

A Lossy Image Codec Based on Adaptively Scanned Wavelet Difference Reduction

James S. Walker
Department of Mathematics
University of Wisconsin–Eau Claire
Eau Claire, WI 54702–4004

Phone: 715–836–3301
Fax: 715–836–2924
e-mail: walkerjs@uwec.edu

To appear in *Optical Engineering*

Abstract

This paper describes a new method of lossy still image compression, called *Adaptively Scanned Wavelet Difference Reduction* (ASWDR). The ASWDR method produces an embedded bit stream with region of interest capability. It is a simple generalization of the compression method developed by Tian and Wells, which they have dubbed *Wavelet Difference Reduction* (WDR). While the WDR method employs a fixed ordering of the positions of wavelet coefficients, the ASWDR method employs a varying order which aims to adapt itself to specific image features. This image adaptive approach enables ASWDR to outperform WDR in a rate-distortion sense, and to essentially match the rate-distortion performance of the widely used codec, SPIHT, of Said and Pearlman. ASWDR compressed images exhibit better perceptual qualities, especially at low bit rates, than WDR and SPIHT compressed images. ASWDR retains all of the important features of WDR: low complexity, region of interest capability, embeddedness, and progressive SNR.

Keywords: Image compression; wavelet transform; signal processing.

1 Introduction

This paper describes an improvement of the WDR algorithm of Tian and Wells ([1] and [2]), referred to as ASWDR. The ASWDR algorithm aims to improve the subjective perceptual qualities of compressed images and improve the results of objective distortion measures. We shall treat two distortion measures, PSNR and *edge correlation*, which we shall define in the section on experimental results. PSNR is a commonly used measure of error, while edge correlation is a measure that we have found useful in quantifying the preservation of edge details in compressed images, and seems to correspond well to subjective impressions of the perceptual quality of the compressed images.

ASWDR achieves these improvements of WDR while retaining all of the important features of WDR, such as low complexity, region of interest (ROI) capability, embeddedness, and progressive SNR capability. The improved quality of ASWDR compressed images at low bit rates, plus its ROI capability, has applications to image database search/retrieval, to remote medical image transmission/diagnosis, and to multi-resolution methods for reconnaissance and feature extraction.

The paper is organized as follows. Section 2 briefly describes the WDR algorithm. In Section 3 we discuss the rationale behind ASWDR and give a detailed description of this new algorithm. In Section 4 we discuss experimental results of applying ASWDR to compressing test images, and compare it to WDR and the SPIHT algorithm [3]. A brief concluding section ends the paper.

2 The WDR algorithm

The ASWDR method is a generalization of the WDR method of Tian and Wells ([1] and [2]), so we shall begin by briefly summarizing the WDR method. The WDR method has two principal advantages. First, it produces an embedded bit stream—thereby facilitating progressive transmission over small bandwidth channels and/or enabling multiresolution searching and processing algorithms. Second, it encodes the precise indices for significant transform values—thereby allowing for Region of Interest (ROI) capability and for image processing operations on compressed image files [4].

The WDR algorithm is a remarkably simple procedure. A wavelet transform is applied to the image. Then, the bit-plane encoding procedure for the transform values, described in [5] and [3], is carried out. This bit-plane encoding procedure consists of a *significance pass* and a *refinement pass*. During the significance pass, the values of the wavelet transform of the image are scanned through in a linear order (say, $x[1], x[2], \dots, x[M]$, where M is the number of pixels), and a value is deemed significant if it is greater than or equal to a threshold value. An index n is removed from the scanning order if it is found to be significant. During the refinement pass, previous significant values are refined to a further precision.

The distinguishing feature of WDR is its method of encoding the positions of significant transform values. This method is called *difference reduction*. It is most easily described through an example. Suppose that the significant indices found in the significance pass are 2, 3, 7, 12, and 34. Rather than work with these values, we work with their successive differences: 2, 1, 4, 5, 22. In this latter list, the first number is the *starting index* and each successive number is the *number of steps* needed to reach the next index. The binary expansions of these successive differences are $(10)_2$, $(1)_2$, $(100)_2$, $(101)_2$, and $(10110)_2$. Since the most significant bit in these expansions is always 1, we can drop this bit and use the signs of the significant transform values as separators in the symbol stream. For example, suppose that these significant transform values are $x[2] = +34.2$, $x[3] = -33.5$, $x[7] = +48.2$, $x[12] = +40.34$, and $x[34] = -54.36$, then the

resulting symbol stream would be $+0 - +00 + 01 - 0110$.

