Wavelet Transforms in the JPEG-2000 Standard

Michael D. Adams and Rabab Ward

Dept. of Elec. and Comp. Engineering, University of British Columbia, Vancouver, BC, Canada, V6T 1Z4
mdadams@ieee.org and rababw@ece.ubc.ca

Abstract

Studied are numerous issues associated with wavelet transforms in the JPEG-2000 Part-1 standard (i.e., ISO/IEC 15444-1). The dynamic range of wavelet transform coefficients and computational complexity of wavelet transforms are examined. Also, the effect of wavelet transforms on coding efficiency is investigated. The information presented herein may prove beneficial to those developing JPEG-2000 codec implementations.

1. Introduction

JPEG-2000 Part 1 has recently been approved as a new international standard for still image compression. This standard, formally known as ISO/IEC 15444-1 [1], provides a rich feature set including support for both lossy and lossless coding. One of the core technologies employed by the JPEG-2000 codec is the wavelet transform. Thus, in order to build a high-quality JPEG-2000 encoder or decoder, one must be able to construct an effective wavelet transform engine. This observation motivated us to examine numerous practical issues related to the implementation of wavelet transforms in the context of the JPEG-2000 standard. In this paper, we present the results of our study.

The remainder of this paper is structured as follows. First, we begin with a brief introduction to the JPEG-2000 codec. Then, we study various issues related to the implementation of wavelet transforms. For example, we examine the dynamic range of transform coefficients and computational complexity of transforms. Also, we study the effects of wavelet transforms on coding efficiency. Finally, we conclude with a summary of our results.

2. JPEG-2000 Codec

As far as the JPEG-2000 codec is concerned, an image is a collection of components, where each component consists of a rectangular array of samples. The samples from the various components have a particular relative alignment, and each component represents a specific type of information (e.g., spectral, opacity). Since each component can be sampled at a different spatial resolution, all of the components need not have the same width and height. The samples of each component can be either signed or unsigned integers with ρ bits of precision, where $1 \leq \rho \leq 38$. Thus, in the signed and unsigned cases, the component sample values have a nominal dynamic range of $[-2^{\rho-1}, 2^{\rho-1} - 1]$ and $[0, 2^{\rho} - 1]$, respectively.

The JPEG-2000 codec is transform-based. It employs multicomponent transforms, wavelet transforms, and bit-plane coding techniques, in order to provide a framework for both lossy and lossless compression. Both reversible integer-to-integer and nonreversible real-to-real transforms are employed, the latter being referred to as "irreversible" in the terminology of the standard.

The general structure of the encoder is shown in Fig. 1(a). The input to the encoding process is an image consisting of one or more components. Before any further processing takes place, each component has its sample values adjusted by an additive bias, in a process called DC level shifting. The bias is chosen such that the resulting sample values have a nominal dynamic range (approximately) centered about zero. Then, a multicomponent transform

This work was supported by the Natural Sciences and Engineering Research Council of Canada.

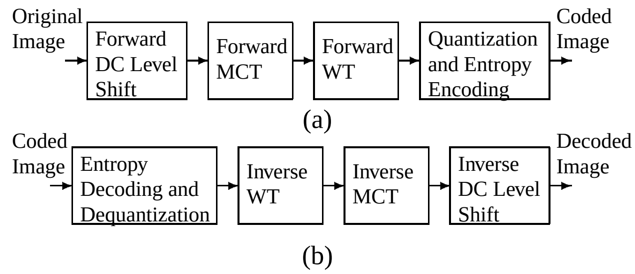

Fig. 1. The general structure of the JPEG-2000 codec. The (a) encoder and (b) decoder.

(MCT) may be applied collectively to a number of the components. Next, a wavelet transform (WT) may be applied to each component individually. Finally, the resulting transform coefficients are quantized and then encoded. In the case of lossless coding, reversible transforms must be employed and all quantizer step sizes are forced to be one. In the lossy case, either reversible or nonreversible transforms can be used, but the two types of transforms cannot be intermixed.

The general structure of the decoder is illustrated in Fig. 1(b). As is evident from the diagram, the decoder structure simply mirrors that of the encoder.

