Comparison of the JPEG2000 lossy image compression algorithm with WDR-based algorithms

James S. Walker[*†] Ying-Jui Chen[‡] Tarek M. Elgindi[*†]

walkerjs@uwec.edu yrchen@mit.edu elgindtm@uwec.edu

[*]Department of Mathematics; University of Wisconsin–Eau Claire; Eau Claire, WI 54702–4004; Phone: 715–836–3301; Fax: 715–836–2924.

[†]JSW and TME express their thanks for a UWEC ORSP grant which partially supported our research.

[‡]MediaTek, Inc.; No. 1–2, Innovation Road 1; Science-Based Industrial Park; Hsin-Chu City, Taiwan 300, R.O.C.; Phone: +886-3-567-0766 ext. 21384.

Abstract

We compare the new still-image compression standard, JPEG2000, with two competing algorithms based on the Wavelet Difference Reduction (WDR) technique of encoding. Our comparison shows that these WDR methods provide far simpler encoding while also providing essentially the same performance as JPEG2000 at moderately high to very high compression ratios. They provide useful alternative algorithms for applications where high compression ratios are needed and/or further processing of the compressed images is needed.

Keywords: image compression; signal processing; WDR compression; JPEG2000 compression.

Introduction

The JPEG2000 image compression standard [1] is a highly sophisticated algorithm for compression of still images which replaces the now-outmoded JPEG algorithm [2]. JPEG2000, however, may not be suitable for all applications, due to its incorporation of a large number of features which generates a very complicated bitstream. As we will show, there are far simpler algorithms for image compression based on the Wavelet Difference Reduction algorithm [3], [4]. These WDR-based algorithms are described in great detail in [5] and [6]. *The simplicity of the WDR method allows for ease of modification for specific applications* (for example the data processing application described in [7]). Here we report our findings that (1) WDR-based methods provide nearly equivalent performance (using the standard Peak Signal to Noise Ratio [PSNR] for error measurements) as JPEG2000 at medium compression ratios (we tested 40:1), (2) when high compression ratios are used (we tested 80:1 and 160:1), WDR-based methods equal JPEG2000. These two features highlight the importance of using WDR-based algorithms for some image compression applications—*an example of Ockham's Razor applied to engineering science.*

The paper is organized as follows. In section 1, we summarize the main features of the lossy JPEG2000 algorithm; we shall be very brief as JPEG2000 is described in complete detail in [1]. Section 2 contains descriptions of the WDR-based algorithms that we employed. As mentioned above, these algorithms are described in detail in [5] and [6], we only provide a brief summary in section 2. These WDR-based algorithms are two important variants of the original WDR algorithm which we will show are useful alternatives to JPEG2000. The first variant is an algorithm which performs equally as well as the original WDR algorithm but *does not use arithmetic compression*—dropping the arithmetic compression portion of the original WDR algorithm further simplifies the WDR method without sacrificing performance, *hence providing a very fast implementation that would be useful for applications such as video coding* (where JPEG2000, for example, would be too slow). The second variant is a low-memory and low-power version of WDR which, while not as effective as the first variant, nevertheless nearly equals the performance of JPEG2000 at very high compression ratios (we tested 160:1) and yet employs very little RAM (less even than JPEG2000, because it employs 16-bit integer values for its wavelet transform instead of the 32-bit floating point values used by JPEG2000). In section 3, which is the heart of the paper, we provide objective comparisons (using PSNR values) on several test images at various compression ratios and also provide subjective (visual) comparisons. The paper concludes with a brief analysis/summary and some directions for future research.

1 JPEG2000 briefly summarized

We shall briefly summarize the JPEG2000 algorithm in order to clarify the need for alternative compression algorithms such as the WDR-based methods described in the next section. Full details are given in [1]. From this point on, to save space, we shall refer to JPEG2000 as J2K.

The J2K lossy compression algorithm consists of the following four steps: (1) a Daub 9/7 wavelet transform of the image (*Warning:* some implementations of J2K use the Daub 5/3 transform for lossy as well as lossless compression, but its' performance is significantly worse for the lossy case.); (2) embedded encoding of block subdivisions of the wavelet transform; (3) optimal truncation of transform values for optimizing PSNR values within the constraint of a given bit-budget, using quality layers and fractional bit-plane encoding; (4) sophisticated arithmetic encoding, using multiple contexts, of the raw binary output generated by step 3. It is important to emphasize that the block subdivisions in step 2 are employed in order to make the J2K algorithm useful for very large images. While this block-subdividing might have entailed a performance sacrifice, it is very effectively compensated for by steps 3 and 4; although at the cost of creating a complex encoding process and a very complex (richly structured) bitstream.