When arithmetic coding is employed, then a probabilistic model is used for encoding the four symbols $+$, $-$, 0 , and 1 , into a compressed stream of bits [6]. The WDR method described in [1] and [2] uses an arithmetic coding technique to slightly improve compression performance, although it is also possible to use the two-bit encoding of these four symbols described below.

Once the positions of significant transform values have been determined for a certain threshold, then these positions are removed from the scan order. For instance, if the initial positions are $x[1], x[2], \dots, x[256]$, say, and $x[1]$ and $x[12]$ are found to be significant, then the insignificant transform values are $x[2], \dots, x[11], x[13], \dots, x[256]$, which are then mapped to a new set of values $\{\tilde{x}[m]\}$ as follows: $\tilde{x}[1] = x[2], \dots, \tilde{x}[10] = x[11], \tilde{x}[11] = x[13], \dots, \tilde{x}[254] = x[256]$. This has the effect of reducing the lengths of subsequent binary expansions of the number of steps between new significant positions.

The refinement bits generated in the refinement pass are produced via the standard bit-plane quantization procedure. For instance, if an old significant transform value's magnitude lies in the interval $[32, 48)$ say, and the present threshold is 8 , then it will be decided if this magnitude lies in either $[32, 40)$ or $[40, 48)$. In the former case, the bit 0 is generated, while in the latter case, the bit 1 is generated. This amounts to encoding each significant value in a binary expansion using the initial threshold as unit-value. The symbol stream generated during the refinement pass is already naturally encoded in bits, but arithmetic compression can be used to slightly reduce the size of this bit stream.

3 The ASWDR algorithm

Now that we have briefly outlined the WDR algorithm, we can describe the improved version of this algorithm, the ASWDR algorithm. We shall begin with a step-by-step description of ASWDR, and then discuss the rationale behind it.

The ASWDR Method

Step 1. Perform a wavelet transform of the discrete image, $\{f[i, j]\}$, producing the transformed image, $\{\hat{f}[i, j]\}$. In the experiments described below, a Daub 9/7 transform (see [7]) was used.

Step 2. Choose a *scanning order* for the transformed image, which is a one-to-one and onto mapping, $\hat{f}[i, j] = x[k]$, whereby the transform values are scanned through via a linear ordering $k = 1, 2, \dots, M$. In [1] and

[2], the scanning order is a zigzag through subbands from higher-scale to lower-scale [5] with row-based scanning in the lowpass/highpass (horizontal) subbands and column-based scanning in the highpass/lowpass (vertical) subbands.

Step 3. Choose an initial threshold, T , such that at least one transform value, $x[n]$ say, satisfies $|x[n]| \geq T$ and *all* transform values, $x[k]$, satisfy $|x[k]| < 2T$.

Step 4. (Significance pass). Record positions for *new* significant values—i.e., those new indices m for which $x[m]$ has a magnitude greater than or equal to the present threshold. Encode these new significant indices using the difference reduction method.

Step 5. (Refinement pass). Record *refinement bits*, the next significant bits, for the *old* significant transform values (significant transform values determined using larger threshold values). This generation of refinement bits is the standard bit-plane encoding that is employed by all embedded codecs (as described in [5] and [3], and which we briefly summarized above).

Step 6. (New Scanning Order). For the highest-scale level (the one containing the all-lowpass subband), use the indices of the remaining insignificant values as the scan order at that level. Use the scan order at level j to create the new scan order at level $j - 1$ as follows. Run through the significant values at level j in the wavelet transform. Each significant value, called a *parent* value, induces a set of four *child* values as described in the spatial-orientation tree definition in [3]. The first part of the scan order at level $j - 1$ contains the insignificant values lying among these child values. Run through the insignificant values at level j in the wavelet transform. The second part of the scan order at level $j - 1$ contains the insignificant values lying among the child values induced by these insignificant parent values. Use this new scanning order for level $j - 1$ to create the new scanning order for level $j - 2$, until all levels are exhausted.

