LOSSY AND LOSSLESS IMAGE COMPRESSION USING REVERSIBLE INTEGER WAVELET TRANSFORMS

Fang Sheng, Ali Bilgin, Philip J. Sementilli†, Michael W. Marcellin

Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721.
† Science Applications International Corp. (SAIC), 101 N. Wilmot Rd., Tucson, AZ 85711
{fsheng,bilgin,mwm}@ece.arizona.edu, sementip@aries.tucson.saic.com
http://www-spacl.ece.arizona.edu

ABSTRACT

There has been recent interest in using reversible integer wavelet transforms for image compression. These transforms allow both lossless and lossy decoding – by resolution and/or accuracy – using a single bitstream. We investigate the lossless and lossy performance of these transforms in the JPEG-2000 Verification Model 0. The lossless compression performance of the presented method is comparable to JPEG-LS. The lossy performance is quite competitive with other efficient lossy compression methods.

1. INTRODUCTION

The wavelet transform has been widely used in image compression. However, until recently, its use has been limited to lossy compression applications. This is due to the fact that most wavelet transforms produce floating-point coefficients which are not well-suited for lossless coding applications. With the introduction of wavelet transforms that map integers to integers, there has been interest in using wavelet transforms for lossless image coding [1, 2, 3, 4, 5].

Using reversible integer wavelet transforms for compression of images has several advantages. Perhaps the most important one is that, through the use of appropriate techniques, a fully embedded bitstream can be generated. In other words, the decoder can extract a lossy version of the image, possibly at reduced resolution, at a desired rate from the bitstream, and continue to decode at higher and higher rates, until the image is perfectly reconstructed. This rate scalability is valuable in many applications. By integrating lossy and lossless compression in a natural fashion, a single image compression method provides excellent lossy performance as well as supporting the many applications that require the ability to exactly recover the original image.

In this paper, we investigate both lossless and lossy performance of reversible integer wavelet transforms. An image coder consisting of a reversible integer wavelet transform and a bit plane coder is presented. The bitplane coder used in this work is the same coder employed in JPEG-2000 Verification Model 0. We compare the lossless compression performance of the presented scheme with that of other state-of-the-art lossless compression schemes in the literature, including wavelet-based, and non-wavelet based coders.

We also compare the progressively decoded lossy performance of several reversible integer wavelet transforms, and compare these with the popular 7x9 wavelet filter of [6] that produces floating-point coefficients.

2. REVERSIBLE INTEGER WAVELET TRANSFORMS

The wavelet transform, in general, produces floating point coefficients. Although these coefficients can be used to reconstruct the original image perfectly in theory, the use of finite precision arithmetic and quantization results in a lossy scheme.

Recently, reversible integer wavelet transforms, i.e. wavelet transforms that transform integers to integers and allow perfect reconstruction of the original signal, have been introduced [1, 2, 3, 4]. In [3], Calderbank et al. introduced a method for building reversible integer wavelet transforms using the lifting scheme of [7]. Here, the input is first split into even and odd indexed samples. Let $x(n)$ be the input signal. Then,

$$s^{(0)}[n] = x[2n] \tag{1}$$

and

$$d^{(0)}[n] = x[2n+1]. \tag{2}$$

Next, M alternating "dual lifting" and "lifting" steps are applied using

$$d^{(i)}[n] = d^{(i-1)}[n] - \lfloor (\sum_k p^{(i)}[k] s^{(i-1)}[n-k]) + \frac{1}{2} \rfloor \tag{3}$$

and

$$s^{(i)}[n] = s^{(i-1)}[n] + \lfloor (\sum_k u^{(i)}[k] d^{(i)}[n-k]) + \frac{1}{2} \rfloor \tag{4}$$

respectively, for $i = 1, \cdots, M$. Finally, the even samples $s^{(M)}[n]$ become the low pass coefficients $s[n]$, and the odd samples $d^{(M)}[n]$ become the high pass coefficients $d[n]$, after scaling by a factor K, such that

$$d[n] = K d^{(M)}[n] \tag{5}$$

and

$$s[n] = s^{(M)}[n]/K. \quad (6)$$

For the transforms considered in this work $K = 1$. The inverse transform is obtained by reversing the steps of the forward transform. The reader is referred to [3] for details.