3. Wavelet Transforms

All of the wavelet transforms employed by the JPEG-2000 Part-1 codec are fundamentally one dimensional (1-D) in nature. Two dimensional (2-D) transforms are formed by applying 1-D transforms in the horizontal and vertical directions. As suggested earlier, both reversible integer-to-integer [2,3] and nonreversible real-to-real [4] wavelet transforms are employed by the codec. In order to handle filtering at signal boundaries, symmetric extension is used. At the time of this writing, two wavelet transforms are supported by the codec: the reversible 5/3 and nonreversible 9/7. An amendment to the standard is currently being drafted, proposing the addition of another transform (namely, the nonreversible 5/3 transform). Since the future of this amendment is, at best, uncertain, we will not further consider the transform specified in this amendment.

Before proceeding further, we present the formal definitions of the two currently supported transforms. As a matter of notation, the input signal, lowpass subband signal, and highpass subband signal are denoted as $x[n]$, $s[n]$, and $d[n]$, respectively. For convenience, we also define the quantities $s_0[n] = x[2n]$ and $d_0[n] = x[2n+1]$. With the preceding definitions, the forward equations for the reversible 5/3 transform are given by

$$d[n] = d_0[n] - \lfloor \tfrac{1}{2}(s_0[n+1] + s_0[n]) \rfloor \quad (1a)$$

$$s[n] = s_0[n] + \lfloor \tfrac{1}{4}(d[n] + d[n-1]) + \tfrac{1}{2} \rfloor \quad (1b)$$

Similarly, the forward equations for the nonreversible 9/7 trans-

Table 1. Test images

Image	Size	Depth	Description
bike	2048×2560	8	collection of objects
cr	1744×2048	10	computer radiology
elev	1201×1201	12	fractal-like pattern
gold	720×576	8	houses and countryside
sar2	800×800	12	synthetic aperture radar
target	512×512	8	patterns and textures

form are given by

$$d_1[n] = d_0[n] - \alpha_0(s_0[n+1] + s_0[n]) \tag{2a}$$
$$s_1[n] = s_0[n] - \alpha_1(d_1[n] + d_1[n-1]) \tag{2b}$$
$$d_2[n] = d_1[n] + \alpha_2(s_1[n+1] + s_1[n]) \tag{2c}$$
$$s_2[n] = s_1[n] + \alpha_3(d_2[n] + d_2[n-1]) \tag{2d}$$
$$s[n] = \beta_0 s_2[n] \tag{2e}$$
$$d[n] = \beta_1 d_2[n] \tag{2f}$$

where

$$\alpha_0 \approx 1.586134,\ \alpha_1 \approx 0.052980,\ \alpha_2 \approx 0.882911,$$
$$\alpha_3 \approx 0.443506,\ \beta_0 \approx 0.812893,\ \beta_1 = 1/\beta_0.$$

In the case of both of the above transforms, the inverse transformation equations can be trivially deduced from the forward transformation equations, and thus are not given here.

4. Lifting Realization

In order to implement the wavelet transform engine for a JPEG-2000 encoder or decoder, one must first select an appropriate computational structure for this purpose. As it turns out, the lifting framework [3, 5] is a natural choice. This framework facilitates efficient wavelet transform realizations and can handle all of the reversible integer-to-integer and nonreversible real-to-real wavelet transforms associated with the JPEG-2000 standard. In passing, we note that the transformation equations given in (1) and (2) correspond to lifting realizations. In the remainder of this paper, we assume that the lifting realization is employed.

5. Experimental Results

Throughout the remainder of this paper, we present experimental results obtained with a JPEG-2000 codec implementation. In order to obtain these results, we employed JasPer Version 1.200. (JasPer [6] is a free publically-available software implementation of the JPEG-2000 codec.) For image data, we used several test images from the JPEG-2000 test set, as listed in Table 1.

6. Dynamic Range of Transform Coefficients

When the (forward) wavelet transform is applied to data, the dynamic range of the data typically increases. This dynamic range growth is important because it has the potential to lead to numerical overflow. Therefore, it is important to characterize the dynamic range of the wavelet transforms supported by the codec. (Ideally, a smaller dynamic range is preferable, since it allows a smaller word size to be employed.)