Step 7. Divide the present threshold by 2. Repeat Steps 4–6 until either a given bit budget is exhausted or a given distortion metric is satisfied.

When decoding, the steps above are recapitulated to produce a quantized output. At the very end, each quantized value is rounded into the midpoint of the quanti-

zation bin that it lies in. During this ASWDR procedure, the symbol output can also be subjected to arithmetic coding in order to achieve further compression.

In practice it was found to be better to skip Step 6 for the first five rounds. The reason for doing this will be provided below, along with the rationale for performing Step 6 after the first five rounds.

This seven-step ASWDR procedure is nothing more than the WDR procedure, but with the addition of the new Step 6 for creation of new scanning orders. We shall now give a justification for performing this step. It does not greatly increase the complexity of the algorithm, since it adds only four comparison operations per pixel (the testing of insignificance of child values for inclusion in the new scanning order) to one-quarter of the total number of pixels. That is an average of one comparison operation per pixel. This is not a substantial increase over the low-complexity of the WDR algorithm.

The reasons for producing this new scanning order are to (1) use a scan order that reflects the correlation between significant parent transform values and significant child transform values, and (2) to force insignificant values lying in zerotrees to the back of the scanning order at each level. The aim is to reduce the number of steps between significant transform values, thereby decreasing the lengths of the symbol strings needed for encoding these distances. In [8] it is shown that there is a high correlation between significant transform values, whose magnitudes are at least T , and significant child transform values, whose magnitudes are at least $T/2$. Figure 4B in [8] provides a good illustration of this correlation. That figure shows a conditional histogram for fine scale horizontal subband transform values from the Boats test image. The conditional histogram is of $\log_2 |C|$ (base 2 log of child magnitudes) versus $\log_2 |P|$ (base 2 log of parent magnitudes). It is clear from the figure that a large percentage of child magnitudes are above $\log_2 |P| - 1$, i.e., are either significant at the present threshold (hence already coded) or will be significant at the next threshold (hence should be scanned first).

In Fig. 1, we provide an illustration of this correlation. This figure was obtained from a Daub 9/7 wavelet transform of the Lena image. Figure 1(a) depicts the insignificant child values (shown in white) in the 1st level vertical subband of significant parent values in the 2nd level vertical subband, when the threshold is 32. Figure 1(b) depicts the *new* significant values for the half-threshold—those whose magnitudes are less than 32 and greater than or equal to 16—in the 1st vertical subband. Notice that the child locations in Fig. 1(a) are good predictors for the new significant values in Fig. 1(b). Although these predictions are not perfectly accurate, there is a great deal of overlap between the two images (in fact, the fraction of new significant values that lie within the first part of the scan order

created by Step 6 is 0.58). By putting these child locations into the first part of the new scanning order at each level, the number of steps between them should be reduced. In fact, if a prediction is correct, then the output specifying that location will consist of only the sign of the new significant value.

