Complexity Reduction and Fast Algorithm for 2-D Integer Discrete Wavelet Transform Using Symmetric Mask-Based Scheme

Chih-Hsien Hsia[1], Jing-Ming Guo[2], and Jen-Shiun Chiang[1]
[1] *Multimedia IC Design Lab., Department of Electrical Engineering*
Tamkang University, Taipei, Taiwan.
E-mail: chhsia@ee.tku.edu.tw, chiang@ee.tku.edu.tw

[2] *Multimedia Signal Processing Lab., Department of Electrical Engineering*
National Taiwan University of Science and Technology, Taipei, Taiwan.
E-mail: jmguo@seed.net.tw

Abstract

Wavelet coding has been shown to be better than Discrete Cosine Transform (DCT) in image/video processing. Moreover, it has the feature of scalability, which is involved in modern video standards. This work presents novel algorithms, namely 2-D Symmetric Mask-based Discrete Wavelet Transform (SMDWT), to improve the critical issue of the 2-D Lifting-based Discrete Wavelet Transform (LDWT), and then obtains the benefit of low latency, high-speed operation, and low temporal memory. The SMDWT also has the advantages of high-performance embedded periodic extension boundary treatment, reduced complexity, regular signal coding, short critical path, reduced latency time, and independent subband coding processing. Moreover, the 2-D lifting-based DWT performance can also be easily improved by exploiting appropriate parallel method inherently in SMDWT. Comparing with the normal 2-D 5/3 integer lifting-based DWT, the proposed method significantly improves lifting-based latency and complexity in 2-D DWT without degradation in image quality. The algorithm can be applied to real-time image/video applications, such as JPEG2000, MPEG-4 still texture object decoding, and wavlet-based Scalable Video Coding (SVC).

Keywords: critical path, latency, discrete wavelet transform, lifting-based discrete wavelet transform, symmetric mask-based discrete wavelet transform, temporal memory.

1. Introduction

In recent years, the development of communication and multimedia is more and more speedy. A variety of digital media and services are around our daily life, such as digital camera, VCD, DVD, HDTV, and video conference. Some well-known compression schemes, such as Differential Pulse Code Modulation (DPCM)-based method [1], DCT-based methods [2], [3], and Wavelet-based methods [4]-[13] have been well-developed in the past few decades. Recently, the lifting-based scheme offers a low-complexity solution for image/video applications, e.g., JPEG2000 [7]-[13] or MC-EZBC [19]. However, the real-time 2-D DWT is still difficult to be achieved. Hence, an efficient transformation scheme for the large amount of multimedia is highly demanded.

Filter banks for the applications of subband image/video coding were introduced in the 1990s. Since then, the wavelet coding has been studied extensively. There has been great success in applying wavelet coding to many applications. The most notable applications include subband coding for audio, image, and video, signal analysis and representation using wavelets, and so forth. In the past few years, DWT [4] has been used for a wide range of applications including image coding and video compression, such as speech analysis, numerical analysis, signal analysis, image coding, pattern recognition, computer vision, biometric, and etc. The DWT can be viewed as a multiresolution decomposition of a signal, which means that it decomposes a signal into several components in different wavelet frequency bands. Moreover, the 2-D DWT is a modern tool for compression, such as JPEG2000 still images compression standard, denoising, region of interest, and watermarking applications.

Low temporal memory requirement and latency reduction are the major concerns in 2-D DWT implementation. In this work, a new approach, 2-D Symmetric Mask-based DWT algorithm (SMDWT), is proposed to improve the 2-D lifting-based DWT (LDWT), and is further applied to 2-D DWT real-time applications.

The rest of this paper is organized as follows. In Section 2, the LDWT is briefly introduced. The proposed SMDWT is presented in Section 3. Section 4 demonstrates the performance comparisons. The conclusions are given in Section 5.

2. Lifting-based DWT (LDWT)

The filtering and convolution are applied to achieve the signal decomposition in classical DWT. In 1986, Meyer and Mallat [4] found that the orthonormal wavelet decomposition and reconstruction could be implemented in the multiresolution signal analysis framework. The multiresolution analysis is now a standard to construct the orthonormal wavelet bases. In the decomposition process, the low-pass filter *H* and high-pass filter *G* represent the scaling functions and the corresponding wavelets,

0-7695-3058-3/07 $25.00 © 2007 IEEE
DOI 10.1109/ISM.2007.27

respectively. If we have a filter of length four, the corresponding transfer functions of filters H and G can be represented as,

$$H(z)=h_0+h_1z^{-1}+h_2z^{-2}+h_3z^{-3}, \tag{1}$$

$$G(z)=g_0+g_1z^{-1}+g_2z^{-2}+g_3z^{-3}. \tag{2}$$

The downsampling operation is then applied to the filtered results. A pair of filters are applied to the signal to decompose the image into the low-low (LL), low-high (LH), high-low (HL), and high-high (HH) wavelet frequency bands. Suppose an image of size N×N each band is subsampled by a factor of two, so that each wavelet frequency band contains N/2×N/2 samples. The four bands can be combined to create an output image with the same number of samples as the original.

In most image compression applications, the 2-D wavelet decomposition described above can be applied again to the *LL* sub-image, forming four new subband images, and so on to achieve more compact energy in the lower frequency bands.