First, we consider the reversible 5/3 transform. The behavior of reversible transforms is quite rigidly specified in the standard. This is necessary in order to ensure exact transform invertibility (which is clearly needed for lossless coding). To study the reversible 5/3 transform, we transformed several images, and noted the range of the input and output data as well as the subbands with the minimum and maximum coefficient values. A representative subset of the results is given in Table 2. It is not unusual to see the dynamic range grow by as much as a factor of eight (e.g., for the `bike` image with five decomposition levels, the dynamic range grows by a factor of about six). Therefore, at least three extra bits are required to accommodate this growth. For example, to handle 8 bit/sample images, we need a word size of at least 11 (i.e., $8+3$) bits for storing transform coefficients. From the above results, we can also see that the subbands at the coarsest resolution (i.e., with the largest decomposition level index) do not necessarily contain the minimum/maximum coefficient values (e.g., the `target` image with four or five decomposition levels).

Next, we consider the nonreversible 9/7 transform. Although reversible transforms are quite rigidly specified, there is a great deal of flexibility in the nonreversible case. In order to avoid unnecessary dynamic range growth, it is often desirable to normalize a transform such that the lowpass and highpass analysis filters have a gain of one at DC and the Nyquist rate, respectively. This often leads to transform coefficients with a smaller dynamic range. In the case of the nonreversible 9/7 transform, a careful examination of equation (2) shows that the highpass analysis filter has a Nyquist gain of two. By using a scaling factor of $\beta_1' = \beta_1/2$ in equation (2f) instead of β_1, a unity gain is obtained for the highpass analysis filter. In what follows, we refer to the original and modified nonreversible 9/7 transform normalizations as (1,2) and (1,1), respectively. Due to the difference in highpass analysis filter normalization, the (1,2) normalization yields HL, LH, and HH bands with the coefficients scaled by two, two, and four, respectively, relative to the (1,1) normalization.

For both normalizations of the nonreversible 9/7 transform, we transformed several images, and observed the input and output dynamic ranges and well as the subbands with the minimum and maximum coefficient values. A representative subset of the results is given in Table 3. Clearly, the (1,1) normalization yields a smaller dynamic range than the (1,2) normalization. Theoretically, the two dynamic ranges can differ by a factor of up to four. In practice, however, the difference is usually somewhat smaller, although still significant. In the case of the (1,1) normalization, the LL, LH, and HL bands tend to contain the minimum and maximum coefficient values, while in the case of the (1,2) normalization the HH, LH, and HL bands tend to contain these values. Although using the (1,1) normalization will cause the transform coefficients to be scaled differently, one can trivially compensate for this incorrect scaling during transform coefficient quantization. (For a more concrete example of how this compensation can be performed, one can look at the JasPer software.)

7. Computational Complexity

In the design of most practical systems, complexity is often an important concern. Therefore, it is beneficial to study the computational complexity of the various transforms supported by JPEG-2000 Part 1. To this end, we examined the complexity of the wavelet transform code of a typical software implementation (namely, JasPer).

In our experiment, the wavelet transform was applied to two images, and in each case the execution times for the forward and inverse transformations were measured. The results are given in Table 4. Examining the results for the two different transforms, we can clearly see that the nonreversible 9/7 transform is significantly more complex, typically requiring about twice as much time as the reversible 5/3 transform. In the case of each transform, the computation time for the forward and inverse are roughly comparable.

Basically, two types of operations are associated with the wavelet transform computation: lift/scale operations (i.e., filtering) and split/join operations (i.e., rearrangement of samples required as a result of upsampling/downsampling). The difference in the computational complexities of the reversible 5/3 and nonreversible 9/7 transforms is solely attributable to the lift/scale operations, as the split/join operations are identical for both transforms. More detailed profiling of the code shows that the lift/scale operations of the nonreversible 9/7 transform are about three times slower than

Table 3. Dynamic range of the transform coefficients for two different normalizations of the nonreversible 9/7 transform

