15.50009	0.406

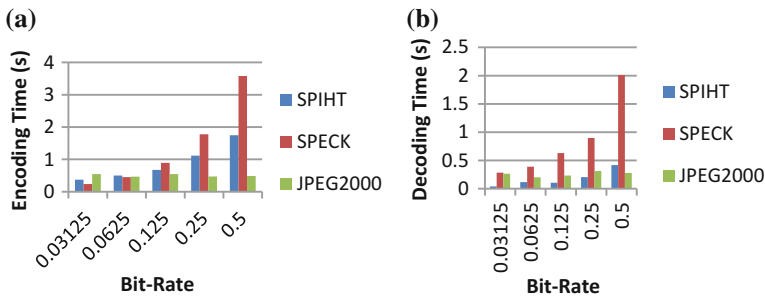


Fig. 6 Coding complexity: **a** encoding time (sec) and **b** decoding time (sec) for SPIHT, SPECK and JPEG2000 for Lena image of dimension 256 × 256

small values of the encoding and decoding times for Lena and Barbara image of 256 × 256 and 512 × 512 but at high bit rate, JPEG2000 outperforms them.

Figures 6 and 7 show the encoding and decoding times (in sec) for SPIHT, SPECK, JPEG2000 for Lena image of size 256 × 256 and 512 × 512 against different bit rates, and Figs. 8 and 9 show the encoding and decoding times (in sec) for Barbara image of size 256 × 256 and 512 × 512, respectively.

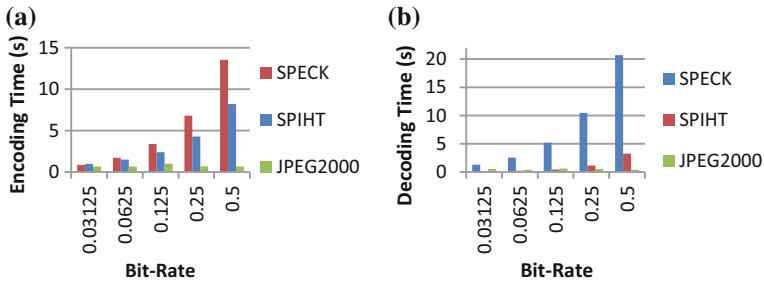


Fig. 7 Coding complexity: a encoding time (sec) and b decoding time (sec) for SPIHT, SPECK and JPEG2000 for Lena image of dimension 512 × 512

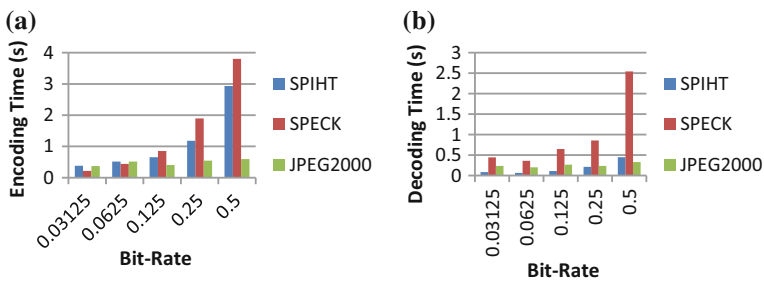


Fig. 8 Coding complexity: a encoding time (sec) and b decoding time (sec) for SPIHT, SPECK and JPEG2000 for Barbara image of dimension 256 × 256

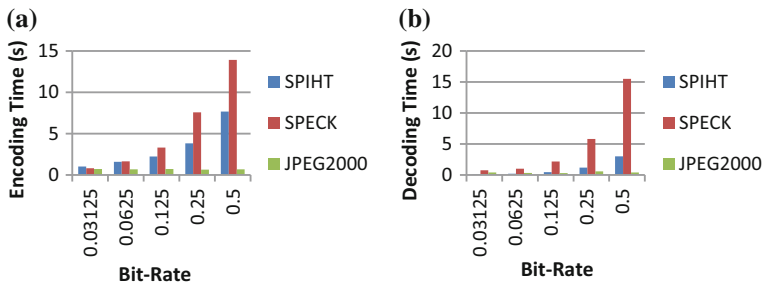


Fig. 9 Coding complexity: a encoding time (sec) and b decoding time (sec) for SPIHT, SPECK and JPEG2000 for Barbara image of dimension 512 × 512

