Kasetsart J. (Nat. Sci.) 47 : 925 - 939 (2013)

A Practical Wavelet Compression for Arbitrarily-Sized Natural Color Images

Poonlap Lamsrichan* and Vutipong Areekul

ABSTRACT

A practical method for wavelet-based image compression was proposed, where 'practical' means that this method can be performed on typical images of any size, whether grey-scaled or color images. Its performance with simple methods used for wavelet decomposition of an arbitrarily-sized image were compared . Color images were transformed into the appropriate domain before the wavelet decomposition was employed in the same way as grey-scaled images. The wavelet coefficient matrices were encoded with an existing bit-plane encoding algorithm of the authors, (non-list context adaptive wavelet difference reduction). With its fast and memory-effective algorithm, the proposed coder performed at the same level as the JPEG2000 standard and significantly outperformed the existing JPEG standard for all test images which were downloaded from the Internet.

Keywords: wavelet image compression, arbitrarily-sized, wavelet difference reduction, asymmetric wavelet decomposition, fast and memory-effective algorithm

INTRODUCTION

Many works have proposed image compression in the literature and evaluated the performance of their proposed algorithms using standard test images (Shapiro, 1993; Said and Pearlman, 1996; Taubman *et al.*, 2002). Test images of perfect size (divisible by 2^L for a large value of integer L) such as 512 × 512, are considered ideal for a high level of wavelet decomposition (DWT) and yield high rate-distortion performance (Taubman *et al.*, 2002). However, in real image compression applications, it is often not possible to choose the size of the image. Problems may arise when an image has a non-ideal size such as 333 × 237 pixels. Since a conventional wavelet transform divides the image into two equal parts, images containing an odd number of rows or columns cannot be decomposed simply. There are two main approaches to this problem: 1) make the image have an appropriate size for conventional transformation by adding or padding some extra pixels to the images as proposed by Jaroensawaddipong and Lamsrichan (2013); and 2) modify the wavelet transform to be capable of decomposing image of any sizes. Both of these approaches are discussed in this paper with details of their implementation and performance analysis.

Color is another issue in real-world image coding since color images are more widely used nowadays and may gain more attention from observers than grey-scaled ones. With a suitable color transform, context adaptive wavelet difference reduction or CAWDR (Lamsrichan and Sanguankotchakorn, 2006) can be applied to

Electrical Engineering Department, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand.
* Corresponding Author, e-mail: fengpll@ku.ac.th

Received date : 10/05/13 Accepted date : 22/08/13

a color image and its performance was discussed in Lamsrichan and Sanguankotchakorn (2009).

Non-list CAWDR (Lamsrichan, 2011) is an embedded wavelet image compression that encodes the whole image without any segmentation. This is the non-list version of CAWDR (Lamsrichan and Sanguankotchakorn, 2006) but it uses much less memory during the encoding process with a faster encoding time due to the smaller number of pixels examined in each bit-plane.

By using padding or non-padding techniques, non-list CAWDR can encode a color image of arbitrary size efficiently with a high level of wavelet decomposition. The results of the proposed coders are comparable to JPEG2000, the details of which can be found in Taubman and Marcellin (2002) and are much better than the current JPEG standard (Wallace, 1992). By changing the filter from 9-7 CDF (Cohen-Daubechies-Feauveau wavelet) to 17-11 CDF, the performance is slightly increased without increasing the computational time.

MATERIALS AND METHODS

Wavelet-based bit-plane encoding for image compression

Most of the existing standards and state-of-the-art algorithms for wavelet image and video coding have the main purpose of finding and encoding the wavelet coefficient as fast as possible. The largest non-zero bit of each coefficient will be encoded in the significant pass together with its sign (or polarity). The following lower significant bits will be encoded later in the (magnitude) refinement pass. The first ground-breaking embedded wavelet image encoding algorithm was the embedded zerotree wavelet or EZW (Shapiro, 1993). In the algorithm, zerotree is used to represent a large number of insignificant wavelet coefficients with just a few bits. Motivated by the EZW, set partitioning in hierarchical trees or SPIHT (Said and Pearlman, 1996) uses three types of lists of coefficients during encoding to gain better peak signal-to-noise ratio (PSNR) results with fast implementation.

Another well-known bit-plane encoding technique is embedded block coding with optimal truncation or EBCOT (Taubman *et al.*, 2002) which is utilized in the JPEG2000 standard, the details of which can be found in Taubman and Marcellin (2002). In block coding, wavelet coefficients are divided into blocks of equivalent size (except perhaps for some blocks at the border of the image). The bit-plane encoding inside each block generates a complete embedded bit-stream for scalable reconstruction from the coarsest approximation to near lossless representation. The significant pass and magnitude refinement pass will encode the bit output of each pixel in the current bit-plane starting from the highest bit-plane to the least significant bit until the last coefficient. When a pixel is found to be significant, its polarity will be encoded with sign coding. The run mode in a significant pass will be performed to encode consecutively insignificant pixels. The block coding is quite simple and fast since it can be implemented independently for each block. However, for the encoded bit-stream to be embedded, there must be another encoding pass called post-compression rate-distortion optimization (PCRD-opt). This post processing adds more complexity to JPEG2000.

Another wavelet encoding approach is wavelet difference reduction or WDR (Tian and Wells, 1998). Without separation of the normal mode and run mode, the WDR algorithm generates the significant bit, sign bit and run-length of insignificant bits in one bit-stream of a significant pass. The output symbol from WDR significant coding can be '+' for a positive significant coefficient, '-' for a negative significant coefficient and a combination of '0' and '1' which represent the distance (difference) between adjacent significant pixels in reduced form. For two significant pixels separated by N insignificant pixels, the difference between them will be N + 1.