In this work, we use the reversible integer wavelet transforms presented below. Here, the notation $(N, \tilde{N})$ represents a transform with N and $\tilde{N}$ vanishing moments in the analysis and synthesis high pass filters, respectively. The notation $[m, n]$ represents a transform with m coefficients in the low-pass analysis filter and n coefficients in the high-pass analysis filter.

- A (2,2) transform [3]:

$$d[n] = x[2n+1] - \lfloor \frac{1}{2}(x[2n] + x[2n+2]) + \frac{1}{2} \rfloor \quad (7)$$

$$s[n] = x[2n] + \lfloor \frac{1}{4}(d[n-1] + d[n]) + \frac{1}{2} \rfloor. \quad (8)$$

- A (4,2) transform [3]:

$$d[n] = x[2n+1] - \lfloor \frac{9}{16}(x[2n] + x[2n+2]) - \frac{1}{16}(x[2n-2] + x[2n+4]) + \frac{1}{2} \rfloor \quad (9)$$

$$s[n] = x[2n] + \lfloor \frac{1}{4}(d[n-1] + d[n]) + \frac{1}{2} \rfloor. \quad (10)$$

- A (4,4) transform [3]:

$$d[n] = x[2n+1] - \lfloor \frac{9}{16}(x[2n] + x[2n+2]) - \frac{1}{16}(x[2n-2] + x[2n+4]) + \frac{1}{2} \rfloor \quad (11)$$

$$s[n] = x[2n] + \lfloor \frac{9}{16}(d[n-1] + d[n]) - \frac{1}{16}(d[n-2] + d[n+1]) + \frac{1}{2} \rfloor. \quad (12)$$

- A (2,4) transform [3]:

$$d[n] = x[2n+1] - \lfloor \frac{1}{2}(x[2n] + x[2n+2]) + \frac{1}{2} \rfloor \quad (13)$$

$$s[n] = x[2n] + \lfloor \frac{19}{64}(d[n-1] + d[n]) - \frac{3}{64}(d[n-2] + d[n+1]) + \frac{1}{2} \rfloor. \quad (14)$$

- A (6,2) transform [3]:

$$d[n] = x[2n+1] - \lfloor \frac{75}{128}(x[2n] + x[2n+2]) - \frac{25}{256}(x[2n-2] + x[2n+4]) + \frac{3}{256}(x[2n-4] + x[2n+6]) + \frac{1}{2} \rfloor \quad (15)$$

$$s[n] = x[2n] + \lfloor \frac{1}{4}(d[n-1] + d[n]) - \frac{1}{2} \rfloor. \quad (16)$$

- A (2+2,2) transform [3]:

$$d^{(1)}[n] = x[2n+1] - \lfloor \frac{1}{2}(x[2n] + x[2n+2]) + \frac{1}{2} \rfloor \quad (17)$$

$$s[n] = x[2n] + \lfloor \frac{1}{4}(d^{(1)}[n-1] + d^{(1)}[n]) + \frac{1}{2} \rfloor \quad (18)$$

$$d[n] = d^{(1)}[n] - \lfloor \frac{1}{16}(-s[n-1] + s[n] + s[n+1] - s[n+2]) + \frac{1}{2} \rfloor. \quad (19)$$

- A [2,10] transform used in the current version of the CREW algorithm [1]:

$$d^{(1)}[n] = x[2n+1] - x[2n] \quad (20)$$

$$s[n] = x[2n] + \lfloor \frac{d^{(1)}[n]}{2} \rfloor \quad (21)$$

$$d[n] = d^{(1)}[n] - \lfloor \frac{1}{64}(22(s[n+1] - s[n-1]) + 3(s[n-2] - s[n+2])) + \frac{1}{2} \rfloor. \quad (22)$$

- An S+P transform [2]:

$$d^{(1)}[n] = x[2n+1] - x[2n] \quad (23)$$

$$s[n] = x[2n] + \lfloor \frac{d^{(1)}[n]}{2} \rfloor \quad (24)$$

$$d[n] = d^{(1)}[n] + \lfloor \frac{2}{8}(s[n-1] - s[n]) + \frac{3}{8}(s[n] - s[n+1]) + \frac{2}{8}d^{(1)}[n+1] + \frac{1}{2} \rfloor. \quad (25)$$

3. PROGRESSIVE TRANSMISSION

For progressive transmission, the wavelet coefficients need to be arranged in order of importance. Using mean-squared error (MSE) as the distortion measure, the information that will produce a larger decrease in MSE can be considered to be more important.