			(1,1) Normalization		(1,2) Normalization	
Image	Input Range	# of Levels	Output Range	Bands with Min. & Max.	Output Range	Bands with Min. & Max.
`bike`	[-128,127]	1	[-163.9,164.4]	LL_0,LL_0	[-352.0,278.8]	HH_0,HH_0
		2	[-154.9,149.3]	LL_1,LL_1	[-498.4,507.3]	HH_1,HH_1
		3	[-149.2,147.6]	HL_0,LL_2	[-510.7,510.5]	HH_2,HH_2
		4	[-149.2,144.9]	HL_0,HL_1	[-510.7,510.5]	HH_2,HH_2
		5	[-149.2,144.9]	HL_0,HL_1	[-510.7,510.5]	HH_2,HH_2
`cr`	[-512,511]	1	[-583.5,525.5]	LL_0,LL_0	[-583.5,525.5]	LL_0,LL_0
		2	[-569.6,544.1]	LL_1,LL_1	[-587.8,544.1]	LH_1,LL_1
		3	[-555.3,625.8]	LL_2,LL_2	[-587.8,625.8]	LH_1,LL_2
		4	[-513.1,466.4]	LL_3,HL_3	[-911.1,932.8]	HL_3,HL_3
		5	[-527.9,466.4]	LL_4,HL_3	[-911.1,932.8]	HL_3,HL_3
`elev`	[-1094,1552]	1	[-1091.7,1531.9]	LL_0,LL_0	[-1091.7,1531.9]	LL_0,LL_0
		2	[-1088.1,1477.2]	LL_1,LL_1	[-1088.1,1477.2]	LL_1,LL_1
		3	[-1081.7,1401.4]	LL_2,LL_2	[-1081.7,1401.4]	LL_2,LL_2
		4	[-1067.3,1329.3]	LL_3,LL_3	[-1067.3,1329.3]	LL_3,LL_3
		5	[-1056.2,1201.0]	LL_4,LL_4	[-1056.2,1201.0]	LL_4,LL_4
`gold`	[-112,107]	1	[-115.8,107.3]	LL_0,LL_0	[-118.9,107.3]	LH_0,LL_0
		2	[-116.6,107.1]	LL_1,LL_1	[-118.9,111.4]	LH_0,HL_1
		3	[-114.5,106.6]	LL_2,LL_2	[-118.9,111.4]	LH_0,HL_1
		4	[-110.4,113.2]	LL_3,LL_3	[-118.9,113.2]	LH_0,LL_3
		5	[-107.1,114.9]	LL_4,LL_4	[-118.9,114.9]	LH_0,LL_4
`sar2`	[-2048,2047]	1	[-2036.3,1828.6]	LL_0,LL_0	[-2501.2,2161.4]	HH_0,HH_0
		2	[-1756.6,1378.5]	LL_1,LL_1	[-2501.2,2161.4]	HH_0,HH_0
		3	[-1597.3,866.2]	LL_2,LH_1	[-2501.2,2161.4]	HH_0,HH_0
		4	[-1556.1,866.2]	LL_3,LH_1	[-2501.2,2161.4]	HH_0,HH_0
		5	[-1391.1,866.2]	LL_4,LH_1	[-2501.2,2161.4]	HH_0,HH_0
`target`	[-128,127]	1	[-161.6,156.5]	LH_0,LL_0	[-607.3,524.9]	HH_0,HH_0
		2	[-161.6,151.9]	LH_0,LH_0	[-607.3,524.9]	HH_0,HH_0
		3	[-161.6,151.9]	LH_0,LH_0	[-607.3,524.9]	HH_0,HH_0
		4	[-161.6,151.9]	LH_0,LH_0	[-607.3,524.9]	HH_0,HH_0
		5	[-161.6,151.9]	LH_0,LH_0	[-607.3,524.9]	HH_0,HH_0

Table 2. Dynamic range of the transform coefficients for the reversible 5/3 transform

Image	Input Range	# of Levels	Output Range	Bands with Min. & Max.
`bike`	[-128,127]	1	[-242,224]	HL_0,HL_0
		2	[-519,526]	HH_1,HH_1
		3	[-568,712]	HH_2,HH_2
		4	[-568,741]	HH_2,HH_3
		5	[-568,741]	HH_2,HH_3
`cr`	[-512,511]	1	[-640,606]	LL_0,LL_0
		2	[-784,709]	LH_1,LL_1
		3	[-859,947]	HH_2,LL_2
		4	[-1128,1118]	HL_3,HL_3
		5	[-1187,1118]	LH_4,HL_3
`elev`	[-1094,1552]	1	[-1093,1557]	LL_0,LL_0
		2	[-1093,1548]	LL_1,LL_1
		3	[-1091,1491]	LL_2,LL_2
		4	[-1090,1461]	LL_3,LL_3
		5	[-1088,1401]	LL_4,LL_4
`gold`	[-112,107]	1	[-124,114]	LL_0,LL_0
		2	[-129,136]	LL_1,HL_1
		3	[-127,137]	LL_2,HH_2
		4	[-122,147]	HL_3,LL_3
		5	[-130,163]	HL_4,LL_4
`sar2`	[-2048,2047]	1	[-2257,2026]	LL_0,LL_0
		2	[-2194,2233]	HH_1,LH_1
		3	[-3043,2233]	HH_2,LH_1
		4	[-3043,2332]	HH_2,HH_3
		5	[-3043,2332]	HH_2,HH_3
`target`	[-128,127]	1	[-401,332]	HH_0,HH_0
		2	[-448,448]	HH_1,HH_1
		3	[-462,485]	HH_2,HH_2
		4	[-462,485]	HH_2,HH_2
		5	[-462,485]	HH_2,HH_2

those of the reversible 5/3 transform.