For example, when the current bit-plane is 5, the threshold = 32 = 2^5 and the encoding sequence of pixels is 46 18 -10 8 7 -34 12 -23 33, so WDR will generate '+', followed by a run-length of 5, '-', run-length of 3 and then '+'. The reduced form of any run-length value is its binary representation with the most significant bit (MSB) omitted. For 5, its binary is '101' with the reduced form of '01' (the MSB '1' is discarded). Therefore, the encoded symbols for the above sequence in this bit-plane are '+' '0' '1' '-' '1' '+' which consists of 6 symbols (2 bits each). Without any entropy coding, the number of bits required will be 12 bits. For the case where significant pixels are not likely to happen, the distance between them will be long and the reduced form of WDR will be more effective than encoding a long sequence of '0'. For example, when the difference is 18, the binary is '10010' with the reduced form of '0010' and just 8 bits which is considerably less than the 17 bits of '0' needed by the normal mode of EBCOT. The WDR utilizes a raster scan for simplicity, and arithmetic coding (Witten *et al.*, 1987) can be used as entropy code for the output bit-stream. This simple algorithm yields slightly lower compression efficiency than SPIHT and JPEG2000 without any complicated context modeling (Tian and Wells, 1998). Some variations of WDR using context modeling produce increased performance. The quadtree relation is used in adaptively-scanned WDR (Walker, 2000). The prediction of the next significant pixels to be the neighbor pixels of the significant pixels are considered in the context model WDR (Yuan and Mandal, 2003).

Previous work on context adaptive WDR or CAWDR (Lamsrichan and Sanguankotchakorn, 2006) used both the quadtree relationship and energy compaction properties of the wavelet transform to predict the next significant pixels. Together with the context adaptive model for estimation of the probability of occurrence for symbols in the encoded stream, CAWDR produces an improved result compared to the original algorithm (Lamsrichan and Sanguankotchakorn, 2006, 2007). CAWDR has the same drawback of WDR and SPIHT, which is the implementation of lists of pixels. Whereas list implementation enables a very fast encoding speed, it consumes a lot of memory for the storage of indices of pixels for the whole image. Each pixel's address needs 8 bytes (2 + 2 for row + column numbers and 4 for the pointer) in efficient linked-list implementation. Therefore, the efficient algorithms of list-based SPIHT, WDR and all of their variants, though they operate very quickly, consume a lot of memory during the encoding/decoding processes. It should be noted that JPEG2000 does not use lists in its encoding process.

To diminish the memory consumption, packetized SPIHT (Wheeler and Pearlman, 1999) was proposed using a reduction of the image size in the manipulation (one small block of image or packet) at a time. The size of the coefficients' list was reduced; however, the complexity of packetized SPIHT was high due to the combination process of all packetized sub-images into the whole original-sized image. Another approach is the SPIHT encoder using no list as proposed in the work of Wheeler and Pearlman (2000) with slightly lower PSNR results. The non-list CAWDR (Lamsrichan, 2011) uses a status matrix that keeps the state of all coefficients instead of using a list of indices of coefficients. The amount of memory required for the status matrix is about one quarter that of original image which is much less than that of the list-based algorithm. The non-list CAWDR is the core encoder used in this research. For padded images, non-list CAWDR can be applied with no modification. For a non-padding technique, some modification in the quadtree prediction is necessary to keep the process running successfully as is described in the next section.

Practical algorithms for color image encoding of an arbitrarily sized color image transform and coding

Since the information in each color component in an RGB model is correlated at

some scale, it can be thought of as redundancy in the image. This correlation can be reduced by transforming the images into another color model, the YCbCr model, where Y is the intensity and Cr and Cb represent the importance of 'red' and 'blue' color in that image, respectively. The transformation can be obtained by using the matrix operations in Equations 1 and 2:

For RGB to YCbCr:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

For YCbCr to RGB:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.344136 & -0.714136 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \quad (2)$$

These transforms guarantee a perfect reconstruction (after some rounding).

The Y, Cb and Cr components are then transformed with L-level wavelet decomposition using a 9-7 biorthogonal (CDF) filter. Although the YCbCr color components are assumed to be de-correlated, they are not totally independent. The contextual information from one color component can be used for the significance prediction in the other two color components. The prediction for the significant pixel is illustrated in Figure 1.

To make the encoded bit stream embedded, color components are encoded one after another in each bit-plane. The component Y is encoded first, followed by Cb and Cr, respectively. To make the bit allocation more distributed, the encoding process in each bit-plane is started with a significant pass of all components and then followed by a refinement pass of all components. The data structure of the embedded encoded image is shown in Figure 2.

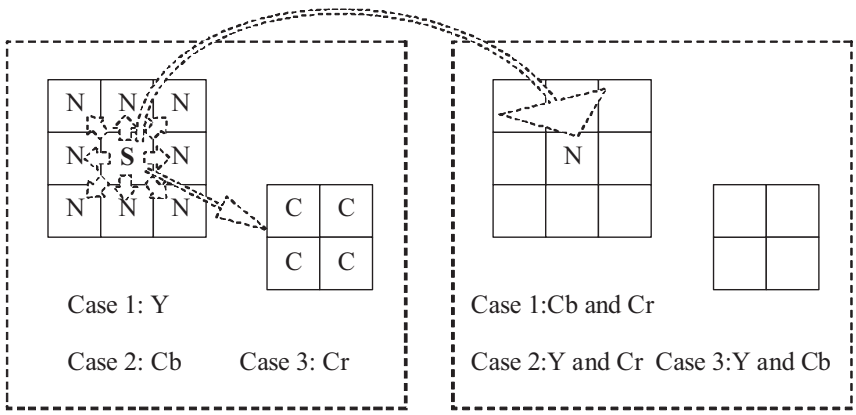

Figure 1 Prediction of the next significant pixels between three color components. S = Significant pixel, N = Pixel predicted to be significant in the next bit-plane, C = Pixel predicted to be significant in the next bit-plane (lower priority to N).

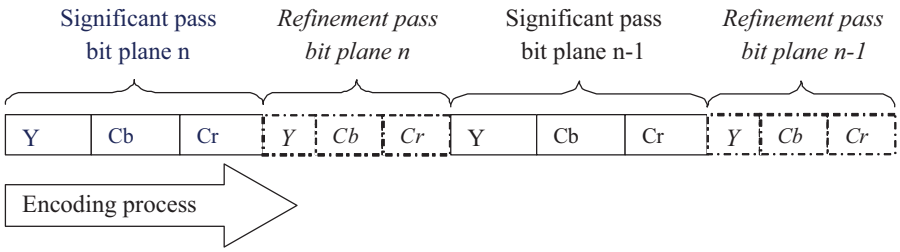

Figure 2 Data arrangement in embedded color image encoding.