Let I be the original image, and T be an orthonormal transform. The transform coefficients C are computed using

$$\mathbf{C} = \mathbf{TI}. \quad (26)$$

If the coefficients are quantized and sent to the decoder, the decoder produces $\hat{\mathbf{C}}$ as an approximation of $\mathbf{C}$, and obtains the reconstructed image $\hat{\mathbf{I}}$ by

$$\hat{\mathbf{I}} = \mathbf{T}^{-1}\hat{\mathbf{C}}. \quad (27)$$

Since orthonormal transforms preserve the L_2 norm, the MSE between the original and reconstructed images is given by

$$MSE = \frac{1}{N}\|I - \hat{I}\|^2 = \frac{1}{N}\|C - \hat{C}\|^2 \quad (28)$$

$$= \frac{1}{N}\sum_x \sum_y |C(x,y) - \hat{C}(x,y)|^2 \quad (29)$$

where N is the number of pixels in the image, and $C(x, y)$ denotes the transform coefficient at coordinate (x, y).

However, the reversible integer transforms used in this work are not orthonormal transforms. For these transforms, the MSE can often be computed by weighting the $|C(x,y) - \hat{C}(x,y)|^2$ terms for each subband in the sum in (29). These scaling factors for each subband can be computed using the method described in [8].

The scaling factors computed using the method in [8] are usually floating point numbers. They can not be used on the integer wavelet coefficients, since scaling these integers by a floating point number would create a floating point number. We normalize the scaling factors so that the minimum scaling factor is 1, the round each scaling factor to the nearest power of two. The scaling factors for the (2,4) transform for 3 levels of dyadic decomposition are shown in Figure 1.

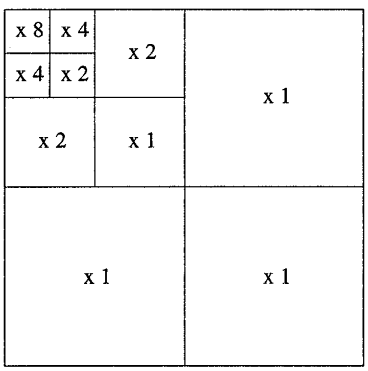

Figure 1: Weights for the (2,4) transform for a 3 level dyadic decomposition.

4. EFFICIENT BIT PLANE CODING

Equation (29) suggests that for efficient progressive transmission larger transform coefficients (after scaling) should be transmitted first, since a larger coefficient would result in a larger decrease in MSE. If the coefficients are represented in binary notation, it would be further beneficial to scan the bitplanes of transform coefficients starting from the most significant bitplane to the least significant bitplane, and transmit the '1' bits in each bitplane. This observation is based on the fact that a '1' in a higher bitplane would produce a larger decrease in MSE, than a '1' in a lower bitplane. In the literature, several image compression methods that benefit from this property have been proposed [1, 2, 9, 10, 11, 12].

The JPEG-2000 VM 0 bitplane coder [13] used in this work has similarities to [1, 10, 11, 12, 14] but has several unique aspects. It uses an embedding principle in which each bitplane within a wavelet subband is "de-interleaved" into three binary sequences: 1) predicted significance bits, 2) refinement bits, 3) predicted insignificance bits. The default encoding order is to encode one subband at a time, in order from lowest to highest resolution. Within each subband, bitplanes are encoded from most- to least-significant. Within each bitplane, the bits are de-interleaved and encoded by an adaptive binary arithmetic coder in order 1), 2), 3). The image is always encoded losslessly.

During decoding of a bitplane at any given resolution layer (i.e., across all subbands within the resolution of the decoded image), bits from de-interleaved sequence 1) are always decoded before any bits are decoded from sequence 2), which in turn are decoded before any bits from sequence 3). (We have found this progressive transmission ordering to be nearly optimal when using floating point wavelet transforms.) This decoding process can stop at any point to produce a lossy result. The translation between the encoder ordering and the ordering used by the decoder is handled by a *parser*.