For applications where memory is not an issue and high compression ratios are needed, we shall see that WDR-based method can nearly match or equal the performance of J2K. An important issue is that *steps 3 and 4 of J2K generate a very complex bitstream, which is not desirable for applications where one needs to process the bitstream.* An advantage of the two variants of the original WDR algorithm, described in the next section, is that they generate very simple bitstreams which facilitates processing in applications.

2 WDR methods briefly summarized

In this section, we discuss two variants of the WDR algorithm of Tian and Wells. From this point on, we shall denote their WDR algorithm by WDR-AC, which indicates that arithmetic coding is employed in this original WDR algorithm. Variant 1, which we shall denote as WDR-I, is a WDR-based algorithm which drops the arithmetic coding, but is nevertheless essentially equivalent in performance to WDR-AC. WDR-I has been described in detail in [5], here we provide only a brief summary and a few clarifying remarks for the convenience of our readers. Variant 2 uses a 16-bit integer-valued wavelet transform, instead of the 32-bit floating point transform used by WDR-AC and WDR-I, and also employs a *block-subdivision* scheme that removes the requirement of keeping the whole wavelet transform in RAM. This second variant, which we shall denote as WDR-B, is useful for high-compression of large images (for example, Synthetic Aperture Radar images), and matches the performance of J2K at high compression ratios. WDR-B has been thoroughly described in [6], so again we provide just a brief summary and some clarifying remarks.

2.1 WDR-I

The WDR-I method is an improved (simplified) modification of WDR-AC, which does not employ arithmetic coding. This method is described fully in [5]; *an extract of the material describing WDR-I from that reference can be downloaded from the webpage listed in [8].* WDR-I lossy image compression consists of the following steps: (1) Perform a wavelet transform (we used a 6-level Daub 9/7 transform) of the image; (2) choose an initial threshold T_0 equal to the smallest power of 2 that is greater than the maximum of all transform magnitudes [output T_0 at the beginning of the encoding process]; (3) perform *bit-plane encoding* with a significance pass and refinement pass of transform values (exiting when a bit-budget constraint is met) using row-wise scans through horizontal subbands, column-wise scans through vertical subbands, and zig-zag scans through diagonal subbands and the all-lowpass subband (and scan from the largest level scale to the smallest level scale starting with the all-lowpass subband and then vertical, then horizontal, then diagonal subbands within each successive level). The bit-plane encoding process of the significance pass encodes the binary expansions of the number of steps in the scan order between the last encoded significant value and the next encoded significant value (for a given threshold), replacing the most significant bit (which is always 1) by the sign of the next significant value. This significance pass is described in detail in [5] with these two additions/corrections: (1) the output of signs and bits is done with a set of four binary codings ($0 \mapsto 00$, $1 \mapsto 01$, $+ \mapsto 10$, $- \mapsto 11$); (2) the scan order is implemented *in the computer code* no list is kept of significant position locations. The refinement pass simply consists of emitting bits that describe whether each old significant value, computed at a higher threshold, needs to be refined to the accuracy of the new threshold (1 if refinement is needed, 0 if it is not). This standard bit-plane refinement procedure is also described in detail in [5].

WDR-I decompression uses the binary output of the compression process described above to reconstruct an approximate wavelet transform (including a simple estimation procedure of rounding to bin-midpoints described in [5]) and then performs an inverse wavelet transform to obtain the decompressed image.

2.2 WDR-B

The WDR-B method is essentially the same as WDR-I except that a block sub-division scheme is used to greatly reduce the quantity of wavelet transform values that are held in RAM, and a 4-level Daub 5/3 integer-to-integer wavelet transform is used so that an 8-bit integer-valued image can be converted to a 16-bit integer-valued transform (unlike the Daub 9/7 wavelet transform which employs 32-bit floating point values). The WDR-B method is described completely in [6]; we will not repeat that discussion here. *An updated version of that paper (containing corrections of broken webpage links) can be downloaded from [8].*

3 Comparisons

We did our comparisons on five test images: Airfield, Boats, Goldhill, Lena, and Peppers. These are all 512×512, 8-bit grey-scale images—reflecting our emphasis on images that are easily handled by machines with relatively high amounts of RAM, as in PCs, for example. The five images can be downloaded and/or viewed at the webpage [8]. Links for downloading the software employed for our compressions/decompressions can be found at that same webpage.