Wavelet decomposition for image with arbitrary size

Real-world images can be of any size, especially partially cropped ones. When the image size is appropriate, a large number of wavelet decompositions can be performed and the energy of an entire image will be clustered into a small area of the highest level of all low-pass sub-bands. Figure 3 shows that the higher the level of wavelet transform that is applied, the higher the compression performance that can be achieved with the same compression algorithm. For images of normal to large size, the optimal number of levels may be in the range of 5–7 levels based on the authors' experience.

Since the conventional wavelet decomposition separates the signal into two equal-sized parts, a low frequency sub-band and a high frequency one, for an image to be able to be decomposed into L levels, it must have both the row and column sizes divisible by 2^L. For an image with inappropriate size, a low value of L will be used resulting in a lower energy compaction rate and, consequently, lower compression performance. To increase the value of L with the conventional wavelet decomposition, some extra pixels are added to the image at the boundary (both row and column, if it is needed). This idea originated in the work of Liang *et al.* (2008) who proposed many ways of adding or padding extra pixels. It was concluded that padding with the same pixels as the pixel at the border repetitively yielded the best performance compared with other techniques when SPIHT was a core encoder. In the above paper, the method to encode an arbitrarily-sized image without padding any extra pixels was also proposed. However, the encoding results of the original-sized images were somewhat lower in PSNR than the padded ones. The authors of that paper confirmed the perfect reconstruction of the arbitrarily-sized wavelet transform.

For techniques involving multiples of 2^L rectangular padding, the added pixels in each dimension will be in the range $[0,2^L-1]$. When the size of the original image is much larger than 2^L, the increased pixels are not noteworthy. There are three main padding methods:

1. All pixels of the same value such as zero (black pixels) or 255 (white pixels).
2. The repetition of the boundary pixel of the original image. This method can minimize the high frequency components in the wavelet transform.

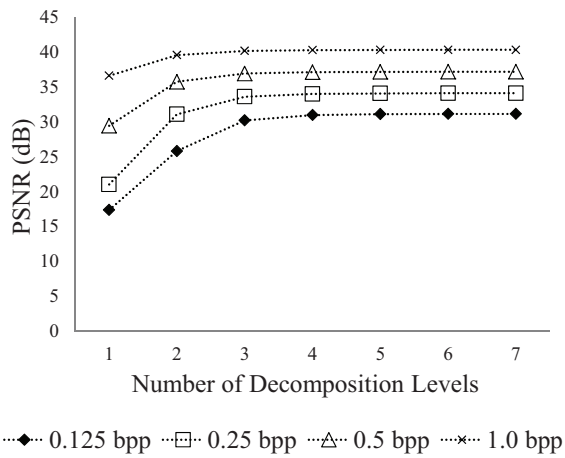

Figure 3 Peak signal-to-noise ratio (PSNR) results of a sample picture in Figure 4 using context adaptive wavelet difference reduction at various levels of wavelet decomposition and bit-per-pixel (bpp).

3. The reflection of the border pixels of the image. With this padding technique, the wavelet coefficients in the padding area are also a reflection of the coefficients in the original area. This is due to the symmetry of the wavelet filter taps.

In summary, these padding techniques produce a larger size of image and all of the pixels are needed to be encoded and decoded definitely. The padding pixels above have a different purpose from the symmetrical extension for the filtering process. The symmetrical extension at the boundary of the image is to ensure perfect reconstruction of the image to the last pixel at the boundary while reducing the effect of Gibbs phenomenon. The symmetrical extended pixels are discarded after the transformation and the size of the image is not changed by using a symmetrical extension.

Figure 4 demonstrates the different padded images. Figures 4a and 4b show padding with all the pixels of the same color, being 'black' and 'white', respectively. Figure 4c shows the image padded with a repetition of the border pixels. Figure 4d is padded with the reflection of the border pixels. These padded images are used for illustrative purpose only.

The number of pixels in the padding area can be calculated using Equation 3:

$$Nsize = \left\lceil \frac{size}{2^L} \right\rceil \times 2^L \quad (3)$$

where size is the number of row or column of the original image, Nsize is the number of rows or columns in the padded image, L is the number of required wavelet decomposition levels and $\lceil x \rceil$ is the lowest integer greater than or equal to x.

The number of pixels that must be added in each dimension is in the range $[0,2^L-1]$ with an estimated value of 2^{L-1}. The test images in Figure 5 have been used to demonstrate the size of the ratio of extra pixels that need to be added, as shown in Table 1.

Figure 4 Sample image showing: padded images (a) black; (b) white; (c) repetition; (d) reflection.

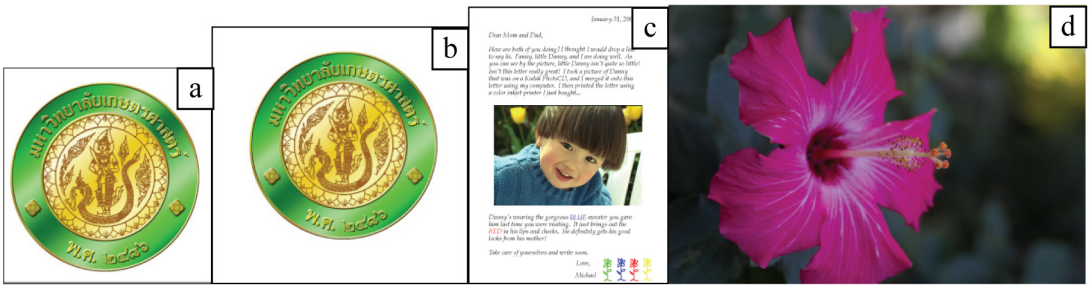

Figure 5 Test images: (a) LOGO and (b) LOGO2 (Kasetsart University, 2013); (c) CMPND2 (International Telecommunication Union, 2013); (d) FLOWER_FOVEON (Rawzor, 2013).