An important feature of the bitplane coder is that the encoding has no inter-sequence dependencies (e.g., no parent/child contexts). This enables the encoded bitstream to be parsed into any arbitrary order, depending on the application. For example, re-synch markers can be inserted after each encoded sequence if the compressed bitstream is ordered by sequence. This enables a degree of error detection and concealment when transmitting encoded images over noisy channels. The usual "embedded" progressive by accuracy/SNR ordering is accommodated by ordering the data by bitplane, and within each bitplane, in de-interleaved order. Subbands can be emphasized in arbitrary ways, e.g., to compensate for characteristics of an output device, using scaling factors analogous to those described in Section 3. Finally, the independent bitplane coding of subbands facilitates parallel hard/software implementation.

5. EXPERIMENTAL RESULTS

Table 1 presents the lossless compression results obtained using different integer wavelet transforms on a set of JPEG 2000 test images, and Table 2 presents a comparison of these results with other lossless compresion techniques in the literature. Similarly, Table 3 presents the lossless compression results of different integer wavelet transforms on other standard images used in the literature, and Table 4 compares these with other lossless compression methods in the literature. The lossless performance of the presented approach is comparable to JPEG-LS.

Tables 5, and 6 compare the progressive compression performances of the integer wavelet transforms on Barbara and Goldhill images, respectively. In the tables "7x9 FP" entry gives the results using 7x9 filters of [6] with floating-point coefficients, scalar quantization, and progressive decoding from a single bitstream encoded at approximately 4 bpp. The results in these tables were obtained using scaling factors as discussed in the previous sections. (Note that scaling does not change the lossless performance of the algorithm.) These results are for progressive decoding from the losslessly encoded bitstream. We also include the results obtained using SPIHT with S+P transform for comparison.

Although the lossy performance using integer transforms is not quite as good as that for floating point transforms, integer transforms are less complex and offer excellent progressive decoding performance for the many applications that demand the capability to losslessly recover the original image. Improved lossy performance when using integer transforms is a pursuit of our on-going work.

6. REFERENCES

[1] A. Zandi, J. D. Allen, E. L. Schwartz, and M. Boliek, "CREW: Compression with reversible embedded wavelets", in *Proc. of Data Compression Conference*, Mar. 1995, pp. 212–221.

[2] A. Said and W.A. Pearlman, "An image multiresolution representation for lossless and lossy compression", *IEEE Transactions on Image Processing*, vol. 5, pp. 1303–1310, Sept. 1996.

[3] R. Calderbank, I. Daubechies, W. Sweldens, and B.-L. Yeo, "Wavelet transforms that map integers to integers", *submitted to Journal of Applied and Computational Harmonics Analysis.*

[4] S. Dewitte and J. Cornelis, "Lossless integer wavelet transform", *IEEE Signal Processing Letters*, vol. 4, pp. 158–160, June 1997.

[5] N. Memon, X. Wu, and B.-L. Yeo, "Entropy coding techniques for lossless image compression with reversible integer wavelet transforms", Tech. Rep., IBM Research Report RC 21010, Oct. 1997.

[6] M. Antonini, M. Barlaud, P. Mathieu and I. Daubechies, "Image coding using wavelet transform", *IEEE Transactions on Image Processing*, vol. 1, pp. 205–220, Apr. 1992.

[7] W. Sweldens, "The lifting scheme: A custom-design construction of biorthogonal wavelets", *Journal of Applied and Computational Harmonic Analysis*, vol. 3(2), pp. 186–200, 1996.

[8] J. W. Woods and T. Naveen, "A filter based bit allocation scheme for subband compression of HDTV", *IEEE Trans. on Image Processing*, vol. 1, pp. 436–440, July 1992.

[9] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients", *IEEE Transactions on Signal Processing*, vol. 41, pp. 3445–3462, Dec. 1993.

[10] D. Taubman and A. Zakhor, "Multirate 3-D subband coding of video", *IEEE Transactions on Image Processing*, vol. 3, pp. 572–588, Sept. 1994.

[11] J. Li and S. Lei, "Rate-distortion optimized embedding", in *Proc. of Picture Coding Symposium*, Sept. 1997, pp. 201–206.

[12] E. Ordentlich, M. Weinberger, and G. Seroussi, "A low-complexity modeling approach for embedded coding of wavelet coefficients", Tech. Rep., HP-Labs Report, HPL-97-150, Dec. 1997.

[13] P. J. Sementilli, A. Bilgin, J. H. Kasner, and M. W. Marcellin, "Wavelet TCQ: Submission to JPEG 2000", in *Proc. of SPIE, Applications of Digital Image Processing*, July 1998.