2.1. Lifting-based DWT Algorithm

The lifting-based scheme proposed by Daubechies and Sweldens [5], [6] requires fewer computations than the conventional convolution-based approach. The lifting-based scheme is an efficient implementation of DWT. Using the lifting-based scheme, the integer operation is easily employed and avoids the problems caused by the finite precision or rounding. The Euclidean algorithm can be used to factorize the poly-phase matrix of a DWT filter into a sequence of alternating upper and lower triangular matrices and a diagonal matrix. In Eq. 3, the variables h(z) and g(z) denote the low-pass and high-pass synthesis filters, which can be divided into even and odd parts to form a poly-phase matrix P(z) as in Eq. 4.

$$\begin{aligned} g(z)&=g_e(z^2)+ z^{-1}g_o(z^2), \\ h(z)&=h_e(z^2)+z^{-1}h_o(z^2). \end{aligned} \tag{3}$$

$$P(z)=\begin{bmatrix} h_e(z) & g_e(z) \\ h_o(z) & g_o(z) \end{bmatrix}. \tag{4}$$

Using the Euclidean algorithm, it recursively finds the greatest common divisors of the even and odd parts of the original filters. Since h(z) and g(z) form a complementary filter pair, P(z) can be factorized into Eq. 5. Therefore, the filter bank can be factorized into three lifting steps.

$$P(z)=\prod_{i=1}^{m}\begin{pmatrix} 1 & si(z) \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & 0 \\ ti(z) & 1 \end{pmatrix}\begin{pmatrix} k & 0 \\ 0 & 1/k \end{pmatrix}. \tag{5}$$

where si(z) and ti(z) are Laurent polynomials corresponding to the prediction and update steps, respectively, and k is a nonzero constant.

A lifting-based scheme composes the following four stages:

1) Split phase: Divide the original signal into two disjoint subsets. Notably, the variable *Xe* stands for the set of even samples and *Xo* stands for the set of odd samples. This is also referred to as the lazy wavelet transform because it does not decorrelate the data, but only subsamples the signal into even and odd samples.

2) Predict phase: The predicting operator P is applied on the subset *Xo* to obtain the wavelet coefficients d[n] as in Eq. 6.

$$d[\mathrm{n}]=Xo[\mathrm{n}]+\mathrm{P}\times(Xe[\mathrm{n}]). \tag{6}$$

3) Update phase: The *Xe*[n] and d[n] can be combined to obtain the scaling coefficients s[n] after an update operator U as in Eq. 7.

$$s[\mathrm{n}]=Xe[\mathrm{n}]+\mathrm{U}\times(d[\mathrm{n}]). \tag{7}$$

4) Scaling: In the last step, we apply the normalization factor on s[n] and d[n] to obtain the wavelet coefficients. Equations 8 and 9 describe the implementation of the 5/3 integer lifting analysis DWT. Equations 8 and 9 are used for calculating the odd coefficients (high-pass coefficients) and even coefficients (low-pass coefficients), respectively.

$$d^*[n]=X(2n+1)-\lfloor X(2n)+X(2n+2)/2\rfloor. \tag{8}$$

$$s^*[n]=X(2n)+\lfloor d(2n-1)+d(2n+1)+2/4\rfloor. \tag{9}$$

Although the lifting-based scheme involves low complexity, the long and irregular data path is the major limitations for the efficient hardware implementation. In addition, the increasing pipeline registers increases the internal memory size of the 2-D DWT architecture [15].

2.2. Lossless 2-D 5/3 lifting-based DWT structure

The 2-D DWT uses a vertical 1-D DWT subband decomposition and a horizontal 1-D DWT subband decomposition to obtain the 2-D DWT coefficients. Hence, the memory requirement dominates the hardware cost and complexity of the architectures for 2-D DWT.

The lifting step associated with the wavelet is shown in Fig. 1. The original signals, s0, d0, s1, d1, s2, d2, and etc., are the original input pixel sequences. Assuming that the original data are infinite in length, the first stage lifting is first applied to update the odd index data s0, s1, …. In Eq. 10, the parameters -1/2 and Hi represent the first stage lifting parameters and outcomes, respectively. After all the odd index data points are calculated, the second stage lifting can be performed with Eq. 11, where the parameters represent the second stage lifting parameters and outcomes, respectively. The variables Hn and Ln are the high-pass and low-pass coefficients. The values of the lifting parameters -1/2, 1, and 1/4 as shown in Fig. 1. are used for the prediction module (H_i), the update (L_i) module and the Kn module (scaling by Kn=1).

$$\mathrm{H}_i= [(\mathrm{s}_i+\mathrm{s}_{i+1})\times-1/2+\mathrm{d}_i]\times\mathrm{K}_{0,} \tag{10}$$

$$L_i= [(H_i+H_{i-1})\times 1/4+s_i]\times K_1, \tag{11}$$

$$k_0 = k_1 = 1. \tag{12}$$

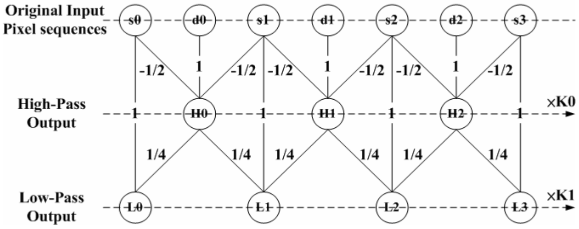


Fig. 1. Lifting-based 5/3 DWT algorithm.

3. The proposed 2-D symmetric mask-based DWT (SMDWT)

The LDWT is widely employed in the image/video subband coding, since it inherently has the well-known perfect reconstruction property. However, in 2-D transform, it has high temporal memory requirement and high critical path. In this work, we present an algorithm which processes one integer 2-D DWT by simply applying one matrix. The SMDWT has many advanced features, such as short critical path, high-speed operation, regular signal coding, and independent subband processing. The 2-D SMDWT is introduced step by step in the following subsections, and the coefficients of mask wavelet coefficient derivation is based on the 2-D 5/3 integer lifting-based DWT.