Table 1 Percentage of padding pixels compared to the original images in Figure 5.

Image	Original size (row×column)	Decomposition level	Padded size (row×column)	% of extra pixels
CMPND2	1,433,600 (1400×1024)	3	1,433,600 (1400×1024)	0.00
		5	1,441,792 (1408×1024)	0.57
		7	1,441,792 (1408×1024)	0.57
LOGO	116,245 (347×335)	3	118,272 (352×336)	1.74
		5	123,904 (352×352)	6.59
		7	147,456 (384×384)	26.85
LOGO2	360,000 (600×600)	3	360,000 (600×600)	0.00
		5	369,664 (608×608)	2.68
		7	409,600 (640×640)	13.78
FLOWER_ FOVEON	3,429,216 (1512×2268)	3	3,435,264 (1512×2272)	0.18
		5	3,489,792 (1536×2272)	1.77
		7	3,538,944 (1536×2304)	3.20

From Table 1, when the original image has a small size, a high decomposition level can result in a high ratio of padding pixels.

Referring to the images in Figure 5, LOGO, at 7 levels, needs more than one quarter extra padding pixels compared to original image. Large images (CMPND2 and FLOWER_FOVEON) need a relatively small number of padding pixels even at the high decomposition level.

A non-padding technique is a different approach to increase the number of wavelet transforms for an image with an inappropriate size. Instead of changing the image size and using a conventional wavelet transform that can handle only an even number in each decomposition, the image is kept unchanged and the wavelet transform is modified to be able to decompose odd-numbered sequences successfully. Starting with a signal of arbitrary size, it is passed through lowpass and highpass filters simultaneously. The output of each filter will be subsampled (decimated) by choosing even-indexed samples for the lowpass and odd-indexed ones for the highpass. With this sampling policy, the original image with its length being an even number will have the same length for the two sub-images. When the size of the original image is an odd number, the low frequency sub-image will have one more pixel than the high frequency sub-image. If the size of original sequence is X, the length of the lowpass sequence will simply be Floor((X + 1) / 2) or $\lfloor (X + 1) / 2 \rfloor$ and the length of the highpass sequence can be calculated with $\lfloor X/2 \rfloor$. This formula gives the correct results for when X is odd and X is even, since $X = \lfloor (X+1)/2 \rfloor + \lfloor X/2 \rfloor$ is always true for all positive integers X. Figure 6 shows how to perform this asymmetric wavelet transform of arbitrarily-size sequence. The just-mentioned asymmetric decomposition will be called asymmetric DWT Type I in this paper. The downsampling policy of lowpass and highpass can be interchanged; that is, the lowpass sub-image can be selected from odd-indexed samples and the highpass sub-image selected from even-indexed samples. This latter method will be called asymmetric DWT Type II. Although the transformed coefficients of Type I and Type II are not exactly the same, both have very similar

structures and guarantee perfect reconstruction. Moreover, the coding performances of both Type I and Type II asymmetric DWT are almost the same. Therefore, the PSNR results of the non-padding method are selected to be from using only Type I from now on in this paper.

The symmetrical extension used here is simple symmetry without repetition at the boundary pixel (see Figure 6b). The objective of this symmetrical extension is for perfect reconstruction in the inverse wavelet transform. Since the wavelet filter has symmetrical coefficients, the wavelet-transformed output sequence will be symmetrical at the boundary in the same way as the extended sequence in the spatial domain. The downsampling of the sequence starts with the zeroth index and collects every even numbered index of the lowpass output sequence. For the highpass output, only odd numbered samples are kept. The authors intentionally left the extension pixels of the decimated sequence in Figures 5e and 5f just to illustrate how to appropriately extend the decoded coefficients to obtain a perfect reconstruction process in the reverse wavelet transform. Figure 5 can be used as a guideline to determine the suitable type of symmetric extension for a sequence with an even number.