We compared these images both objectively (using PSNR values) and subjectively (visual comparisons) at four compression ratios (20:1, 40:1, 80:1, 160:1) which span a range from moderate to very high compression. The decompressions for the three methods compared (J2K, WDR-I, WDR-B) were very difficult to visually distinguish for the full images, hence the visual comparisons provided below will be for various 2×2 magnifications of regions selected from the decompressions.

3.1 20:1 compressions

In Table 1, we report PSNR values for decompressions of 20:1 compressions of our test images. These data show the following three features:

1. The PSNRs for WDR-I and WDR-AC are *essentially equivalent:* For the Airfield image, WDR-AC has a somewhat higher PSNR; on the other hand, for the Peppers image WDR-I has a somewhat higher PSNR; while for the remaining images the differences in PSNRs are negligible.

2. For J2K compared with WDR-I, we see that J2K significantly outperforms WDR-I. That is to be expected, since J2K was specifically designed for obtaining optimal PSNRs at low to moderate compression ratios like 20:1. We shall see below, however, that J2K and WDR-I are essentially equal in performance at higher compression ratios.

3. WDR-B shows significantly lower PSNR performances, relative to J2K and WDR-I. We shall see in later subsections, however, that WDR-B's performance becomes more similar to these other two methods as the compression ratio increases.

We now provide a visual comparison. In Figure 1, we show four 2×2 magnifications of a portion of the Boats image and corresponding decompressions. These images appear to be consistent with the local PSNR values reported in Table 2, which show J2K to be clearly superior to both WDR-I and WDR-B, and WDR-I superior to WDR-B. *Our emphasis on WDR-based methods,* *however, is predicated on their performance at medium to very high compression ratios* (we tested 40:1, 80:1, 160:1) and we now turn to comparisons at those compression ratios.

3.2 40:1 compressions

In Table 3, we report PSNR values for decompressions of 40:1 compressions of our test images. These data show the following three features:

1. Again, as for 20:1, the PSNRs of WDR-I and WDR-AC are essentially equivalent. The entries marked NA for WDR-AC are apparently due to a bug in the WDR-AC software that we downloaded from Tian's Webpage [9].

2. The PSNRs for J2K and WDR-I are nearly equal; although J2K produces higher PSNRs, the differences are slight (less than 0.4 dB for every image but Boats).

3. The PSNRs for WDR-B are still significantly lower than for the other methods, but the gaps are narrowing from what they were at 20:1.

For a visual comparison, in Figure 2 we show 2×2 magnifications of a portion of the Lena image. One noticeable feature that is better preserved by both WDR-I and WDR-B, versus J2K, is the structure of Lena's eyes, especially her pupils and irises, and her eyelashes. Furthermore, the tiny flower in Lena's hat (at the middle left, just above the feathers) seems best preserved by WDR-I. But, J2K seems to better preserve the shape of Lena's nose than WDR-I (but not WDR-B).

In Table 2, we see that the local PSNRs show WDR-I to be just slightly better than J2K, and J2K just slightly better than WDR-B. The images in Figure 2 are in correspondence with these objective values.

3.3 80:1 compressions

In Table 4, we report PSNR values for decompressions of 80:1 compressions of our test images. These data show the following three features:

1. WDR-AC and WDR-I are again more or less equal in performance.

2. WDR-I and J2K have essentially equal PSNRs. This result is interesting in that WDR-I employs a very simple bitstream (easily modified for applications), which these data show provides an important alternative to J2K for high compression ratios (we shall see similar results at 160:1 as well).

3. WDR-B has nearly equal PSNRs as J2K (within about 0.5 dB). This is a striking result in that WDR-B *does not employ arithmetic coding*, and uses a low-power, low-memory transform (a 16-bit integer-to-integer Daub 5/3 transform, see [6]) instead of the 32-bit floating point Daub 9/7 transform employed by J2K. (For similar results showing essential equivalence, even superiority, of the integer-to-integer Daub 5/3 transform to the floating point Daub 9/7 transform at high compression ratios, see [6].)