3.1. The 2-D SMDWT structure

In this sub-section, the proposed SMDWT is discussed in three aspects: lifting structure, transpose memory, as well as latency and critical path. The proposed SMDWT algorithm has the advantages: fast computational speed, reduced latency, lossless compression, and regular data flow.

In speed and simplicity, generally, four masks, 3×3, 5×3, 3×5, and 5×5, are used to perform spatial filtering tasks. Moreover, the four-subband processing can be further optimized to speed up and reduce the temporal memory of DWT coefficients. The four-matrix processors consist of four mask filters, and each filter is derived from one 2-D DWT of 5/3 integer lifting-based coefficients. In DWT implementation, a 1-D DWT needs massive computations and therefore the computation unit dominants the hardware cost [10]-[12]. A 2-D DWT composes two 1-D DWTs and a block of transpose memory, which is of the same size of the processed image. Regarding the computation unit, the transpose memory is the main overhead in the 2-D DWT. The block diagram of a conventional 2-D DWT is shown in Fig. 2. Without loss of generality, the 5/3 lifting-based 2-D DWT is adopted for comparison. Assuming that the image is of size N×N, during the transformation, a large amount of transpose memory (order of N^2) is needed to store the temporary data after the first stage 1-D DWT decomposition. The second stage 1-D DWT is then applied to the stored data to obtain the four-subband (HH, HL, LH, and LL) results of the 2-D DWT. Since the memory requirement of size N^2 is huge and the processing takes too much time, in this work, a new approach, 2-D SMDWT, is introduced to reduce the temporal computing latency and critical path, as shown in Fig. 3(b).

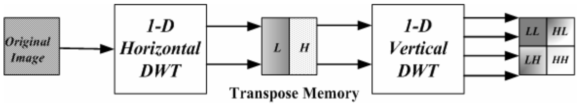
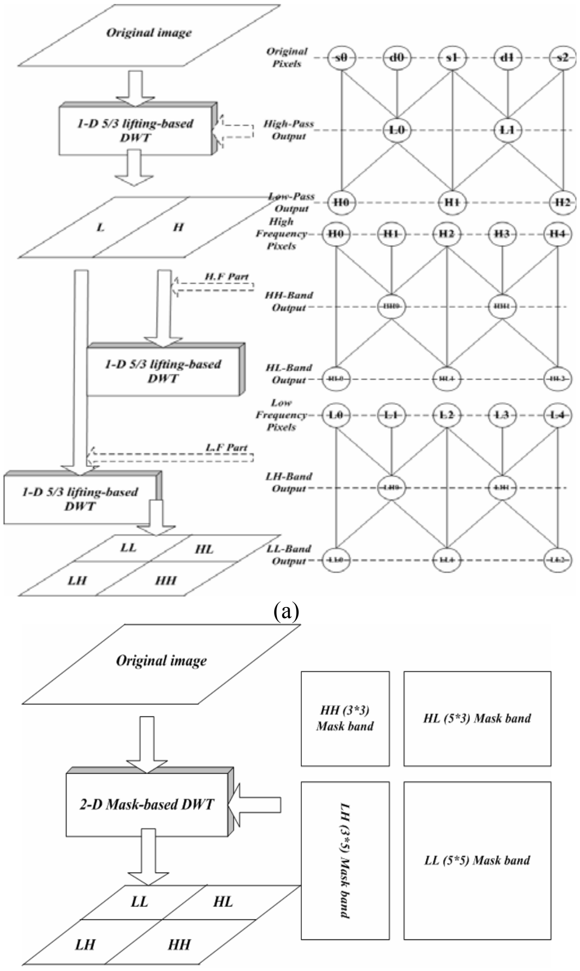


Fig. 2. The block diagram of a conventional 2-D DWT.

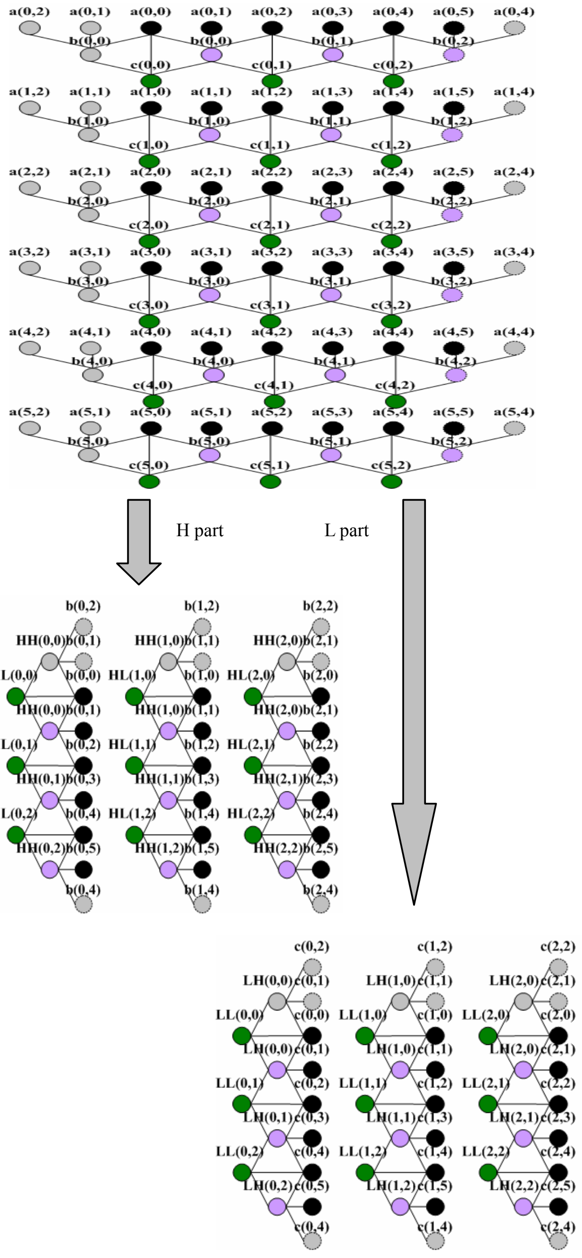


(a)

(b)

Fig. 3. Structure comparisons. (a) 2-D LDWT [18]. (b) 2-D SMDWT.