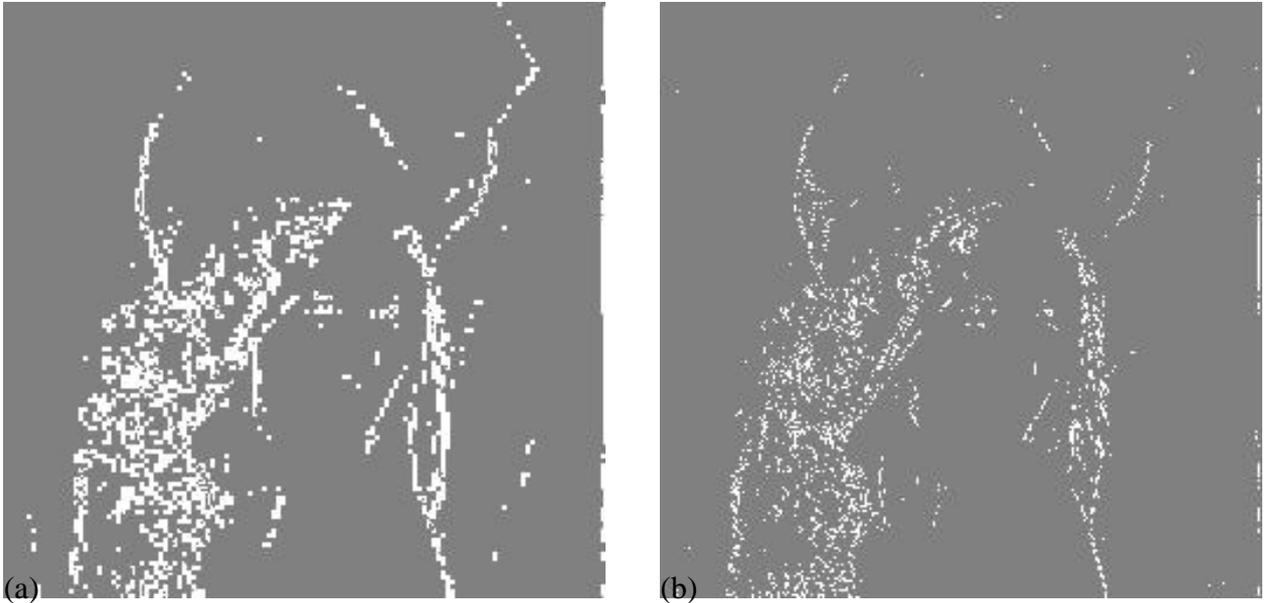


Figure 1: (a) Insignificant children in the 1st vertical subband having significant parents in the 2nd vertical subband when the threshold is 32. (b) New significant values in the 1st vertical subband when the threshold is decreased to 16.

Notice also how the locations of significant values are highly correlated with the location of edges in the Lena image. The scanning order of ASWDR dynamically adapts to the locations of edge details in an image, and this enhances the resolution of these edges in ASWDR compressed images.

We now present some statistics for estimating the conditional probability $P(1^{\text{st}} | \text{new})$, defined by

$$P(1^{\text{st}} | \text{new}) = \text{Prob}(\text{new sig. value in } 1^{\text{st}} \text{ part of scan} | \text{new sig. value}), \quad (1)$$

of a new significant value being found within the first part of the new scan order (for a fixed level) created by ASWDR. In Table 1 we give the fraction of new significant values being found in the first part of the new scan order at several levels for four test images and for a random noise image. The data in this table clearly show that the probability in Eq. (1) is much greater for high-magnitude threshold values for the test images than for the random noise image. (By high-magnitude values we mean those values whose magnitudes are greater than the standard deviation of the child subband.) Such high probabilities justify scanning through child values of significant parents first. It should be noted that probabilities between 0.3 and 0.6 only begin to be realized at thresholds below 256. Such thresholds occur, for the images used, after the fifth pass through the loop in the ASWDR procedure. That is the justification for skipping Step 6 in the ASWDR procedure the first five times through.

Notice that, for the random noise only, it is unlikely that a newly significant child value will be found in the first part of the scan order when the threshold is greater than the standard deviation for the child subband. This facilitates the separation of noisy transform values from image transform values at much lower thresholds than is possible with the standard wavelet thresholding methods. This new denoising methodology is described in [9].

The new scanning order step in the ASWDR procedure also assigns indices for insignificant children of insignificant parents to the second part of the new scanning order. This is an attempt to exploit the prevalence of zerotrees in wavelet transforms. The prevalence of zerotrees in wavelet transforms of natural images is well-known ([5] and [3]). Since significant transform values will never be found in zerotrees, each new scanning order aims to force the components of zerotrees to the ends of the scanning order at each level. The goal is to reduce the number of steps between those significant transform values whose parents are insignificant by forcing those values toward the beginning of the scan order.

In the paragraphs above, we have given an *a priori* justification for creating a new scanning order. We now give an *a posteriori* justification. We present some data which show that the new scanning order step in ASWDR does allow it to encode more transform values than WDR. The only difference between ASWDR and WDR is in the predictive scheme employed by ASWDR to create new scanning orders. Consequently, if ASWDR typically encodes more values than WDR does, then this must be due to the success of the predictive scheme. In Table 2, there is a comparison of the number of values encoded by WDR and ASWDR for four test images at different compression ratios. The ASWDR and WDR methods both used the Daub 9/7 wavelet transform with 7 subband levels. The values for