4 Conclusion

This paper compares the set partition coding algorithms (viz. SPIHT and SPECK) and JPEG2000. The simulation results indicate that JPEG2000 technique is fast compared to set partition coding algorithms. The encoding and decoding times for

JPEG2000 are small compared to other algorithms. The PSNR values of JPEG2000 are better for a given bit rate for Lena and Barbara images of 256×256 and 512×512 .

References

1. Ahmed N., Natarajan T. and Rao K. R.: Discrete cosine transform: IEEE Transactions on Computers, vol. 23 no. 1, pp. 90–93 (1974)
2. Calderbank A. R., Daubechies I., Sweldens W., and Yeo B.L.: Lossless Image Compression Using Integer to Integer Wavelet Transforms, in Proc.: IEEE Int. Conf. Image Processing, vol. 1, Santa Barbara, CA, opp. 596–599. (1997)
3. Antonini M., Barlaud, M., Mathieu, P., and Daubechies, I.: Image coding using the wavelet transform, IEEE Trans. on Image Processing, 1, 205–220, (1992)
4. Wallace G.K., The JPEG still picture compression standard: IEEE Transaction Consumer Electronics, vol. 38, no. 1, Feb (1992)
5. ISO/IEC 1091 8 (JPEG), Information Technology-Digital Compression and Coding of Continuous-Tone Still Images.”
6. Pennebaker W. B. and Mitchell J. L.: JPEG: Still Image Compression Standard.: Van Nostrand Reinhold, New York (1993)
7. Shapiro J. M.: Embedded image coding using zerotrees of wavelet coefficients: IEEE Trans. Signal Process., vol. 41, pp. 3445–3462 (1993)
8. Said A. and Pearlman W. A.: A new, fast, and efficient image codec based on set partitioning in hierarchical trees: IEEE Trans. Circuits Syst. Video Technol., vol. 6, no. 3, pp. 243–250 (1996)
9. A. Islam and W. A. Pearlman: Embedded and efficient low-complexity hierarchical image coder: IEEE Transactions on circuits and systems for video technology, vol. 14, no. 11 (2004)
10. Taubman D.: High performance scalable image compression with EBCOT: IEEE Trans. Image Processing, vol. 9, pp. 1158–1170 (2000)
11. J.Z. Gang, G.X. Dong, L.L. Sheng: a fast image compression algorithm based on SPIHT: IEEE international conference (2009)
12. Akter M., Reaz M. B. I., Mohd-Yasin F., Choong F.: A modified set partitioning in hierarchical trees algorithm for real-time image compression: J. Commun. Technol. Electron., vol. 53, no. 6, pp. 642–650 (2008)
13. Pan H., Siu W.-C., and Law N.-F., A fast and low memory image coding algorithm based on lifting wavelet transform and modified SPIHT: Signal Process., Image Commun., vol. 23, no. 3, pp. 146–161 (2008)
14. Baojun H and Yan L.: An Improved SPECK Image Coding Algorithm.: IEEE Fifth International Conference on Information Assurance and Security, pp 227–229 (2009)
15. Senapati R. K., Pati U. C., Mahapatra K. K.: Listless block tree set partitioning algorithm for very low bit rate embedded image compression, Int. J. Electron. Commun., vol. 66, no. 12, pp. 985–995 (2012)
16. Taubman D., Ordentlich E., Weinberger M. and Seroussi G.: Embedded Block Coding in JPEG2000 (2001)