Without loss of generality, a 6×6-pixel image is employed to demonstrate the 5/3 LDWT operations as shown in Fig. 4. In Fig. 4, the variable $a(i,j)$ represents the original image. The upper part of Fig. 4 shows the first stage 1-D LDWT operations, and the lower part of Fig. 4 shows the second stage 1-D LDWT operations for evaluating the four-subband coefficients, HH, HL, LH, and LL. In the first stage of 1-D LDWT, three pixels are used to evaluate a 1-D high frequency coefficient. For example, $a(0,0)$, $a(0,1)$, and $a(0,2)$ are used to calculate the high frequency wavelet coefficient $b(0,0)$, where $b(0,0)=-[a(0,0)+a(0,2)]/2+a(0,1)$. The pixels, $a(0,2)$, $a(0,3)$, and $a(0,4)$, are used to calculate the next high frequency wavelet coefficient $b(0,1)$. Here $a(0,2)$ is used to calculate both of $b(0,0)$ and $b(0,1)$, and is called an overlapped pixel. The low frequency wavelet coefficient is calculated using two consecutive high frequency wavelet coefficients and the overlapped pixel. For example, $b(0,0)$ and $b(0,1)$ cope with $a(0,2)$ to find the low frequency wavelet coefficient $c(0,1)$, where $c(0,1)=[b(0,0)+b(0,1)]/4+a(0,2)$. The calculated high frequency wavelet coefficients, $b(i,j)$, and the low frequency wavelet coefficients, $c(i,j)$, are then used in the second stage 1-D LDWT to calculate the four subbands coefficients, HH, HL, LH, and LL. The general form of the mask coefficients is derived first, and the complexity is further reduced by employing the symmetric feature of the mask.



a(*i*,*j*): original image, *i* = 0~5 and *j* = 0~5
b(*i*,*j*): high frequency wavelet coefficient of 1-D LDWT
c(*i*,*j*): low frequency wavelet coefficient of 1-D LDWT
HH: high-high frequency wavelet coefficient of 2-D LDWT
HL: high-low frequency wavelet coefficient of 2-D LDWT
LH: low-high frequency wavelet coefficient of 2-D LDWT
LL: low-low frequency wavelet coefficient of 2-D LDWT

Fig. 4. Example of 5/3 LDWT operations.

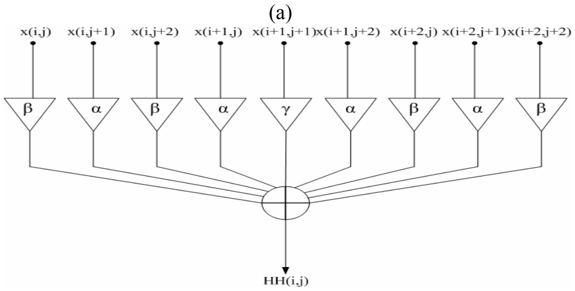
3.2. Simplified 2-D SMDWT using symmetric features

3.2.1. High-High (HH) band mask coefficients reduction for 2-D SMDWT. According to the 2-D 5/3 LDWT, the HH-band coefficients of the SMDWT can be derived as follows:

$$HH(i,j)=x(2i+1,2j+1)+(1/4)\sum_{u=0}^{1}\sum_{v=0}^{1}x(2i+2u,2j+2v)+(-1/2)\sum_{u=-1}^{2}x(2i+2|u|,2j+|1-u|). \quad (13)$$

β	α	β
α	γ	α
β	α	β

(a)



(b)

Fig. 5. HH-band mask coefficients and the corresponding DSP architecture. (a) Coefficients. (b) DSP architecture.

The mask as shown in Fig. 5(a) can be obtained via Eq. 13, where the variables α, β, and γ are -1/2, 1/4, and 1, respectively. Figure 5(b) shows the DSP architecture. The complexity of the mask-based method is further reduced by employing the symmetric feature of the mask. First, the initial horizontal scan is expressed by:

$$HH(0,0)=\beta\times x(0,0)+\alpha\times x(0,1)+\underline{\beta\times x(0,2)}+\alpha\times x(1,0)+\gamma\times x(1,1)+\underline{\alpha\times x(1,2)}+\beta\times x(2,0)+\alpha\times x(2,1)+\underline{\beta\times x(2,2)}. \quad (14)$$