Parent level/Threshold	512	256	128	64	32	16	8
Lena , 4 th , $\sigma = 37$	(0.30)	0.37	0.46	0.57	0.66	0.68	0.68
Lena , 3 rd , $\sigma = 15$		(0.07)	0.31	0.50	0.56	0.55	0.50
Lena , 2 nd , $\sigma = 19$			(0.95)	0.51	0.54	0.49	0.38
Barbara , 4 th , $\sigma = 38$	(0.06)	0.43	0.54	0.60	0.63	0.68	0.77
Barbara , 3 rd , $\sigma = 22$		0.01	0.09	0.26	0.38	0.51	0.60
Barbara , 2 nd , $\sigma = 12$			0.03	0.21	0.37	0.51	0.51
Goldhill , 4 th , $\sigma = 34$	(0.00)	0.31	0.42	0.48	0.59	0.69	0.68
Goldhill , 3 rd , $\sigma = 15$		(0.04)	0.24	0.37	0.45	0.54	0.65
Goldhill , 2 nd , $\sigma = 6$			(0.07)	0.32	0.35	0.40	0.46
Airfield , 4 th , $\sigma = 66$	0.21	0.36	0.46	0.56	0.61	0.76	0.86
Airfield , 3 rd , $\sigma = 28$	(0.00)	0.29	0.39	0.46	0.50	0.76	0.71
Airfield , 2 nd , $\sigma = 10$		(0.14)	0.30	0.43	0.43	0.38	0.54
Noise , 4 th , $\sigma = 42$			0.01	0.18	0.50	0.74	0.86
Noise , 3 rd , $\sigma = 44$			0.01	0.19	0.52	0.74	0.88
Noise , 2 nd , $\sigma = 43$		0.00	0.01	0.21	0.54	0.76	0.88

Table 1: Fraction of new significant values captured by first part of the new scan order created by ASWDR. The standard deviations σ are for the child subbands of each level. The Noise image was created by wavelet transforming a simulation of Gaussian random noise with mean 0 and standard deviation 48. A fraction in parentheses indicates that it may be an unreliable estimate of the conditional probability $P(1^{\text{st}} | \text{new})$ in Eq. (1), due to the number of new significant values being too small (less than 200).

ASWDR and WDR were obtained without arithmetic coding. The coding used was a *binary* encoding. In other words, the significance symbols 0, 1, +, and – were encoded as the two-bit strings 00, 01, 10, and 11, respectively, and the refinement bits 0 and 1 were left as is, and all these bits were then packed together as 8-bit bytes.

The data in Table 2 shows that, in almost every case, ASWDR encodes more values than WDR. This provides an *a posteriori* justification for adding the new scanning order step to the WDR procedure.

4 Experimental results

In this section we shall compare ASWDR with WDR and with one of the best image codecs—one that gives nearly the highest PSNR¹ values over a wide range of test images—the SPIHT method of [3]. We used the test images, Lena, Goldhill, Barb, and Airfield [10].

We begin with PSNR values for ASWDR, WDR, and SPIHT. In Table 3 we show the results of using ASWDR and SPIHT without arithmetic coding. Both ASWDR and WDR used a 7-level Daub 9/7 wavelet transform. The distortion measure used in the table is PSNR, which although it is not always reliable as a gauge of subjective visual quality, has become a defacto standard for comparing codecs. We shall report below the results for an edge correlation measure, which seems to be more faithful to the perceptual quality of compressed images.

The results in Table 3 show that SPIHT and ASWDR perform at essentially the same level when arithmetic compression is not employed. There are situations where the need for speed is critical and the use of arithmetic compression is avoided in such cases. When arithmetic compression is not employed, the advantages of ASWDR—such as its ROI capability and its ability to carry out image processing operations on compressed data—make it a worthwhile alternative to the SPIHT method.

When arithmetic compression is employed, the SPIHT method performs better than the ASWDR and WDR methods in terms of the PSNR metric. This is shown by the results in Table 4. The differences in PSNR between ASWDR and SPIHT are relatively small, however, and may only reflect the fact that ASWDR uses a relatively unsophisticated model for arithmetic coding. It employs a context-1

¹The *Peak Signal to Noise Ratio*, PSNR, in decibels is $10 \log_{10}(255^2/\|f - g\|_2^2)$, where f and g are the original and compressed images.