8. Number of Decomposition Levels and Coding Efficiency

The JPEG-2000 encoder must choose the number of wavelet decomposition levels to employ. The standard allows from 0 to 32 levels (inclusive). Therefore, some guidelines for selecting the number of decomposition levels would be beneficial.

Generally, as the number of decomposition levels is increased from zero, coding efficiency increases up to some optimum point, and then levels off (or even decreases slightly). This behavior is illustrated for the cases of lossless and lossy coding in Figs. 2 and 3, respectively. As can be seen from these graphs, most of the coding efficiency is contributed by the first three to five decomposition levels. Based on these observations, it would seem reasonable to use five decomposition levels as an encoder default.

Table 4. Computation time for a 5-level wavelet transform (on a system with an 800 MHz Pentium III processor)

	Execution Time (s)			
	Reversible 5/3		Nonreversible 9/7	
Image	Forward	Inverse	Forward	Inverse
`elev`	0.178	0.178	0.300	0.317
`target`	0.131	0.135	0.253	0.260

The above relationship between coding efficiency and the number of decomposition levels can be easily explained. As the number of decomposition levels grows, the transform's coding gain increases rapidly, asymptotically approaching some limit value. This causes the rapid increase in coding efficiency for the first few decomposition levels. Once the coding gain nears its limit value, other factors can begin to adversely affect coding efficiency. For example, as the number of decomposition levels increases: 1) the resulting subbands become smaller, and the boundary effects associated with the filtering (i.e., transform) process can adversely impact coding efficiency; 2) the number of resolution levels increase, and more packets are required in the code stream, resulting in greater packetization overhead; and 3) the effects of finite-precision arithmetic (i.e., increased rounding error) can become more pronounced.

9. Transforms and Coding Efficiency

When lossless coding is desired, we need to use the reversible 5/3 transform. In the lossy case, however, either the reversible 5/3 or nonreversible 9/7 transform can be employed. Therefore, the relative performance of these two transforms for lossy coding is of great interest. With this in mind, we compared the coding per-

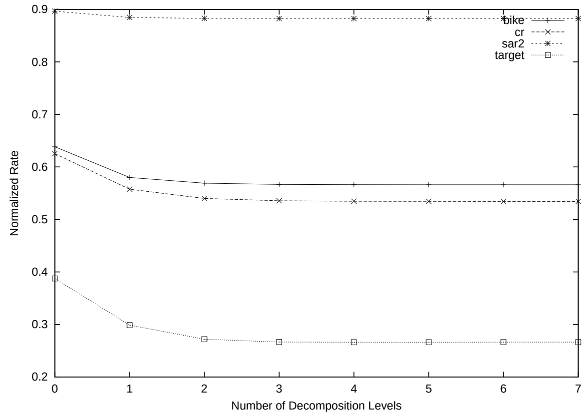

Fig. 2. The effects of varying the number of decomposition levels on lossless coding efficiency (with the reversible 5/3 transform).

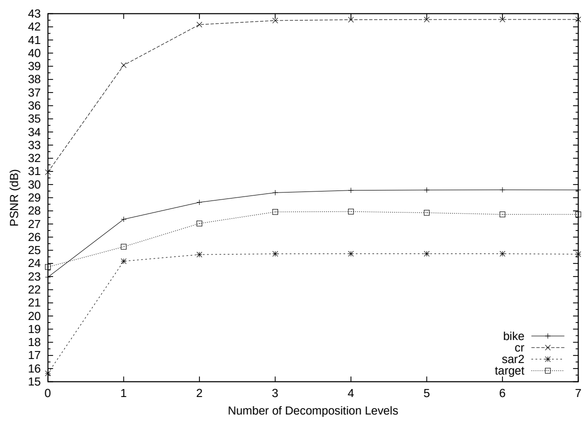

Fig. 3. The effects of varying the number of decomposition levels on lossy coding efficiency (with the nonreversible 9/7 transform at 32:1 compression).

formance obtained with both transforms for several test images coded at various bit rates. A representative subset of the results is shown in Table 5. As can be seen from these results, the nonreversible 9/7 transform consistently yields a higher PSNR than the reversible 5/3 transform.