The next coefficient can be calculated by:

$$\begin{aligned}HH(0,1)&=\underline{\beta\times x(0,2)}+\alpha\times x(0,3)+\beta\times x(0,4)+\underline{\alpha\times x(1,2)}+\gamma\times x(1,3)+\alpha\times x(1,4)+\underline{\beta\times x(2,2)}+\alpha\times x(2,3)+\beta\times x(2,4)\\&=\alpha\times x(0,3)+\beta\times x(0,4)+\gamma\times x(1,3)+\alpha\times x(1,4)+\alpha\times x(2,3)+\beta\times x(2,4)+XM_R\\&=\beta\times(x(0,4)+x(2,4))+\alpha\times(x(0,3)+x(1,4)+x(2,3))+\gamma\times x(1,3)+XM_R, \quad (15)\end{aligned}$$

where the variable XM_R denotes the repeated part after the horizontal third coefficient. The general form can be derived as:

$$XM_R=\beta\times x(i,2j+2)+\alpha\times x(i+1,2j+2)+\beta\times x(i+2,2j+2)\,. \quad (16)$$

Since γ=1, the general form can be expressed as:

$$HH(i,j+1)=\beta\times(x(i,2j+4)+x(i+2,2j+4))+\alpha\times(x(i,2j+3)+x(i+1,2j+4)+ x(i+2,2j+3))+x(i+1,2j+3)+XM_R, \quad (17)$$

where i=0~N-1, j=0~N-2.

The vertical scan can be done in the same way, where HH(0,0) is the same as that in Eq. 14. The next coefficient can be calculated by:

$$\begin{aligned}HH(1,0)&=\underline{\beta\times x(2,0)}+\underline{\alpha\times x(2,1)}+\underline{\beta\times x(2,2)}+\alpha\times x(3,0)+\gamma\times x(3,1)\\&\quad+\alpha\times x(3,2)+\beta\times x(4,0)+\alpha\times x(4,1)+\beta\times x(4,2)\\&=\alpha\times x(3,0)+\beta\times x(4,0)+\gamma\times x(3,1)+\alpha\times x(4,1)+\alpha\times x(3,2)+\\&\quad\beta\times x(4,2)+XM_D, \qquad (18)\end{aligned}$$

where the variable XM_D denotes the repeated part after the vertical third coefficient. The general form can be derived as:

$$XM_D=\beta\times x(2i+2,j)+\alpha\times x(2i+2,j+1)+\beta\times x(2i+2,j+2)\ . \qquad (19)$$

Since γ=1, the general form can be expressed as:

$$HH(i+1,j)=\beta\times(x(2i+4,j)+x(2i+4,j+2))+\alpha\times(x(2i+3,j)+x(2i+4, j+1)+x(2i+3,j+2))+x(2i+3,j+1)+XM_D. \qquad (20)$$

where i=0~N-1, j=0~N-2.
Finally, the oblique oriented scan can be derived as:

$$\begin{aligned}HH(1,1)&=\underline{\beta\times x(2,2)}+\underline{\alpha\times x(2,3)}+\underline{\beta\times x(2,4)}+\alpha\times x(3,2)+\gamma\times x(3,3)\\&\quad+\alpha\times x(3,4)+\beta\times x(4,2)+\alpha\times x(4,3)+\beta\times x(4,4)\\&=\gamma\times x(3,3)+\alpha\times x(3,4)+\alpha\times x(4,3)+\beta\times x(4,4)+XM_{RD}\\&=\beta\times x(4,4)+\alpha\times(x(3,4)+x(4,3))+\gamma\times x(3,3)+XM_{RD,}\qquad (21)\end{aligned}$$

where the variable XM_{RD} denotes the repeated part after the vertical fifth coefficient. The general form can be expressed as:

$$XM_{RD}=\beta\times x(2i+2,2j+2)+\alpha\times x(2i+2,2j+3)+\beta\times x(2i+2,2j+4)+\alpha\times x(2i+3,2j+2)+\beta\times x(2i+4,2j+2). \qquad (22)$$

Since γ=1, the general form can be expressed as:

$$HH(i+1,j+1)=\beta\times x(2i+4,2j+4)+\alpha\times(x(2i+3,2j+4)+\beta\times x(2i+4,2 j+3))+x(2i+3,2j+3)+XM_{RD}. \qquad (23)$$

where i=0~N-1, j=0~N-2.

The repeat part is only needed to be calculated once throughout the whole image. Hence it greatly reduces the complexity of the SMDWT.

3.2.2. High-Low (HL), Low-High (LH) and Low-Low (LH) frequency coefficient reduction for 2-D SMDWT. The HL, LH, and LL bands can be derived in the same way. According to the 2-D 5/3 lifting-based DWT, HL, LH, and LL band coefficients of the mask-based DWT can be derived as in Tables 3 to 5.

4. Experimental results and performance comparisons

The proposed 2-D SMDWT algorithm is generally used for performing the 2-D DWT for still image. The schematic diagram of the 2-D SMDWT is shown in Fig. 6. The wavelet transform provides a multiscale representation of image/video in the space-frequency domain. Aside from the energy compaction and decorrelation properties that facilitate compression, a major advantage of the DWT is the scalability. The proposed algorithm is based on the four-subband matrices (HH, HL, LH, and LL) processing to achieve the same performance as the 5/3 LDWT algorithm. The SMDWT has been implemented into JPEG2000 reference software VM 9.0 and is compared with the original JPEG2000. The test image, Lena of size 512×512, is used in this experiment. Experimental results show that the proposed algorithm not only significantly improves lifting-based latency but also has the same visual quality as the normal 2-D 5/3 LDWT [20] as shown in Fig. 7.

