PAPER *Special Section of Selected Papers from the 15th Workshop on Circuits and Systems in Karuizawa*

Lifting Architecture of Invertible Deinterlacing*

Tatsuumi SOYAMA[†], **Takuma ISHIDA**[†], ***Student Members***, **Shogo MURAMATSU**[†a)], **Hisakazu KIKUCHI**[†], ***and*** **Tetsuro KUGE**[††], ***Regular Members***

SUMMARY Several lifting implementation techniques for invertible deniterlacing are proposed in this paper. Firstly, the invertible deinterlacing is reviewed, and an efficient implementation is presented. Next, two deinterlacer-embedded lifting architectures of discrete wavelet transforms (DWT) is proposed. Performances are compared among several architectures of deinterlacing with DWT. The performance evaluation includes dual-multiplier and single-multiplier architectures. The number of equivalent gates shows that the deinterlacing-embedded architectures require less resources than the separate implementaion. Our experimental evaluation of the dual-multiplier architecture results in 0.8% increase in the gate count, whereas the separate implementation of deinterlacing and DWT requires 6.1% increase from the normal DWT architecture. For the proposed single-multiplier architecture, the gate count is shown to result in 4.5% increase, while the separate counterpart yields 10.7% increase.

key words: *deinterlacing, discrete wavelet transforms, lifting scheme, JPEG2000, intra-frame-based motion picture coding*

1. Introduction

The standard television and video cameras for consumers deal with interlaced scanning videos. Interlaced scanning videos are composed of two fields whose scanlines are acquired at different vertical position at different instant each other [1]–[3]. It is known that, when frame pictures are simply obtained by field interleaving from two successive fields, horizontal comb-tooth artifacts are generated at edges of the moving objects [3]–[5]. If intra-frame-based codec, such as Motion-JPEG2000 [6]–[10], is applied to these frame pictures, flickering is occured at edges of moving objects, because vertical high frequency components associated with comb-tooth artifacts are roughly quantized, especially for low-bit rate coding [11]–[13]. To solve this problem, we have proposed an invertible deinterlacing technique as preprocessing of intra-frame-based motion picture codec, and have developed design methods for filter coefficients [14]–[17]. In this paper, implementation issues of the invertible deinterlacer is mainly discussed, so that a lifting scheme with one for wavelets is obtained.

Wavelet transforms are known to provide a multiresolution analysis functionality and higher-quality picture coding techniques than the discrete cosine transform (DCT) in JPEG. They are employed in the latest still color image compression standard, JPEG2000, in order to serve a lossy/lossless unified compression functionality, and various scalable functionalities [7], [10]. JPEG2000 has the following features: 1) high quality in wide compression range from high bit rate to low bit rate, 2) rich amount of scalabilities due to layered structure, 3) unified lossy/lossless compression functionality, etc. [7], [10]. In JPEG2000 Part-1, the 5/3-filter lifting-based DWT for lossy and lossless compression modes and the 9/7-filter lifting-based DWT for lossy compression mode are defined, and they are chosen according to applications. Until now, several architectures have already been proposed in order to implement both of the 5/3- and 9/7-filter transforms with common hardware resources [18].

In this paper, we consider developing some architectures for invertible deinterlacing. By taking the application to Motion-JPEG2000 into account, we provide efficient implementation schemes based on the conventional DWT architectures, that is, deinterlacer-embedded DWT lifting architectures. Organization of this paper is as follows: Sect. 2 reviews interlaced scanning videos and introduces deinterlacing and reinterlacing with sampling-dencity preservation. Section 3 describes the efficient implementation of deinterlacing and reinterlacing with in-place operations. Section 4 discusses DWT of JPEG2000, and proposes two deinterlacing architectures combined with DWT lifting. Those performances are evaluated by using VHDL models to verify the significance in Sect. 5, followed by conclusions in Sect. 6.

Throughout this paper, the following notations are used:

z: a 3 × 1 vector which consists of variables in a 3-D Z-domain, that is, $\mathbf{z} = (z_0, z_1, z_2)^T$. For progressive ar-

Manuscript received June 27, 2002.
Manuscript revised September 30, 2002.
Final manuscript received November 13, 2002.