Image\Method	WDR	ASWDR	% increase
Lena, 0.25 bpp	10,450	11,105	6.3%
Lena, 0.5 bpp	20,809	22,370	7.5%
Goldhill, 0.25 bpp	10,410	10,210	-1.9%
Goldhill, 0.5 bpp	22,905	23,394	2.1%
Barbara, 0.25 bpp	11,681	12,174	4.2%
Barbara, 0.5 bpp	23,697	24,915	5.1%
Airfield, 0.25 bpp	10,519	11,228	6.7%
Airfield, 0.5 bpp	19,950	21,814	9.3%

Table 2: Comparison of numbers of values encoded by WDR and ASWDR.

Image\Method	SPIHT	ASWDR	WDR
Lena, 0.25 bpp	33.51	33.44	33.22
Goldhill, 0.25 bpp	30.15	30.17	30.20
Barbara, 0.25 bpp	27.07	26.92	26.77
Airfield, 0.25 bpp	25.50	25.46	25.28
Lena, 0.5 bpp	36.71	36.52	36.27
Goldhill, 0.5 bpp	32.66	32.62	32.47
Barbara, 0.5 bpp	30.82	30.74	30.53
Airfield, 0.5 bpp	28.05	28.03	27.76

Table 3: PSNR values for three methods, *without* arithmetic compression.

model [11] for encoding the four significance symbols. That is, four separate histograms are used for encoding each symbol based on its frequency of occurrence following one of the four possible previous symbols. For encoding the two refinement bits, ASWDR simply uses a single histogram (i.e., a context-0 model [11]). SPIHT, on the other hand, employs a more sophisticated coding procedure based on spatial context modeling [12].

PSNR values, however, do not tell the whole story. We now turn to an examination of some compressed images done with each algorithm.

In Fig. 2, we show compressions of the Lena image at 0.5 bpp (16:1). It is difficult to observe any differences at all between any of these images. This illustrates that all three compression methods produce equally good compressions at moderately high bit rates.

Some differences between compressed images do appear at lower bit rates. For example, in Fig. 3 we show 0.25 bpp (32:1) compressions of the Barbara image. The ASWDR and WDR images are perceptually superior to the ones produced by SPIHT. The SPIHT compression has distorted Barbara's left eye and erased large parts of the striping on the tablecloth. The ASWDR and WDR compressions are more difficult to distinguish. The ASWDR image does preserve some fine details which WDR erases or distorts, we show this in the magnified images in Fig. 4. The ASWDR compression in Fig. 4(b) has retained more of the striping in the tablecloth than the WDR compression in Fig. 4(a). At the bottom corner of the tablecloth, the striping in the ASWDR compression is in the proper orientation, while in the WDR compression it has lost much of its coherence (and appears as slightly thicker stripes with a different orientation from the original). The toy track is also slightly better preserved by the ASWDR compression.

As another example, we examine compressions of the Airfield image at a very low bit rate. In Fig. 5 we show compressed images of Airfield at 0.0625 bpp (128:1). The WDR and ASWDR methods both preserve more of the fine details in the image. At the top of its image in Fig. 5(c), SPIHT erases many fine details such as the telephone pole and two small square structures to the right of the thin black rectangle. These details are preserved, at least partially, by both WDR and ASWDR. The ASWDR image does the best job retaining some structure in the telephone pole. ASWDR is also best at preserving the outline of the swept-back winged aircraft, especially its thin nose, located to the lower left of center.

Of course, a few images are not sufficient for drawing broad conclusions. Clearly, some objective measure of the preservation of details is needed. Observers have commented that WDR and ASWDR compressed images appear sharper than SPIHT compressed images. Reasoning that human visual systems focus on



(a)



(b)



(c)



(d)

Figure 2: Compressions of Lena at 0.5 bpp. (a) WDR compression. (b) ASWDR compression. (c) SPIHT compression. (d) Original (8 bpp).



(a)



(b)



(c)



(d)

Figure 3: Compressions of Barbara at 0.25 bpp. (a) WDR compression. (b) ASWDR compression. (c) SPIHT compression. (d) Original (8 bpp).