Of course, subjective image quality is also an important consideration, since PSNR does not always correlate well with distortion as perceived by the human visual system. This said, however, informal subjective tests suggest that the two transforms are roughly comparable, with the nonreversible 9/7 transform faring slightly better. Since the nonreversible 9/7 transform does not yield vastly superior compression results, the reversible 5/3 transform is quite attractive, given its significantly lower computational complexity and ability to facilitate lossless coding (unlike the nonreversible 9/7 transform).

10. Conclusions

Several issues associated with wavelet transforms in the JPEG-2000 standard have been studied. First, we examined the dynamic range of transform coefficients. In the case of the reversible 5/3 transform, we found that the dynamic range may often grow by about three bits during the transform process. In the case of the nonreversible 9/7 transform, we found that the (1,1) normalization leads to a significantly smaller dynamic range than the (1,2)

Table 5. Lossy compression performance with the reversible 5/3 and nonreversible 9/7 transforms (five decomposition levels)

Image	Compression Factor	Rev. 5/3 PSNR (dB)	Nonrev. 9/7 PSNR (dB)
`bike`	1/128	23.24	23.76
	1/64	25.99	26.50
	1/32	29.05	29.59
	1/16	32.93	33.47
	1/8	37.32	37.95
`cr`	1/128	40.25	40.53
	1/64	41.36	41.63
	1/32	42.20	42.55
	1/16	43.53	43.78
	1/8	45.94	46.33
`elev`	1/128	49.81	50.76
	1/64	54.15	55.55
	1/32	58.36	60.08
	1/16	62.86	64.72
	1/8	68.99	70.19
`gold`	1/128	27.16	27.62
	1/64	29.22	29.53
	1/32	31.22	31.54
	1/16	33.80	34.17
	1/8	37.08	37.59
`sar2`	1/128	22.45	22.74
	1/64	23.39	23.56
	1/32	24.37	24.74
	1/16	26.23	26.42
	1/8	29.81	30.08
`target`	1/128	19.11	20.27
	1/64	22.28	23.89
	1/32	26.67	27.86
	1/16	33.10	34.76
	1/8	42.77	44.41

normalization. With the former normalization, the dynamic range typically grows by at most one bit. Next, we examined the computational complexity of the various transforms. The nonreversible 9/7 transform takes approximately twice as long to compute as the reversible 5/3 transform. Therefore, for encoder implementations with tight complexity constraints, the exclusive use of the reversible 5/3 transform is clearly very attractive. This transform allows for a smaller word size, requires less computation, and facilitates both lossy and lossless coding. For an encoder default, we observed that five wavelet decomposition levels is a reasonable choice. Lastly, in the case of lossy coding, the reversible 5/3 transform was shown to yield somewhat poorer results (in terms of PSNR) compared to the nonreversible 9/7 transform.

11. References

[1] ISO/IEC, *ISO/IEC 15444-1: Information technology—JPEG 2000 image coding system—Part 1: Core coding system*, 2001.

[2] M. D. Adams and F. Kossentini, "Reversible integer-to-integer wavelet transforms for image compression: Performance evaluation and analysis," *IEEE Trans. on Image Processing*, vol. 9, no. 6, pp. 1010–1024, June 2000.

[3] M. D. Adams, "Reversible wavelet transforms and their application to embedded image compression," M.A.Sc. thesis, Dept. of Elec. and Comp. Engineering, University of Victoria, Victoria, BC, Canada, Jan. 1998, Available from `http://www.ece.ubc.ca/~mdadams`.

[4] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 205–220, Apr. 1992.

[5] W. Sweldens, "The lifting scheme: A new philosophy in biorthogonal wavelet constructions," in *Proc. of SPIE*, San Diego, CA, USA, Sept. 1995, vol. 2569, pp. 68–79.

[6] M. D. Adams and F. Kossentini, "JasPer: A software-based JPEG-2000 codec implementation," in *Proc. of IEEE ICIP*, Vancouver, BC, Canada, Oct. 2000, vol. 2, pp. 53–56.