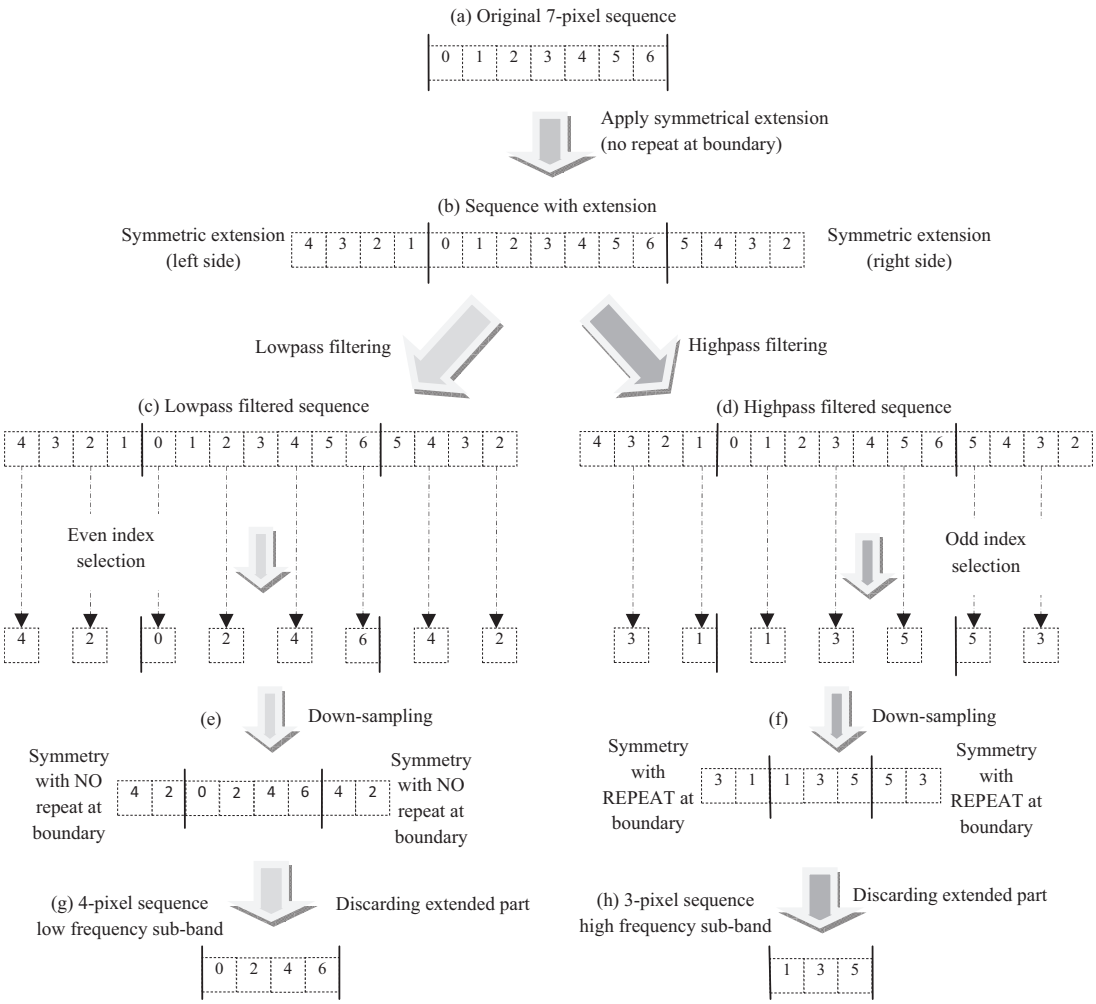

Figure 6 Asymmetric wavelet decomposition (Type I) for sequence of arbitrary size, with (a)–(h) showing the sequence.

The problem of imperfection in the quadtree relationship may arise in the non-padding wavelet decomposition since the number of children is not always four for each parent pixel. The algorithm to address this problem has been already stated in Liang *et al.* (2008) and is also applied in the current work.

After getting wavelet coefficients, the encoding process will follow the methods described in the non-list CAWDR (Lamsrichan, 2011). The PSNR results of the proposed coder using all of the mentioned techniques are shown in the next section.

RESULTS AND DISCUSSION

The test set of four color images (Figure 5) of various sizes downloadable from the Internet are used in this section. The first two images, 'LOGO' and 'LOGO2', are images of logos of Kasetsart University, one of the leading universities in Thailand. LOGO has 347 × 335 pixels, whereas LOGO2's dimensions are 600 × 600. Kasetsart University has a circular-shaped logo located in the middle of a white background. The rapid changes of intensity between the logo and the background results in sharp edge and high frequency components in the wavelet domain (see Figures 5a and 5b). According to its odd-numbered size, LOGO cannot be decomposed with conventional wavelet transform and LOGO2 can be transformed with three levels. The next test image is 'CMPND2' (Figure 5c), from the test set of the Telecommunication Standardization Sector of the International Telecommunications Union. With a size of 1400 × 1024 pixels, CMPND2 allows just three levels of wavelet transform. This image represents the combination of natural (a boy's face) and computer-generated (text) components in the same image. The last test image is 'FLOWER_FOVEON' with 1512 × 2268 pixels (Figure 5d), which represents a natural scene image with a combination of various frequencies from the high (flower pollen) to middle (flower petals) and then very low frequency of the blurred background.

The average PSNR values of the reconstructed images at various bit-per-pixel (bpp) levels and at many decomposition levels can be calculated from Equation (4).

$$\mathrm{PSNR}=10\log\left(\frac{255^2}{(\mathrm{MSE}_R+\mathrm{MSE}_G+\mathrm{MSE}_B)/3}\right) \quad (4)$$

where MSE_C is mean square error of the color component C which can be red (R), green (G) or blue (B).

Table 2 shows the performance of all four padding techniques along with only one non-padding technique (Type I). As expected, the padding with abruptly-changed white and black

Table 2 Peak signal-to-noise ratio (dB) from encoding 'FLOWER_FOVEON' with each padding technique.

bpp	Non-padding	Black-padding	White-padding	Repetitive padding	Symmetrical padding
0.0625	38.45	37.99	37.95	38.47	38.45
0.125	41.28	40.96	40.94	41.28	41.26
0.25	44.08	43.84	43.84	44.06	44.05
0.5	46.62	46.59	46.58	46.63	46.63
0.75	48.48	48.41	48.39	48.47	48.46
1	49.15	49.10	49.09	49.13	49.12
2	51.09	51.09	51.09	51.09	51.09

The image FLOWER_FOVEON is presented in Figure 5d.
bpp = Bit-per-pixel.

pixels resulted in high frequency components and low compression performance and so they cannot compete with the other padding techniques. Consequently, both 'white' and 'black' pixel padding will be omitted in the comparison tables and graphs from now on. The results from repetitive padding were always equal to or slightly better than the symmetrical padding because the repetitive pixels remain unchanged for some duration and represent the lower frequency components in the wavelet domain than in the symmetrical padding ones. The full results of all padding techniques are shown in detail in Jaroensawaddipong and Lamsrichan (2013).

For the non-padding technique, the number of wavelet coefficients to be encoded is the same as that of the original image, which is somewhat lower than the number of pixels from any padding method. However, the results from non-padding are almost the same as those of repetitive padding. This may result from the imperfection of the quadtree relationship in the non-padding wavelet decomposition since the number of children is not always four for each parent pixel. Another reason is that the location of the quadtree may not occur at exactly the same spatial location as their parent (that is, it is approximately the same, but not exactly).

Table 3 presents the coefficients of wavelet filters used in this paper, being the well-known CDF 9-7 and the longer CDF 17-11. The 17-11 filter is included in this experiment to show that the longer length of the filter yields slightly better performance for some images. Table 4 shows the experimental results of encoding all test images using the proposed techniques. Only the PSNR results from repetitive padding and non-padding (modified wavelet decomposition) have been chosen to be shown in the table since they are among the best of all the padding techniques. In

Table 3 Wavelet filter coefficients used in the study.

