A COMPARISON OF FIXED-POINT 2D 9X7 DISCRETE WAVELET TRANSFORM IMPLEMENTATIONS

Hyung Cook Kim and Edward J. Delp

Video and Image Processing Laboratory
School of Electrical and Computer Engineering
Purdue University
West Lafayette, IN 47907-1285
{kim74, ace}@ecn.purdue.edu

ABSTRACT

In this paper, we describe three 2D Discrete Wavelet Transform fixed-point implementations and compare them in terms of quantization error for the Daubechies 9x7 filter bank. The three implementations are the polyphase form, lifting scheme, and reduced scaling lifting scheme. Experimental results show that the reduced scaling lifting scheme is more robust than other schemes. Also, the numbers of cycles the implementations take on a Texas Instruments TMS320C6201 simulator are given as reference.

1. INTRODUCTION

The 2D discrete wavelet transform (DWT) is used in many image and video algorithms including the JPEG 2000 still image compression standard [1], [2]. The 2D wavelet transform is usually obtained by using the 1D discrete wavelet transform in a separable manner in the column and the row directions. The most popular 1D DWT uses the Daubechies 9x7 filter-bank, which was first introduced in [3]. For better computational efficiency, it is well known that the 1D discrete wavelet transform can be implemented using the polyphase form [4]. Recently, alternative implementation of the 1D DWT has been proposed, known as the lifting scheme [5], [6]. In this paper, we compare 3 fixed-point 2D DWT implementations for the 9x7 filter-bank: polyphase form, lifting scheme and reduced scaling lifting scheme. We chose the 9x7 filter-bank because the length of the analysis filter and the synthesis filter is similar so we can remove the effects different analysis and synthesis filter lengths have on quantization errors and it is used in JPEG 2000 standard part I [1]. We have found no other comparisons in the literature with respect to fixed-point lifting implementations. The quantization errors introduced by DWT and inverse DWT (IDWT) are compared among different implementations. Also, the cycle times of each implementation on a Texas Instruments TMS320C62X DSP simulator are given as reference. All the values of the constants used in this paper are given in [2].

This work was partially supported by a grant from the Indiana 21st Century Research and Technology Fund.
Address all correspondence to E. J. Delp

2. 2D 9X7 FIXED-POINT DWT IMPLEMENTATIONS

A 1D DWT block diagram is shown in Figure 1. The direct implementation of the 1D DWT is shown in Figure 2 where h_0 and h_1 are low-pass and high-pass analysis filters, respectively. To increase computational efficiency, polyphase form and the lifting scheme exploits the redundancies in the direct implementation of the 1D DWT. The 2D wavelet transform is obtained by using the 1D DWT in a separable manner in the row and the column direction as shown in Figure 3.

In this paper, we implement all constants and wavelet coefficients as 16 bit integers. This is to reflect the fact that current DSPs and general purpose processors usually multiply 16 bit integers faster than 32 bit integers and represent the largest integer as 32 bits. Also, roundoff is always used if there is a change in the number representation format.

2.1. Polyphase form

The polyphase form of the 1D DWT is given in Figure 4. h_0^e and h_0^o are polyphase filters for the low-pass analysis filter and h_1^e and h_1^o are polyphase filters for the high-pass analysis filter, respectively. The polyphase form approximately reduces the number of operations in half compared to the direct implementation by basically not calculating the samples that are discarded in the downsampling operation. A similar form can be obtained for the IDWT. Define g_0^e and g_0^o as polyphase filters for the low-pass synthesis filter and g_1^e and g_1^o as polyphase filters for the high-pass synthesis filter for the IDWT. The formats we use for each filter are given in Table 1. This insures that we obtain the maximum precision without sacrificing computational efficiency. The polyphase form and the direct form are well suited for DSPs because it can use the multiply-accumulate (MAC) operations in DSPs efficiently.

2.2. Lifting scheme

1D DWT implementation using the lifting scheme is given in Figure 5. Lifting scheme consists of the reversible 1D lifting scheme [5] given in Figure 6 followed by scaling operations. The advantages of the lifting scheme is that it reduces the number of operations approximately by half and the inverse DWT can be constructed by following the forward DWT in reverse order. It achieves these advantages by decomposing the polyphase filters into elementary lifting steps [6].

The constants used in the lifting scheme are given in [2]. K_0, $1/K_0$, p1, u1, p2 and u2 are represented in Q14, Q15, Q14, Q19, Q15 and Q16 format, respectively. This is to reduce the quantization error as much as possible.