[†] The authors are with the Faculty of Engineering, Niigata University, Niigata-shi, 950-2181 Japan.

[††] The author is with the NHK Science & Technical Research Laboratories, Tokyo, 157-0073 Japan.

a) E-mail: shogo@eng.niigata-u.ac.jp

[*] This work was in part supported by the Grand-in-Aid for Scientific Research No.14750283 from Society for the Promotion of Science and Culture of Japan. The design estimation in this study has been derived by VLSI design tools licensed via the VLSI Design and Education Center (VDEC), the University of Tokyo. We would like to appreciate their cooperation.

rays, $z_0^{n_0}$, $z_1^{n_1}$ and $z_2^{n_2}$ denote the variables for the temporal, vertical and horizontal directions, respectively. For these arrays, we express $\mathbf{z}$ as $(z_\mathrm{T}, z_\mathrm{V}, z_\mathrm{H})^T$.

$\mathbf{z}^\mathbf{n}$: a product defined by $\mathbf{z}^\mathbf{n} = z_0^{n_0} z_1^{n_1} z_2^{n_2}$, where $\mathbf{n}$ is a 3×1 integer vector, and n_k denotes the k-th element of $\mathbf{n}$.

$\mathcal{N}(\mathbf{V})$: a set of integer vectors in the fundamental parallelepiped generated with a 3×3 nonsingular matrix $\mathbf{V}$ [19].

$\mathcal{L}(\mathbf{V})$: a set of sample points given by $\mathbf{Vn}$ for all 3×1 integer vectors $\mathbf{n}$, where $\mathbf{V}$ is a 3×3 nonsingular matrix.

$J(\mathbf{V})$: the absolute determinant of $\mathbf{V}$, which equals the number of elements in $\mathcal{N}(\mathbf{V})$.

$\rho(\mathbf{V})$: the inverse of $J(\mathbf{V})$, that is, sampling density [19].

2. Review of Invertible Deniterlacing

In this section, invertible deinterlacing with sampling density preservation is reviewed as a preliminary [14]–[17]. It should be noted that the original array $X(\mathbf{z})$ is assumed to be sampled on lattice $\mathcal{L}(\mathbf{V})$ generated by sampling matrix $\mathbf{V} = \begin{pmatrix} P_\mathrm{T} & P_\mathrm{T} & 0 \\ -P_\mathrm{V} & P_\mathrm{V} & 0 \\ 0 & 0 & P_\mathrm{H} \end{pmatrix}$, where P_T, P_V and P_H are the temporal period between successive fields, the vertical and horizontal sampling periods in a frame, respectively. Figure 1 shows sampling lattice of the original array $X(\mathbf{z})$. This lattice is expressed by a 3×3 matrix, and the sampling density $\rho(\mathbf{V})$ is given as $1/2P_\mathrm{T}P_\mathrm{V}P_\mathrm{H}$.

2.1 Deinterlacing with Sampling Density Preservation

Deinterlacing with sampling density preservation is a kind of 3-dimensional (3-D) sampling-lattice alteration [19], [20]. Figure 2 shows the basic structure, where the circles including $\uparrow \mathbf{Q}$ and $\downarrow \mathbf{R}$ denote the upsampler with factor $\mathbf{Q}$ and downsampler with factor $\mathbf{R}$, respectively. The upsampler converts the interlaced video array $X(\mathbf{z})$ into non-interlaced one. For sampling matrix $\mathbf{V}$, factor $\mathbf{Q}$ has to be $\begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. The downsampling process is applied so that the overall output array $Y(\mathbf{z})$ has the same density as the original, where the ratio $J(\mathbf{R})$ has to be two and the lattice orthogonality has to be kept. $H(\mathbf{z})$ is a 3-D filter, which removes the imaging caused by the upsampler and avoids the aliasing to be caused by the downsampler. One choice of $\mathbf{R}$ is given by $\mathbf{R} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. In this case, $V(\mathbf{z})$ is temporally downsampled. Then, the sampling matrix of $Y(\mathbf{z})$ becomes $\mathbf{V}' = \mathbf{V}\mathbf{Q}^{-1}\mathbf{R} = \begin{pmatrix} 2P_\mathrm{T} & 0 & 0 \\ 0 & P_\mathrm{V} & 0 \\ 0 & 0 & P_\mathrm{H} \end{pmatrix}$. The corresponding sampling lattice is shown in Fig. 3, and the conventional field-interleaving can be regared as a special case of this process.