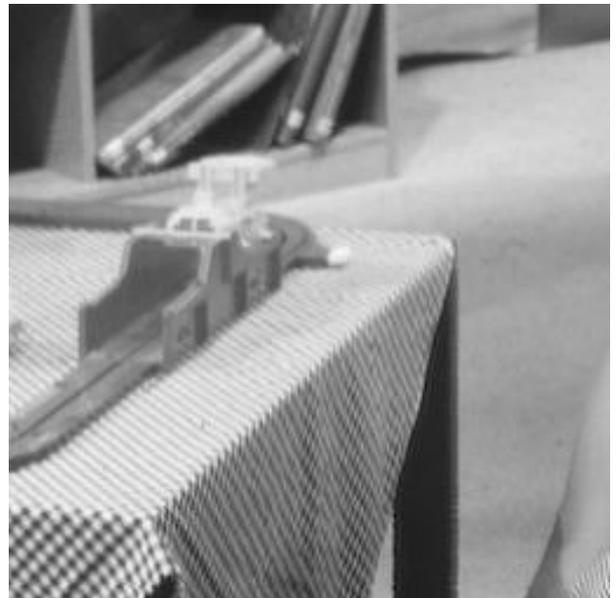
(a)



(b)



(c)



(d)

Figure 4: Magnifications of Barbara compressions at 0.25 bpp. (a) WDR compression. (b) ASWDR compression. (c) SPIHT compression. (d) Original (8 bpp).



(a)



(b)



(c)



(d)

Figure 5: Compressions of Airfield at 0.0625 bpp. (a) WDR compression. (b) ASWDR compression. (c) SPIHT compression. (d) Original (8 bpp).

edges when analyzing images, the following method was used to produce an image that retains only edges. First, a 3-level Daub 9/7 transform of an image is created. Second, the all-lowpass subband is subtracted away from this transform. Third, an inverse transform is performed on the remaining part of the transform. This produces a highpass filtered image, which exhibits edges from the original image. A similar highpass filtered image is created from the compressed image. Both of these highpass filtered images have mean values that are approximately zero. We define the *edge correlation* γ_3 by

$$\gamma_3 = \frac{\sigma_c^2}{\sigma_o^2}$$

where σ_c^2 denotes the variance of the values of the highpass filtered version of the compressed image, and σ_o^2 denotes the variance of the values of the highpass filtered version of the original image. Thus γ_3 measures how well the compressed image captures the variance of edge details in the original image.

Using this edge correlation measure, we obtained the results shown in Tables 5 and 6. In every case, the ASWDR and WDR compressions have higher edge correlations than the SPIHT compressions. The ASWDR edge correlations are, in almost every case, slightly higher than the WDR edge correlations. These numerical results are consistent with the increased preservation of details within ASWDR and WDR images. Notice that, for the magnified image of Barbara, the edge correlations in Table 6 show that ASWDR is clearly superior to both WDR and SPIHT (and WDR is superior to SPIHT as well).

For the airfield compressions in Fig. 5, we show in Table 7 the values for three different edge correlations γ_k , $k = 3, 4$, and 5 . The variable k equals the number of levels in the Daub 9/7 wavelet transform retained by the high-pass filtering. A higher value of k means that edge details at lower resolutions were used in computing the edge correlation γ_k . The edge correlations in Table 7 show that ASWDR is best, at the very low bit rate of 0.0625 bpp, in preserving edges in the airfield image over several resolution levels.

The results described in this section show that ASWDR is a promising new method, which yields improved preservation of details at low bit rates. Preserving details at low bit rates is a vital property for applications such as remote medical diagnosis via rapid transmission of compressed images, and fast search/retrieval of images in databases.

Image\Method	SPIHT	ASWDR	WDR
Lena, 0.25 bpp	33.93	33.64	33.39
Goldhill, 0.25 bpp	30.49	30.34	30.33
Barbara, 0.25 bpp	27.47	27.03	26.87
Airfield, 0.25 bpp	25.90	25.64	25.49
Lena, 0.5 bpp	37.09	36.64	36.45
Goldhill, 0.5 bpp	33.10	32.85	32.70
Barbara, 0.5 bpp	31.29	30.87	30.68
Airfield, 0.5 bpp	28.57	28.36	28.12

Table 4: PSNR values for three methods, *with* arithmetic compression.