The architecture of the 2-D SMDWT has great advantages compared to the 2-D LDWT. For example, the critical path of the 2-D LDWT is potentially longer than that of SMDWT. Moreover, the 2-D LDWT is basically frame-based; the huge amount of the temporal memory size is the bottleneck for implementation. In this work, the symmetric feature of the masks in SMDWT is employed to improve the design. Compared with the other works [7],[10]-[13],[15],[18], experimental results show that the proposed scheme is superior to most of the previous works as shown in Table 7. The proposed algorithm has efficient solutions in reducing critical path (A simple definition of the critical path of the operator is the longest, time-weighted sequence of events from the start of the program to its termination. Figures 8 are examples to show the critical path), latency (The latency is the time between the arrival of a new signal and its first signal output becomes available in the system), and hardware cost, we compute all registers used in XM(need max. is 17 for four-bands) as shown in Table 7, the proposed algorithm has efficient solutions in reducing critical path, latency, and hardware cost, as shown in Figures 5, 8, and Table 6.

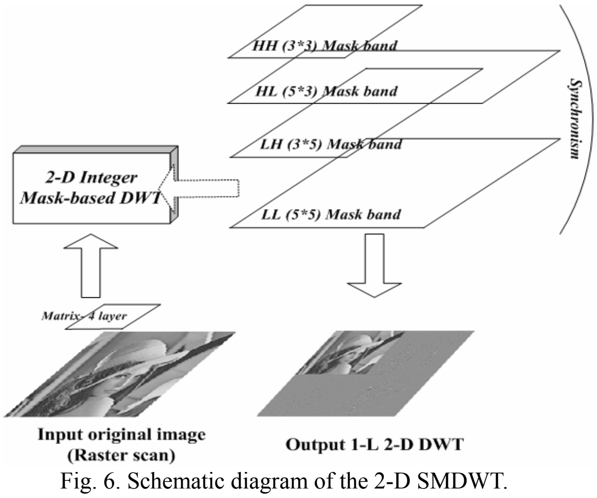


Fig. 6. Schematic diagram of the 2-D SMDWT.

The multilevel DWT computation can be implemented to the proposed 2-D SMDWT in a similar manner. For the multilevel computation, this architecture needs $N^2/2$

off-chip memory. The off-chip memory is used to temporarily store the LL subband coefficients for next iteration computations. The second level computation requires *N*/2 counters and *N*/2 FIFO’s for the control unit. The third level computation requires *N*/4 counters and *N*/4 FIFO’s for the control unit. Generally, in the *j*th level computation, $N/2^{j-1}$ counters and $N/2^{j-1}$ FIFO’s are required. Therefore, the proposed architecture is suitable for multilevel DWT computations. The SMDWT also has the advantages of regular signal coding, short critical path, reduced latency time, and independent subband coding processing. Moreover, the temporal memory access time and power consumption of 2-D LDWT can also be easily improved by SMDWT.

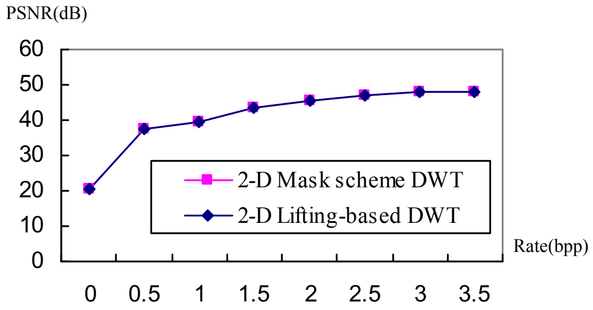
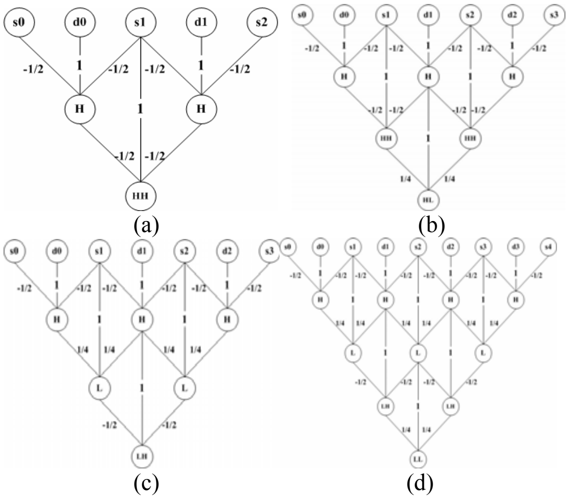


Fig. 7. PSNR (dB) versus Rate (bpp) comparison between 2-D LDWT and the proposed 2-D SMDWT.



(a) (b)

(c) (d)

Fig. 8. 2-D LDWT critical path (a) HH band. (b) HL band. (c) LH band. (d) LL band.

5. Conclusions

A novel 2-D SMDWT fast algorithm, which superiors to the 5/3 LDWT, is proposed in this paper. It solves the latency problem in the previous schemes caused by multiple-layer temporal decomposition operation. Moreover, it provides real-time requirement and can be further applied to the 3-D wavelet video coding [21].

The proposed 2-D SMDWT algorithm has the advantages of fast computational speed, low cpmplexity, reduced latency, regular data flow, and is suitable for VLSI implementation. Moreover, based on the contributions in this work, the research of the future works can be divided as follows:

1) The Dual-Mode 2-D SMDWT on JPEG2000: The dual-mode 2-D SMDWT can be developed to support 5/3 (lossless) lifting and 9/7 (lossy) lifting using similar VLSI architecture, since the 5/3 and 9/7 are very similar and low cost.

2) High Performance JPEG2000 Codec: Since part of the JPEG2000 encoder is symmetric to decoder. The complexity reduction can be considered on both encoder and decoder.

6. References

[1] A. Habibi and R. S. Hershel, “A unified representation of differential pulse code modulation (DPCM) and transform coding systems,” *IEEE Transaction on Communications*, vol. 22, no. 5, pp. 692-696, May 1974.