2.3. Reduced scaling lifting scheme

A variation of the 2D DWT lifting scheme implementation is given in Figure 7. It is equivalent to the 2D DWT lifting scheme implementation but it reduces the number of scaling operations by 1/4 compared to the original 2D lifting scheme implementation. It also reduces the quantization errors produced by scaling as well. This is possible by recognizing the fact that, as mentioned before, lifting scheme consists of the reversible 1D lifting scheme followed by scaling operations. This means lifting scheme requires two scaling operations for each subband. Instead of doing scaling operations twice for each subband, operations can be reduced by implementing 2D DWT as a 2D reversible DWT followed by one scaling operation for each subband as shown in Figure 7.

The constants used in the reduced scaling lifting scheme are the same as in the lifting scheme except that K_0^2 and $1/K_0^2$ are represented in Q14 and Q15 format, respectively.

3. EXPERIMENTAL RESULTS

We compare the quantization errors of the fixed-point 2D DWT implementations for the 9x7 filter-bank by using a 6 level Mallat decomposition and reconstruction on a set of test images. The images are expanded by reflection on the boundaries [2], [7] for perfect reconstruction. The test images used in this paper are given in Figure 8. Test image "random" has only 2 values (0 and 255) with probability 1/2 for each value.

The wavelet coefficients are stored as 16 bit integers in Q4 format. This is because 4 is the maximum number of fractional bits that does not cause overflow for lifting based implementations (lifting scheme and reduced scaling lifting scheme). We used C to implement the algorithms and compiled and ran them on a Pentium III PC using gcc 2.95 running Linux to compare the quantization errors. We also compiled and ran the algorithms on a TMS320C6x simulator with Map 1 configuration and "-o3 -k -c –gp" options to get the cycle times [8].

Table 2 shows the quantization errors that are introduced after 6 level floating-point polyphase DWT and fixed-point IDWT operations for each fixed-point implementation. It shows that the polyphase form introduces the least amount of quantization error of the three reconstruction implementations.

Table 3 shows the quantization errors that are introduced after 6 level fixed-point DWT and floating-point polyphase IDWT operations for each fixed-point DWT implementation. It shows that the reduced scaling lifting scheme introduces the least amount of quantization error. It can be seen that the reduced scaling lifting scheme is the most consistent in the amount of quantization error it generates.

Table 4 shows the cycle time for 6 level DWT and IDWT for a 256x256 image for the DSP implementation. These numbers can be only seen as an upper bound for each implementation and may differ for different compilers and processors.

4. CONCLUSION

This paper compared three 2D 9x7 DWT fixed-point implementations in terms of quantization error. Experimental results show that the reduced scaling lifting scheme is consistent in terms of lowest quantization error if used as a decomposition implementation. The fixed-point implementations of the 2D DWT are important because of their use on fixed-point digital signal processors that will be used for many of the initial implementations of JPEG 2000. For further research, we could investigate the quantization errors caused by different filter-banks for each 2D DWT implementations.

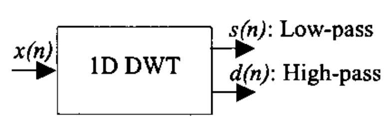


Figure 1. 1D DWT block diagram.

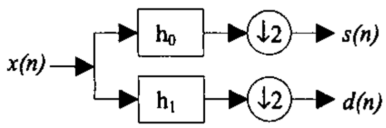


Figure 2. 1D DWT direct implementation.

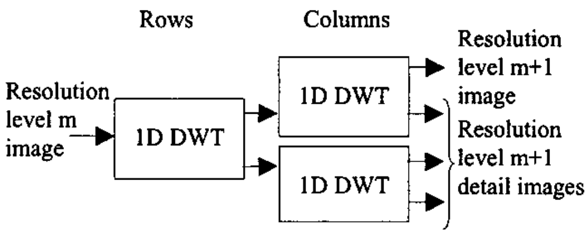
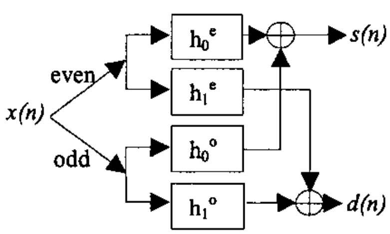


Figure 3. 2D DWT implementation using 1D DWT.



$h_0^e(n) = h_0(2n)$
$h_0^o(n) = h_0(2n+1)$
$h_1^e(n) = h_1(2n)$
$h_1^o(n) = h_1(2n+1)$

Figure 4. 1D DWT polyphase form implementation.