[14] X. Wu, "High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression", in *Proc. of 31st Asilomar Conf. on Signals, Systems, and Computers,*, Nov. 1997.

[15] M. J. Weinberger, G. Seroussi, and G. Sapiro, "LOCO-I: A low complexity, context-based lossless image compression algorithm", in *Proc. of 1996 Data Compression Conference*, 1996, pp. 140–149.

[16] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding", *IEEE Transactions on Communications*, vol. 45, pp. 437–444, Apr. 1997.

Table 1: Comparison of the lossless compression performances of different integer wavelet transforms on JPEG-2000 images (in bpp).

Wavelet	Image				
	Target	Bike	Woman	Cafe	Aerial2
(4,4)	2.390	4.402	4.364	5.180	5.219
(4,2)	2.290	4.409	4.368	5.180	5.222
(2,4)	2.239	4.430	4.409	5.217	5.261
(6,2)	2.368	4.420	4.373	5.195	5.229
(2+2,2)	2.382	4.429	4.383	5.194	5.225
[2,10]	2.363	4.434	4.392	5.208	5.230
(2,2)	2.113	4.414	4.393	5.194	5.243
S+P	2.233	4.415	4.378	5.190	5.233

Table 2: Comparison of different lossless compression methods using JPEG-2000 images (in bpp).

Method	Image				
	Target	Bike	Woman	Cafe	Aerial2
SPIHT [2]	2.646	4.480	4.442	5.283	5.332
LOCO-I [15]	2.186	4.356	4.451	5.092	5.288
CALIC [16]	2.291	4.196	4.287	4.927	5.142
Proposed with (2,2)	2.113	4.414	4.393	5.194	5.243

Table 3: Comparison of the lossless compression performances of different integer wavelet transforms on other images (in bpp).

Wavelet	Image		
	Barbara	Goldhill	Lenna
(4,4)	4.582	4.683	4.158
(4,2)	4.612	4.686	4.166
(2,4)	4.678	4.705	4.208
(6,2)	4.596	4.689	4.170
(2+2,2)	4.625	4.690	4.182
[2,10]	4.653	4.711	4.182
(2,2)	4.699	4.690	4.197
S+P	4.647	4.704	4.187

Table 4: Comparison of different lossless compression methods using other images (in bpp).

Method	Image		
	Barbara	Goldhill	Lenna
SPIHT [2]	4.711	4.778	4.188
LOCO-I [15]	4.863	4.712	4.236
CALIC [16]	4.626	4.629	4.101
Proposed with (4,4)	4.582	4.683	4.158

Table 5: Comparison of the lossy compression performances of different integer wavelet transforms with scaling on Barbara (PSNR (dB)).

Wavelet	Rate (bpp)				
	0.1	0.2	0.5	0.7	1.0
7x9 FP	24.18	26.65	31.64	34.17	36.90
(4,4)	23.89	26.41	31.14	33.35	35.65
(4,2)	23.69	26.13	30.85	33.14	35.43
(2,4)	23.97	26.14	30.65	32.92	35.30
(6,2)	23.50	25.96	30.77	33.20	35.60
(2+2,2)	23.94	26.20	31.05	33.38	35.80
[2,10]	24.04	26.44	31.36	33.54	36.32
(2,2)	23.82	25.86	30.25	32.45	34.91
S+P	23.92	26.35	31.04	33.42	35.92
S+P w/SPIHT	23.89	26.05	30.57	33.02	35.53

Table 6: Comparison of the lossy compression performances of different integer wavelet transforms with scaling on Goldhill (PSNR (dB)).

Wavelet	Rate (bpp)				
	0.1	0.2	0.5	0.7	1.0
7x9 FP	27.77	29.82	33.20	34.67	36.80
(4,4)	27.33	29.14	32.65	33.82	35.73
(4,2)	27.30	29.09	32.56	33.73	35.61
(2,4)	27.17	29.21	32.59	33.85	35.84
(6,2)	27.00	28.90	32.29	33.63	35.50
(2+2,2)	27.33	29.49	32.83	34.12	35.95
[2,10]	27.68	29.67	32.87	34.28	36.16
(2,2)	27.13	29.23	32.63	33.88	35.85
S+P	27.67	29.62	32.86	34.22	36.00
S+P w/SPIHT	27.70	29.49	32.60	34.08	35.80