[2] E. Feig, H. Peterson, and V. Ratnakar, “Image compression using spatial prediction,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 2339-2342, May 1995.

[3] H. Kondo and Y. Oishi, “Digital image compression using directional sub-block DCT,” *International Conference on Communications Technology*, vol. 1, pp.21-25, Augest 2000.

[4] S. G. Mallat, “A theory for multi-resolution signal decomposition: The wavelet representation,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674-693, July 1989.

[5] I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting scheme,” *The Journal of Fourier Analysis and Applications*, vol. 4, pp. 247-269, 1998.

[6] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” *Applied and Computation Harmonic Analysis*, vol. 3, no. 0015, pp.186-200, 1996.

[7] J.-S. Chiang, C.-H. Hsia, H.-J. Chen, and T.-J. Lo, “VLSI architecture of low memory and high speed 2-D lifting-based discrete wavelet transform for JPEG2000 applications,” *IEEE International Symposium on Circuits and Systems*, pp. 4554-4557, May 2005.

[8] J.-S. Chiang and C.-H. Hsia, “An efficient VLSI architecture for 2-D DWT using lifting scheme,” *IEEE International Conference on Systems and Signals*, pp. 528-531, April 2005.

[9] K. Andra, C. Chakrabarti, and T. Acharya, “A VLSI architecture for lifting-based wavelet transform,” *IEEE Workshop on Signal Processing Systems*, pp. 70-79, October 2000.

[10] C. Diou, L. Torres, and M. Robert, ”An embeded core for the 2-D wavelet transform, ” *IEEE on Emerging Technologies and Factory Automation Proceedings*, vol. 2, pp. 179-186, 2001.

[11] S.-C. Chen and C.-C. Wu, “An architecture of 2-D 3-level lifting-based discrete wavelet transform,” *Proceeding of the VLSI Design/ CAD Symposium,* pp. 351-354, August 2002.

[12] K. Andra, C. Chakrabarti, and T. Acharya, “A VLSI architecture for lifting-based forward and inverse wavelet transform,” *IEEE Transactions on Signal Processing*, vol. 50, no.4, pp. 966-977, April 2002.

[13] K.C.B. Tan and T. Arslan, “Shift-accumulator ALU centric JPEG 2000 5/3 lifting based discrete wavelet transform architecture,” *IEEE International Symposium on Circuits and Systems,* vol. 5, pp. V161-V164, May 2003.

[14] B.-F. Wu and C.-F. Lin, “A high-performance and

memory-efficient pipeline architecture for the 5/3 and 9/7 discrete wavelet transform of JPEG2000 codec," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no.12, pp. 1615-1628, December 2005.

[15] W.-T. Lee, H.-Y. Chen, P.-Y. Hsiao, and C.-C. Tsai, "An efficient lifting based architecture for 2-D DWT used in JPEG2000," *Proceeding of the VLSI Design/ CAD Symposium,* pp. 577-580, August 2003.

[16] K. S. S. Srinivasan, "VLSI implementation of 2-D DWT/IDWT cores using 9/7-tap filter banks based on the non-expansive symmetric extension scheme," *IEEE International Conference on VLSI Design*, pp. 435-440. January 2002.

[17] X. Lan, N. Zheng, and Y. Liu," Low-power and high-speed VLSI architecture for lifting-based forward and inverse wavelet transform," *IEEE Transactions on Consumer Electronics*, vol. 51, no.2, pp. 379-385, May 2005.

[18] *JPEG 2000 Part 1 Final Committee Draft Version 1.0, ISO/IEC JTC1/SC29 WG1*, Information Technology, April 2000.

[19] J.-R. Ohm, "Advances in scalable video coding," *Proceedings of The IEEE,* Invited Paper, vol. 93, no.1, pp. 42-56, pp. 42-56, January 2005.

[20] ISO/IEC JTC1/SC29/WG1 Wgln 1684, JPEG 2000 Verification Model 7.0, 2000.

[21] M. Weeks and N. A. Bayoumi, "Three-dimensional discrete wavelet transform architectures," *IEEE Transactions on Signal Processing*, vol. 50, no.8, pp. 2050-2063, August 2002.

Table 1.
The subband mask for DSP.

Mask	Shifters	Clock cycles
HH	8	2
HL	15	2
LH	15	2
LL	25	2

Table 2.
HH-Band wavelet coefficient (mask of size 3×3).

XM_R of HH(i,j+1)	β×x(i,2j+2)+α×x(i+1,2j+2)+β×x(i+2,2j+2).
Complexity reduction	original SMDWT: adder is 8, and multiplier is 9. simplified SMDWT: adder is 6, and multiplier is 2. complexity reduction ratio is around 47.06%.
XM_D of HH(i+1,j)	β×x(2i+2,j)+α×x(2i+2,j+1)+β×x(2i+2,j+2).
Complexity reduction	original SMDWT: adder is 8, and multiplier is 9. simplified SMDWT: adder is 6, and multiplier is 2. complexity reduction ratio is around 47.06%.

Table 3.
HL-Band wavelet coefficient (mask of size 5×3).