Filter CDF 9-7 coefficient index	Analysis		Synthesis	
	Lowpass	Highpass	Lowpass	Highpass
0	0.852698679	-0.788485616	0.788485616	-0.852698679
±1	0.377402856	0.418092273	0.418092273	0.377402856
±2	-0.110624404	0.040689418	-0.040689420	0.110624404
±3	-0.023849465	-0.064538883	-0.064538880	-0.023849465
±4	0.037828456			-0.037828456

Filter CDF 17-11 coefficient index	Analysis		Synthesis	
	Lowpass	Highpass	Lowpass	Highpass
0	0.825922997	-0.758907729	0.758907729	-0.825923000
±1	0.420796285	0.417849109	0.417849109	0.420796285
±2	-0.094059200	0.040367979	-0.040367980	0.094059203
±3	-0.077263170	-0.078722001	-0.078722000	-0.077263170
±4	0.049732903	-0.014467505	0.014467505	-0.049732900
±5	0.011934565	0.014426283	0.014426283	0.011934565
±6	-0.016990640			0.016990640
±7	-0.001914290			-0.001914290
±8	0.001908832			-0.001908830

the comparison tables or graphs, the results from the existing standards JPEG (Wallace, 1992) and JPEG2000 (the details of which can be found in Taubman and Marcellin, 2002) are also included as the reference bars. While the CDF 9-7 filter can perform quite well for an image with sharp-edged details, the encoding performance of the CDF 17-11 filter is slightly better for images of natural scenes.

From all the coders in Table 4, the mostly used JPEG standard always produced the lowest PSNR values with a substantial margin of 2–15 dB. The proposed coders performed at the same level as JPEG2000. Since JPEG2000 has a large number of bits dedicated to header or side information (for its scalability of block coding), the PSNR results in low bit rates were not excellent and these places are where the proposed coders

Table 4 Experimental results of encoding all test images using the proposed techniques.

Image	bpp	CAWDR with CDF 9-7		CAWDR with CDF 17-11		JPEG 2000	JPEG
		Non-padding (6 levels)	Repetitive padding (6 levels)	Non-padding (6 levels)	Repetitive padding (6 levels)		
CMPND2	0.125	22.20	22.20	22.16	22.16	22.12	-
	0.25	26.89	26.89	26.57	26.54	26.33	20.91
	0.5	32.35	32.34	31.90	31.89	32.29	25.88
	0.75	36.49	36.48	36.04	36.03	36.63	29.73
	1	39.73	39.72	39.22	39.20	39.93	32.72
	2	48.41	48.41	47.84	47.83	48.66	39.18
LOGO	0.125	19.22	19.16	19.25	19.17	18.64	-
	0.25	20.73	20.71	20.91	20.86	20.49	18.40
	0.5	23.46	23.38	23.43	23.35	23.19	21.68
	0.75	25.25	25.21	25.41	25.31	25.26	23.21
	1	27.06	26.98	27.05	26.98	27.00	24.24
	2	32.22	32.16	32.25	32.14	32.32	29.04
LOGO2	0.125	23.16	23.16	23.30	23.33	22.98	-
	0.25	25.68	25.67	25.73	25.74	25.68	22.29
	0.5	29.27	29.25	29.32	29.32	29.3	26.27
	0.75	31.76	31.75	31.82	31.81	31.91	28.24
	1	33.98	33.96	33.93	33.92	34.1	29.57
	2	40.59	40.56	40.56	40.55	40.79	34.52
FLOWER_	0.125	41.28	41.28	41.54	41.55	41.6	25.30
FOVEON	0.25	44.08	44.06	44.32	44.33	44.42	38.39
	0.5	46.62	46.63	46.90	46.90	46.9	41.98
	0.75	48.48	48.47	48.58	48.56	48.1	43.63
	1	49.15	49.13	49.27	49.25	48.52	44.18
	2	51.09	51.09	51.24	51.24	48.52	46.33

The images are presented in Figure 5.

CAWDR = Context adaptive wavelet difference reduction; CDF 9-7 and CDF 17-11 = Cohen-Daubechies-Feauveau wavelet filter types.

bpp = Bit-per-pixel.

performed relatively better. For small-sized images such as LOGO and LOGO2, the advantage of the low header coders remained until around 0.5 bpp. For higher bpp values, the effectiveness of the binary adaptive arithmetic encoder of JPEG2000 resulted in better efficiency and slightly better gains in the PSNR. The deficiency of JPEG2000 in the very low bpp values was not seen in the large-sized FLOWER_FOVEON since its large size led to many available bit budgets even in the 0.125 bpp (larger than 50 kBytes available). The results of the coders using CDF 9-7 and 17-11 filters were about the same. CDF 9-7 was appropriate for an image with a scene containing abrupt changes such as CMPND2, whereas CDF 17-11 performed better in the natural scene image of the FLOWER_FOVEON.

Graphical comparisons between the proposed non-padding CAWDR using 9-7, 17-11 filters and the JPEG2000 standard for all test images are shown in Figure 8. The PSNR values displayed are the values that relate to the JPEG standard; the positive values in the graph indicate how much better the PSNR values are compared to the JPEG standard. It can be seen that all coders are better than JPEG at all compression ratios for all test images. There is some indication that the results from JPEG2000 of FLOWER_FOVEON at high bit rates are much lower than the proposed coders due to the fact that JPEG2000 encodes FLOWER_FOVEON until the point of near-lossless around 0.75 bpp and always stops encoding more bits even though it is assigned to encode the image to the higher bit rate.