In Figure 3, we show 2×2 magnifications of the Peppers image. The WDR-I and J2K images appear to be very similar, with perhaps slightly less artifacts in the WDR-I image; while the J2K image is slightly less blurred but suffers from more artifacts than the WDR-B image. The visual results here are consistent with the local PSNRs recorded in Table 2.

3.4 160:1 compressions

In Table 5, we report PSNR values for decompressions of 160:1 compressions of our test images. These data show the following three features:

1. Again, as with the other compression ratios, WDR-AC and WDR-I are more or less equal in performance.

2. As with the 80:1 case, at this high compression ratio of 160:1, WDR-I has essentially equal PSNRs as J2K for all images.

3. WDR-B has nearly equal PSNRs as J2K, even though WDR-B uses a much simpler bitstream than J2K (for reasons mentioned above in item 3 of the 80:1 subsection).

In Figure 4, we show 2×2 magnifications of a portion of the Airfield image. Notice that WDR-I best preserves the structure of the row of airplanes on the left half of the image. The WDR-I and WDR-B images are roughly visually equivalent in the sense that the WDR-B image preserves more details of some structures but also suffers from more artifacts. We also note that the WDR-B image better preserves some details (such as the swept-wing aircraft at upper-left and some of the rectangular structures at upper-right) than the J2K image does. The WDR-B image also appears slightly sharper than for the J2K and WDR-I images; this may be due to the shorter filters that WDR-B employs. In any case, the local PSNRs in Table 2 show that all three methods are nearly equal in performance, which is consistent with the images shown in Figure 4.

Conclusion

Our comparison has yielded the following results:

1. *WDR-AC, which uses arithmetic coding, and WDR-I, which does not use arithmetic coding, yield essentially the same performance at every compression ratio for all images.* This may be due to the fact that WDR-AC, as described in [3] and [4], employs an unnecessary refinement pass at the first stage of bit-plane encoding.

2. *WDR-I performs more or less equally well with J2K at moderately high to very high compression ratios.* Because WDR-I holds the entire wavelet transform in RAM, it is only suitable for small to moderate size images (the kind we tested on); nevertheless, such images are frequently handled by PC's, and hence we can see a large area of application for WDR-I.

3. *At high compression ratios, WDR-B performs as well as J2K.* This may be due to the fact that at high compression ratios both WDR-B and J2K are encoding just a few blocks, and

also because J2K encodes significance and refinement data for each block individually while WDR-B encodes significance data for all blocks at each bit-plane followed by an encoding of refinement data for all blocks at each bit-plane. In any case, the simplicity of the bit-stream of WDR-B recommends it as an alternative to J2K for applications that are constrained by low-memory and/or low-bandwidth as well as requiring high compression ratios (such as compressing SAR images or encoding multiple frame surveillance images).

4. *Using global and local PSNR values gave objective results that correlate reasonably well with subjective visual comparisons.* There is room for improvement, however. In the future, we intend to investigate other objective measures of image quality—such as the edge correlation measure used in [10] and [5]—and the WAPE (Weighted Average of PSNR and Edge correlation) measure introduced in [11]—to see how well they correlate with visual judgements of image quality.

References

[1] D.S. Taubman and M.W. Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice.* Kluwer, Boston, MA, 2002.

[2] W.B. Pennebaker and J.L. Mitchell. *JPEG Still Image Compression.* Chapman and Hall, London, 1992.

[3] J. Tian and R.O. Wells, Jr., “A lossy image codec based on index coding.” *IEEE Data Compression Conference, DCC ’96,* page 456, 1996.

[4] J. Tian and R.O. Wells, Jr., “Embedded image coding using wavelet-difference-reduction.” In *Wavelet Image and Video Compression,* P. Topiwala, ed., pp. 289–301. Kluwer, Norwell, MA, 1998.

[5] J.S. Walker and T.Q. Nguyen, “Wavelet-based image compression.” Chap. 6 in *Handbook of Transforms and Data Compression,* K.R. Rao and P. Yip, Eds., pp. 267–312, CRC Press, Boca Raton, 2001. (An extract of this paper is available for downloading from [8].)

[6] J.S. Walker, T.Q. Nguyen, and Y.-J. Chen, “A Low-power, Low-memory System for Wavelet-based Image Compression.” *Optical Engineering, Research Signposts,* **5**, 111–125, (2003). (A corrected version of this paper is available for downloading from [8].)