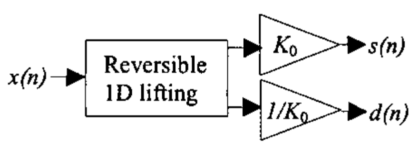
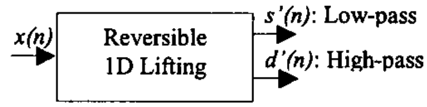


Figure 5. 1D DWT lifting implementation.



$$s^0(n) = x(2n)$$
$$d^0(n) = x(2n+1)$$
$$d^1(n) = d^0(n) + p_1(s^0(n) + s^0(n+1))$$
$$s^1(n) = s^0(n) + u_1(d^1(n-1) + d^1(n))$$
$$d^2(n) = d^1(n) + p_2(s^1(n) + s^1(n+1))$$
$$s^2(n) = s^1(n) + u_2(d^2(n-1) + d^2(n))$$
$$s'(n) = s^2(n)$$
$$d'(n) = d^2(n)$$

Figure 6. Block diagram of reversible 1D lifting for the 9x7 filter-bank.

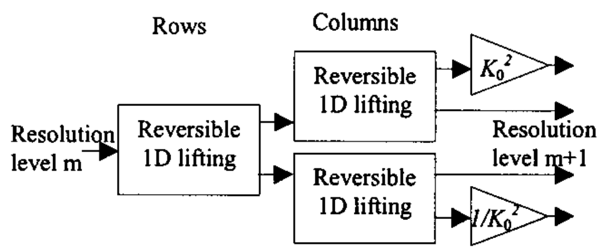
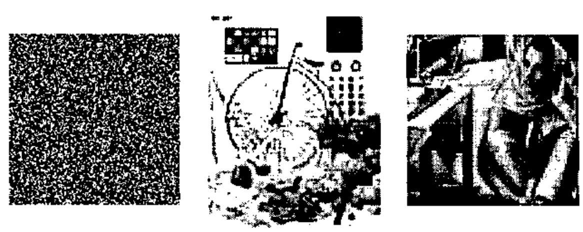


Figure 7. Reduced scaling 2D DWT lifting implementation.



(a) (b) (c)

Figure 8. Test images: (a) random (256x256), (b) bike (2048x2560) and (c) barbara (512x512).

Table 1. Format used for the filters used in polyphase form.

	Q14 format	Q15 format
Filters	h_1^e, h_1^o, g_0^e and g_1^e	h_0^e, h_0^o, g_0^o and g_1^e

Table 2. Quantization errors for 6 level floating-point decomposition and fixed-point reconstruction for test images (a) random, (b) bike and (c) barbara.

	Polyphase	Lifting	Reduced scaling lifting
PSNR (dB)	(a) INF (b) 92.9 (c) 91.9	(a) 72.1 (b) 69.9 (c) 69.4	(c) 76.3 (b) 73.7 (c) 72.1

Table 3. Quantization errors for 6 level fixed-point decomposition and floating-point reconstruction for test images (a) random, (b) bike, and (c) barbara..

	Polyphase	Lifting	Reduced scaling lifting
PSNR (dB)	(a) 56.3 (b) 53.5 (c) 54.3	(a) 71.2 (b) 69.5 (c) 68.1	(c) 77.1 (b) 73.1 (c) 72.5

Table 4. Cycle time for 'C6x for 6 level Mallat decomposition and reconstruction for a 256x256 image..

	Polyphase	Lifting	Reduced scaling lifting
Decomposition	3110749	3745345	3521233
Reconstruction	3116317	3760975	3521995

5. REFERENCES

[1] Information technology – JPEG 2000 Image Coding System, *ISO/IEC International Standard 15444-1*, ITU Recommendation T.800, 2000.

[2] M. Rabbani, R. Joshi, "An Overview of the JPEG2000 Still Image Compression Standard," Signal Processing: *Image Communication*, vol. 17, Number 1, January 2002.

[3] M. Antonini, M. Barlaud, P. Mathieu, "Image Coding Using the Wavelet Transform," *IEEE Trans. Image Processing*, pp. 205-220, April 1992.

[4] J.G. Proakis, D.G. Manolakis, *Digital Signal Processing: Principles, Algorithm, and Applications,*New York: Macmillan Publishing company, 3 ed., 1996.

[5] R.C. Calderbank, I. Daubechies, W. Sweldens, B.-L. Yeo., "Wavelet transforms that map integers to integers," *Appl. Comput. Harmon. Anal.*, vol. 5, no. 3, pp. 332-369, 1998.

[6] I. Daubechies, W. Sweldens, "Factoring wavelet transforms into lifting steps," *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247-269, 1998.

[7] C. M. Brislawn, "Classification of nonexpansive symmetric extension transforms for multirate filter banks," *Appl. Comput. Harmon. Anal.*, vol. 3, pp. 337-57, 1996.

[8] Texas Instruments, *TMS320C6000 Programmer's Guide*, Literature number SPRU198F, February 2001.