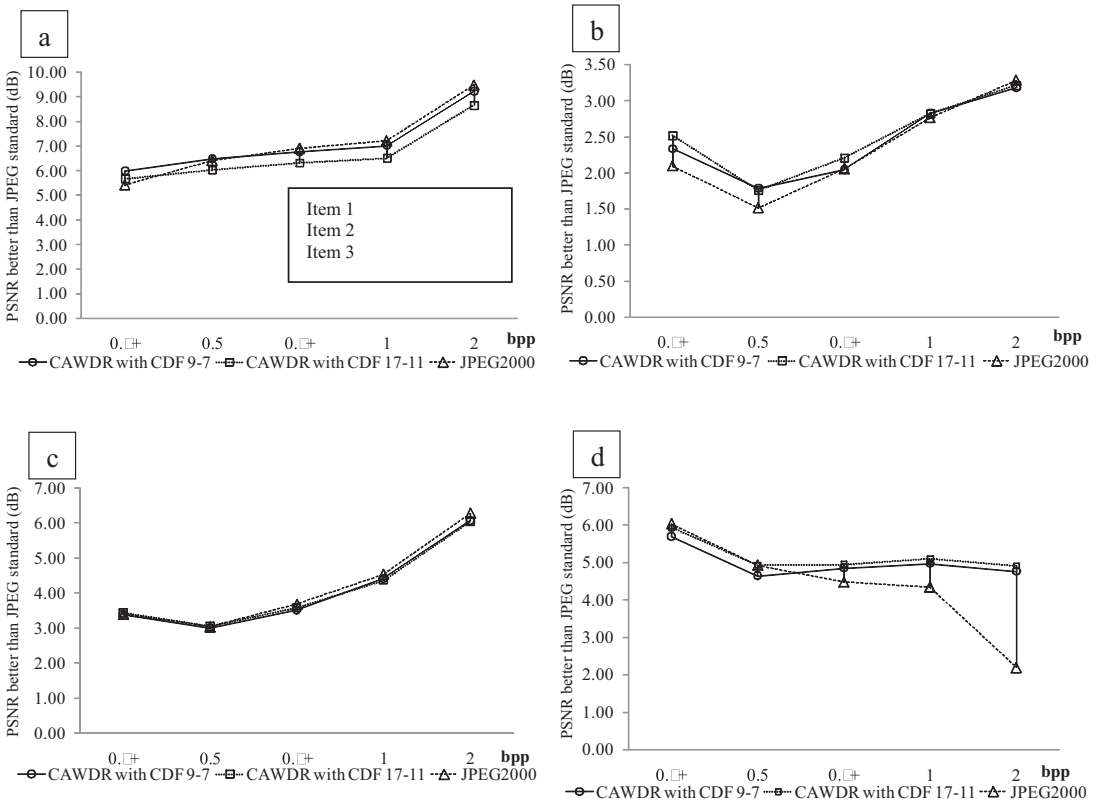

Figure 8 Comparison between the proposed coders and the JPEG2000 standard for the sample images: (a) CMPND2 (b) LOGO (c) LOGO2 (d) FLOWER_FOVEON. (The sample images are presented in Figure 5. PSNR = Peak signal-to-noise ratio, bpp = Bit-per-pixel.)

Subjective comparisons of the reconstructed images from all three coders provided in Table 3 are presented in Figure 9 (cropped CMPND2) and Figure 10 (cropped FLOWER_FOVEON). The blocking effect of the JPEG image at this very low bit rate can be seen without difficulty. The proposed coders (repetitive padding with six levels of decomposition) reveal more detail in the boy's hair and shirt than JPEG2000. The coding results of the proposed coder for other standard test images, such as Lenna, Barb, Bike, among others, can be found in Lamsrichan (2011).

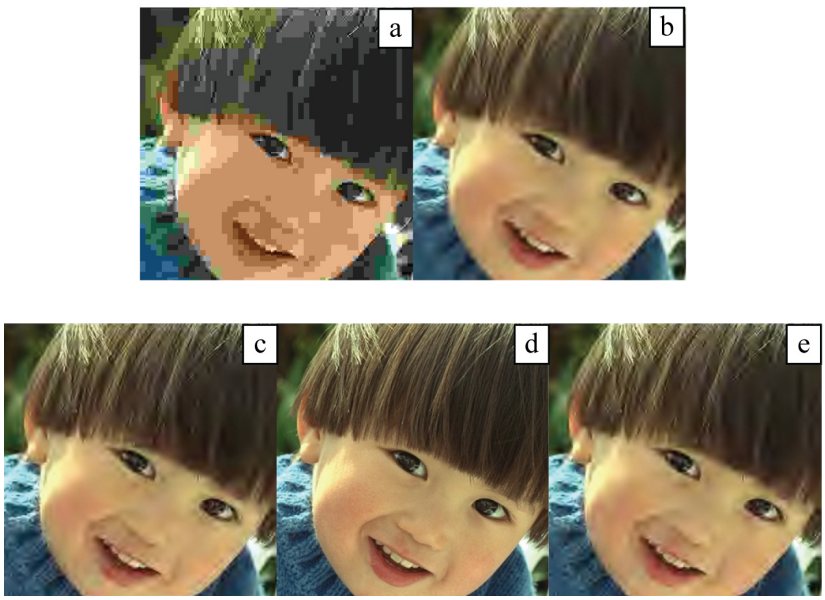

Figure 9 Subjective comparison for 'CMPND2' image encoded at 96:1 ratio or 0.25 bit-per-pixel: (a) JPEG, (b) JPEG2000, (c) CAWDR 17-11 CDF Filter, (d) Original and (e) CAWDR 9-7 CDF Filter.

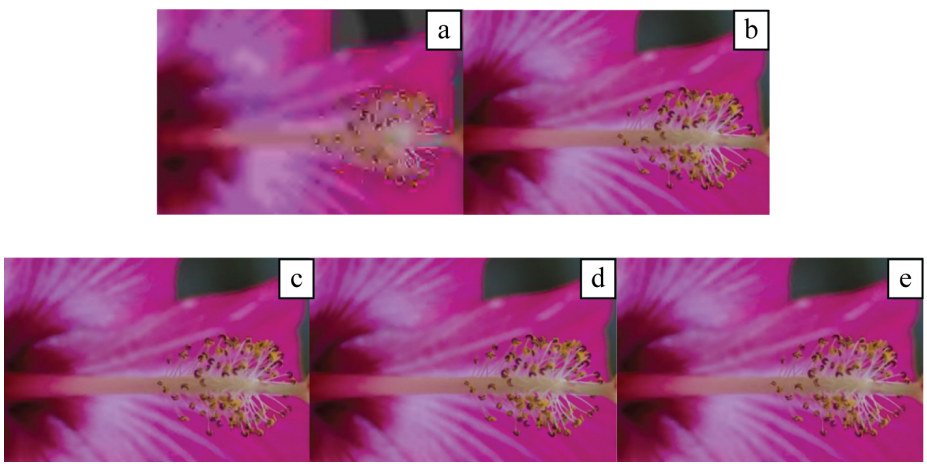

Figure 10 Subjective comparison for 'FLOWER_FOVEON' image encoded at 192:1 compression ratio or 0.125 bit-per-pixel: (a) JPEG, (b) JPEG2000, (c) CAWDR 17-11 CDF Filter, (d) Original and (e) CAWDR 9-7 CDF Filter.