XM_{R+1} of HL(i,1)	α×x(i,3)+γ×x(i+1,3)+α×x(i+2,3).
Complexity reduction	original SMDWT: adder is 14, and multiplier is 15. simplified SMDWT: adder is 12, and multiplier is 12. complexity reduction ratio is around 82.76%.
XM_{R+n} of HL(i,j+2)	β×x(i,2j+4)+α×x(i,2j+5)+α×x(i+1,2j+4)+γ×x(i+1,2j+5)+β×x(i+2,2j+4)+α×x(i+2,2j+5).
Complexity reduction	original SMDWT: adder is 14, and multiplier is 15. simplified SMDWT: adder is 9, and multiplier is 9. complexity reduction ratio is around 62.07%.
XM_D of HL(i+1,j)	β×x(2i+2,j)+α×x(2i+2,j+1)+δ×x(2i+2,j+2)+α×x(2i+2,j+3)+β×x(2i+2,j+4).
Complexity reduction	original SMDWT: adder is 14, and multiplier is 15. simplified SMDWT: adder is 10, and multiplier is 10. complexity reduction ratio is around 68.97%.

Table 4.
LH- Band wavelet coefficient (mask of size 3×5).

XM_R of LH(i,j+1)	β×x(i,2j+2)+α×x(i+1,2j+2)+δ×x(2i+2,j+2)+α×x(i+3,2j+2)+β×x(i+4,2j+2).
Complexity reduction	original SMDWT: adder is 14, and multiplier is 15. simplified SMDWT: adder is 10, and multiplier is 10. complexity reduction ratio is around 68.97%.
XM_{D+1} of LH(1,j)	α×x(2i+3,0)+γ×x(2i+3,1)+α×x(2i+3,2).
Complexity reduction	original SMDWT: adder is 14, and multiplier is 15. simplified SMDWT: adder is 12, and multiplier is 12. complexity reduction ratio is around 62.76%.
XM_{D+n} of LH(i+2,j)	β×x(2i+4,j)+α×x(2i+4,j+1)+β×x(2i+4,j+2)+α×x(2i+5,j)+γ×x(2i+5,j+1)+α×x(2i+5,j+2).
Complexity reduction	original SMDWT: adder is 14, and multiplier is 15. simplified SMDWT: adder is 9, and multiplier is 9. complexity reduction ratio is around 62.07%.

Table 5.
LL- Band wavelet coefficient (mask of size 5×5).

XM$_{R+1}$ of LL(i,1)	$\alpha\times x(i,3)+\gamma\times x(i+1,3)+\varepsilon\times x(i+2,3)+\gamma\times x(i+3,3)+\alpha\times x(i+4,3)$.
Complexity reduction	original SMDWT: adder is 24, and multiplier is 25. simplified SMDWT: adder is 20, and multiplier is 20. complexity reduction ratio is around 81.63%.
XM$_{R+n}$ of LL(i,j+2)	$\beta\times x(i,2j+4)+\alpha\times x(i,2j+5)+\alpha\times x(i+1,2j+4)+\gamma\times x(i+1,2j+5)+\delta\times x(i+2,2j+4)+\varepsilon\times x(i+2,2j+5)+\alpha\times x(i+3,2j+4)+\gamma\times x(i+3,2j+5)+\beta\times x(i+4,2j+4)+\alpha\times x(i+4,2j+5)$.
Complexity reduction	original SMDWT: adder is 24, and multiplier is 25. simplified SMDWT: adder is 15, and multiplier is 15. complexity reduction ratio is around 61.22%.
XM$_{D+1}$ of LL(1,j)	$\alpha\times x(3,j)+\gamma\times x(3,j+1)+\varepsilon\times x(3,j+2)+\gamma\times x(3,j+3)+\alpha\times x(3,j+4)$.
Complexity reduction	original SMDWT: adder is 24, and multiplier is 25. simplified SMDWT: adder is 20, and multiplier is 20. complexity reduction ratio is around 81.63%.
XM$_{D+n}$ of LL(i+2,j)	$\beta\times x(2i+4,j)+\alpha\times x(2i+4,j+1)+\delta\times x(2i+4,j+2)+\alpha\times x(2i+4,j+3)+\beta\times x(2i+4,j+4)+\beta\times x(2i+5,j)+\gamma\times x(2i+5,j+1)+\varepsilon\times x(2i+5,j+2)+\gamma\times x(2i+5,j+3)+\alpha\times x(2i+5,j+4)$.
Complexity reduction	original SMDWT: adder is 24, and multiplier is 25. simplified SMDWT: adder is 15, and multiplier is 15. complexity reduction ratio is around 61.22%.

Table 6.
Subband lifting-based v.s. mask-based for integer DWT.

Subbands	LDWT critical path [18]	SMDWT critical path
HH	$2T_M+2T_A$	***$1T_M+1T_A$***
HL	$3T_M+3T_A$	***$1T_M+1T_A$***
LH	$3T_M+3T_A$	***$1T_M+1T_A$***
LL	$4T_M+4T_A$	***$1T_M+1T_A$***

*T_M: Multiplier operation time; T_A: Adder operation time.

TABLE 7. Performance comparisons.

Methods	2-D DWT	Wave stage	Transpose memory	Embedded extension	Latency	Computing time	Complexity
[7]	LDWT	Integer	N	Yes	7	$(3/4)N^2+7$	Simple
[10]	LDWT	Integer	3.5N	No	N/A	N/A	Simple
[11]	LDWT	Integer	2.5N	N/A	N/A	N^2	Complexity
[12]	LDWT	Integer	3.5N	No	2N+5	$(N^2/2)+N+5$	Simple
[13]	LDWT	Integer	3N	Yes	N/A	$(N^2/2)+N+5$	Mediate
[15]	LDWT	Integer	N	Yes	5	$(N^2/2)+5$	Mediate
[18]	LDWT	Integer	N^2	Yes	N/A	N/A	Simple
Proposed	***SMDWT***	Integer	***17***	***Yes***	***2***	***$N^2/2$***	***Simple***

* Transpose memory is used to store high and low-frequency wavelet coefficients in the 2-D DWT.