[7] J. Tian, 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.

[8] `http://www.uwec.edu/walkerjs/CompJ2KWDR`

[9] `http://www.ece.rice.edu/˜juntian/software.html`

[10] J.S. Walker, "Lossy image codec based on adaptively scanned wavelet difference reduction." *Optical Eng*, **39**(7), 1891–1897, (2000).

[11] J.S. Walker, "Wavelet-based image processing." To appear in *Journal of Applicable Analysis*.

(a) Original

(b) J2K

(c) WDR-I

(d) WDR-B

Figure 1: Four 2×2 magnifications of decompressions at 20:1 of Boats image.

(a) Original

(b) J2K

(c) WDR-I

(d) WDR-B

Figure 2: Four 2×2 magnifications of decompressions at 40:1 of Lena image.

(a) Original

(b) J2K

(c) WDR-I

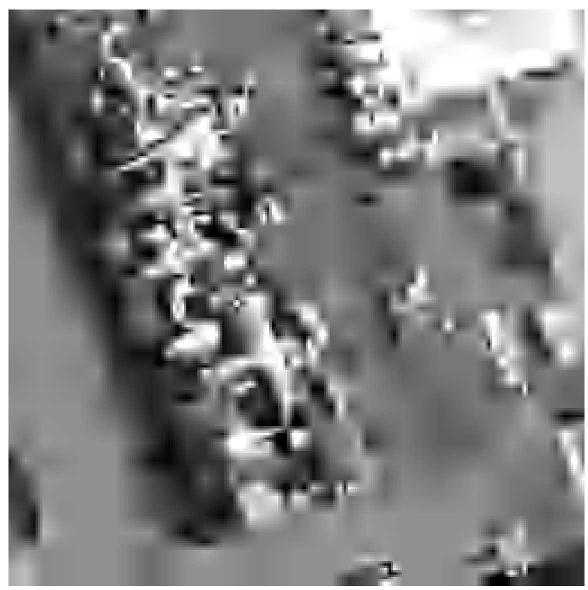
(d) WDR-B

Figure 3: Four 2×2 magnifications of decompressions at 80:1 of Peppers image.

(a) Original

(b) J2K

(c) WDR-I



(d) WDR-B

Figure 4: Four 2×2 magnifications of decompressions at 160:1 of Airfield image.

Table 1: PSNR Values for 20:1 Compression

Image/Method	**WDR-B**	**J2K**	**WDR-I**	**WDR-AC**
Airfield	26.29	27.32	26.79	27.02
Boats	31.72	33.18	32.35	32.42
Goldhill	31.11	32.18	31.63	31.76
Lena	34.24	35.99	35.50	35.72
Peppers	33.80	35.07	34.58	34.21

Table 2: PSNR Values for 2×2 zooms.

Image/Method	**WDR-B**	**J2K**	**WDR-I**
Boat 20:1	32.31	34.32	32.79
Lena 40:1	29.83	29.93	30.27
Peppers 80:1	28.55	28.88	29.24
Airfield 160:1	19.36	19.56	19.40

Table 3: PSNR Values for 40:1 Compression

Image/Method	**WDR-B**	**J2K**	**WDR-I**	**WDR-AC**
Airfield	24.06	24.88	24.52	24.72
Boats	28.72	29.76	29.24	29.32
Goldhill	28.88	29.72	29.37	NA
Lena	31.35	32.75	32.40	NA
Peppers	31.28	32.40	32.05	31.67

Table 4: PSNR Values for 80:1 Compression

Image/Method	**WDR-B**	**J2K**	**WDR-I**	**WDR-AC**
Airfield	22.21	22.64	22.57	22.71
Boats	26.26	26.76	26.82	26.96
Goldhill	27.13	27.69	27.60	27.72
Lena	28.76	29.62	29.47	29.71
Peppers	28.54	29.54	29.35	28.93

Table 5: PSNR Values for 160:1 Compression

Image/Method	**WDR-B**	**J2K**	**WDR-I**	**WDR-AC**
Airfield	20.61	20.84	20.89	20.96
Boats	24.25	24.70	24.91	24.89
Goldhill	25.68	25.76	25.96	NA
Lena	26.15	26.55	26.78	27.13
Peppers	25.80	26.15	26.52	26.14