CONCLUSION

A practical algorithm for the compression of color images was proposed with the proposed coder proving that it can compress color images of arbitrary sizes successfully with substantially better performance than the mainly used current JPEG standard. After the conventional color transform of the RGB components into a YCbCr domain, many levels of decomposition could be applied to the image of arbitrary size by padding some extra pixels or modification of the wavelet decomposition. The non-padding technique and the repetitive padding technique gave the best results among all the proposed techniques and were comparable to the JPEG2000 standard for images from the test set. For images with sharp-edged details (LOGO and CMPND2), encoders with the CDF 9-7 filter provided satisfactory rate-distortion performance, especially for low bit rates. For natural scenery images of large size such as FLOWER_FOVEON, the proposed coders with the CDF 17-11 filter gave slightly better performance than the ones with the CDF 9-7 filter. With its fast and simple algorithm together with low memory requirement during the encoder process, the non-list CAWDR encoder for arbitrarily-sized color images is comparable with other state-of-the-art image coders. The rate-distortion performance of the proposed coder can be improved by using more efficient context adaptive arithmetic coding for a WDR-based or even a hybrid encoder which could select whether to use normal bit-plane encoding or WDR run-length encoding in the current sub-band of the wavelet transform.

ACKNOWLEDGEMENTS

The authors would like to express their gratitude to THNIC, TRF and the Kasetsart University Research and Development Institute (KURDI) for supporting this research.

LITERATURE CITED

International **Telecommunication Union. 2013. Test Images.** [Available from: http://ties.itu.int/ftp/public/itu-t/tsts/GenImage/T024/]. [Sourced: 20 May 2013].

Jaroensawaddipong, N. and P. Lamsrichan. 2013. Arbitrary-sized wavelet image compression, pp. 55–58. *In* ICICTES 2013, (eds.). **Proceeding of the International Conference on Information and Communication Technology for Embedded Systems (IC-ICTES 2013).** 24–26 January 2013, Samut Songkram, Thailand.

Kasetsart University. 2013. **KU Logo.** [Available from: http://kucitypic.kasetsart.org/kucity.com7/KU_Logo/KU_Logo.rar] . [Sourced: 20 May 2013].

Lamsrichan, P. 2011. A fast algorithm for low-memory embedded wavelet-based image coding without list, pp. 979–982. *In* ECTI-CON 2011, (eds.). **Proceeding of Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).** 17–20 May 2011, Khon Kaen, Thailand.

Lamsrichan, P. and T. Sanguankotchakorn. 2006. Embedded image coding using context-based adaptive wavelet difference reduction, pp. 1137–1140. *In* ICIP 2006, (eds.). **Proceeding of IEEE International Conference on Image Processing.** 8–11 October 2006, IEEE, Atlanta, GA, USA.

_____. 2007. Embedded image coding using context adaptive wavelet difference reduction, pp. 81–89. *In* IEICE, (eds.). **Transaction of IEICE Transaction on Information and Systems, Special Section on Advanced Image Technologies.** January 2007, IEICE, Tokyo, Japan.

_____. 2009. Embedded color image coding with context adaptive wavelet difference reduction, pp. 347–350. *In* ISPACS 2009,

(eds.). **Proceedings of Intelligent Signal Processing and Communication Systems.** 7–9 December 2009, Kanazawa, Japan.

Liang, Z., W. Dentin and R. Klepko. 2008. On wavelet coding of arbitrarily sized images, pp. 1–5. *In* BMSB, (eds.). **Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting.** 31 March–2 April 2008, Las Vegas, NV, USA.

Rawzor. 2013. **Images Compression–Benchmark: The New Test Images.** [Available from: http://www.imagecompression.info/test_images]. [Sourced: 20 May 2013].

Said, A. and W.A. Pearlman. 1996. A new, fast and efficient image codec based on set partitioning in hierarchical trees. **IEEE Trans. on Circuits and Systems for Video Technology** 6(3): 243–250.

Shapiro, J.M. 1993. Embedded image coding using zerotrees of wavelet coefficients. **IEEE Transaction on Signal Processing** 41: 3445–3462.

Taubman, D. and M.W. Marcellin. 2002. **JPEG2000 Image Compression Fundamentals, Standards and Practice.** Kluwer Academic. Norwell, Massachusetts. 773 pp.

Taubman, D., E. Ordentlich, M. Weinberger and G. Seroussi. 2002. Embedded block coding in JPEG2000. **Signal Processing-Image Communication.** 17(1): 49–72.

Tian, J. and R.O.J. Wells. 1998. Embedded image coding using wavelet-difference reduction, pp. 289–301. *In* P. Topiwala, (ed.). **Wavelet Image and Video Compression.** Kluwer Academic. Norwell, MA, USA.

Walker, J.S. 2000. A lossy image codec based on adaptively scanned wavelet difference reduction. **Optical Engineering** 39(7): 1891–1897.

Wallace, G.K. 1992. The JPEG still picture compression standard. **IEEE Transactions on Consumer Electronics.** 38(1): 18–34.

Wheeler F.W. and W.A. Pearlman. 1999. Low-memory packetized SPIHT image compression, pp. 1193–1197. *In* ACSSC, (eds.). **Proceedings of 33th Asilomar Conference on Signals, System and Computers.** 24–27 October 1999, Monterey, CA, USA.

_____. 2000. SPIHT image compression without lists. **Proceedings of IEEE International Conference on Acoustic Speech and Signal Processing** 4: 2047–2050.

Witten, I.H., R.M. Neal and J.G. Cleary. 1987. Arithmetic coding for data compression. **Communications of the ACM.** 30: 520–540.

Yuan, Y. and M.K. Mandal. 2003. Context-modeled wavelet difference reduction coding based on fractional bit-plane partitioning. **Proceedings of IEEE International Conference on Image Processing** 2: 251–254.