Image/Method	ASWDR	WDR	SPIHT
Lena, 0.5 bpp	0.96	0.95	0.93
Lena, 0.25 bpp	0.90	0.89	0.87
Goldhill, 0.5 bpp	0.93	0.92	0.85
Goldhill, 0.25 bpp	0.76	0.76	0.71
Barbara, 0.5 bpp	0.92	0.91	0.87
Barbara, 0.25 bpp	0.81	0.80	0.74
Airfield, 0.5 bpp	0.88	0.88	0.85
Airfield, 0.25 bpp	0.77	0.76	0.73

Table 5: Edge correlations, *with* arithmetic compression.

Method/Measure	Edge Corr.
ASWDR	0.81
WDR	0.77
SPIHT	0.69

Table 6: Edge correlations for magnifications of Barbara in Fig. 4.

5 Conclusion

In this paper it was shown that an adaptively scanned wavelet difference reduction method yields an embedded codec, comparable in performance to the widely used SPIHT codec, but which has ROI capability and is better at preserving edge details. Future work will aim to improve the prediction procedure used to adaptively construct new scanning orders. This work should involve a weighted, linear predictor as described in [8].

The tree-based nature of ASWDR applies to uniform subband transforms. Hence it should be adapted to such transforms, in order to produce better performance for images like *Barb* and *Goldhill*.

The present version of ASWDR, like SPIHT, is memory intensive. Further research is needed to produce a block wavelet transform version of ASWDR. Such a block transform should yield an accompanying *localized* denoising procedure.

Acknowledgements

The author would like to thank the University of Wisconsin for its generous support, and the Boston University Departments of Mathematics and of Electrical & Computer Engineering for their hospitality, during my 1998–99 sabbatical year when this paper was written. He also wishes to thank Truong Nguyen for his help and encouragement.

References

- [1] J. Tian and R.O. Wells, Jr. A lossy image codec based on index coding. *IEEE Data Compression Conference, DCC '96*, page 456, 1996.
- [2] J. Tian and R.O. Wells, Jr. Embedded image coding using wavelet-difference-reduction. *Wavelet Image and Video Compression*, P. Topiwala, ed., pp. 289–301. Kluwer Academic Publ., Norwell, MA, 1998.
- [3] A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 6, No. 3, pp. 243–250, June 1996.

- [4] J. Tian and R.O. Wells, Jr. Image data processing in the compressed wavelet domain. *3rd International Conference on Signal Processing Proc.*, B. Yuan and X. Tang, Eds., pp. 978–981, Beijing, China, 1996.
- [5] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Proc.*, Vol. 41, No. 12, pp. 3445–3462, Dec. 1993.
- [6] I.H. Witten, R. Neal, and J.G. Cleary. Arithmetic coding for data compression. *Commun. of the ACM*, Vol. 30, No. 6, pp. 520–540, June 1987.
- [7] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Commun. on Pure and Appl. Math.*, Vol. 45, pp. 485-560, 1992.
- [8] R.W. Buccigrossi and E.P. Simoncelli, Image compression via joint statistical characterization in the wavelet domain. *IEEE Trans. on Image Proc.*, Vol. 8, No. 12, pp. 1688–1701, 1999.
- [9] J.S. Walker. Image denoising using tree-based wavelet subband correlations and shrinkage. Submitted to *Optical Engineering*.
- [10] For Lena, Goldhill, and Barbara, go to the website:

`ftp://ipl.rpi.edu/pub/image/still/usc/gray/.`

For Airfield, go to the website:

`http://www.image.cityu.edu.hk/imagedb/.`
- [11] T.C. Bell, J.G. Cleary, and I.H. Witten. *Text Compression*. Prentice-Hall PTR, Englewood Cliffs, NJ, 1990.
- [12] A. Said and W.A. Pearlman. An image multi-resolution representation for lossless and lossy image compression. *IEEE Trans. Image Proc.*, Vol. 5, No. 9, pp. 1303–1310, 1996.

Corr./Method	ASWDR	WDR	SPIHT
γ_3	0.51	0.48	0.44
γ_4	0.68	0.67	0.61
γ_5	0.78	0.77	0.71

Table 7: Edge correlations for 128:1 compressions of Airfield image.