2.2 Reinterlacing with Sampling Density Preservation

For video codec systems, deinterlaced array $Y(\mathbf{z})$ is encoded, transmitted and then decoded. Especially for high bit-rate decoding, an inverse transform of deinterlacing, which is referred to as reinterlacing in this work, is required to reconstruct interlaced video sequences [15]. Figure 4 shows the basic structure of reinterlacer.

On the assumption that any process is not applied to deinterlaced array $Y(\mathbf{z})$, the relation between $X(\mathbf{z})$ and $\hat{X}(\mathbf{z})$ is derived from Figs. 2 and 4 as follows [14], [16], [17]:

$$\hat{X}(\mathbf{z}) = T\left(\mathbf{z}^{\mathbf{Q}^{-1}}\right) X\left(\mathbf{z}\right) + A\left(\mathbf{z}^{\mathbf{Q}^{-1}}\right) X\left(W_2^{(1\ 1\ 0)}\mathbf{z}\right), \tag{1}$$

where $W_N = e^{-j\frac{2\pi}{N}}$,

$$T(\mathbf{z}) = \frac{1}{4}F(\mathbf{z})H(\mathbf{z}) + \frac{1}{4}F\left(W_2^{(1\ 1\ 0)}\mathbf{z}\right)H\left(W_2^{(1\ 1\ 0)}\mathbf{z}\right), \tag{2}$$

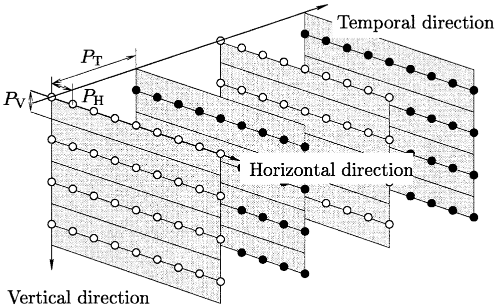


Fig. 1 Interlaced scanning with the sampling lattice $\mathcal{L}(\mathbf{V})$.

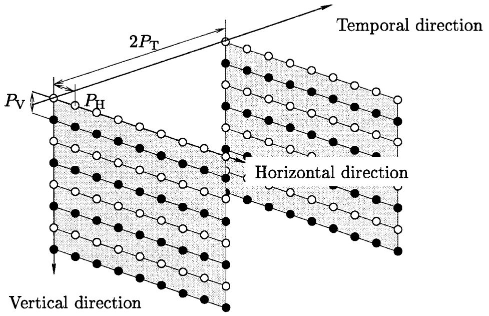


Fig. 3 Sampling lattice $\mathcal{L}(\mathbf{V}')$ of deinterlaced array with temporal decimation.

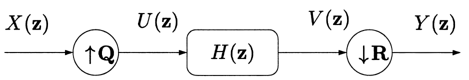


Fig. 2 Basic structure of deinterlacer.

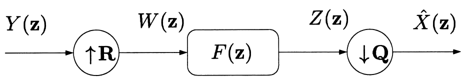


Fig. 4 Basic structure of reinterlacer.

$$A(\mathbf{z}) = \frac{1}{4}F(\mathbf{z})H\left(W_2^{(1\ 0\ 0)}\mathbf{z}\right) + \frac{1}{4}F\left(W_2^{(1\ 1\ 0)}\mathbf{z}\right)H\left(W_2^{(0\ 1\ 0)}\mathbf{z}\right). \quad (3)$$

It can be shown that if the following conditions are satisfied, the source $X(\mathbf{z})$ is perfectly reconstructed, except for delay and scaling [14], [16], [17].

- Alias Free Condition

If $A(\mathbf{z}) = 0$, the alias term in Eq. (1) vanishes. Furthermore, it can be verified that $A(\mathbf{z})$ vanishes if

$$F(\mathbf{z}) = \mathbf{z}^{-\begin{pmatrix} 2S_\mathrm{T}+1 \\ 2S_\mathrm{V} \\ S_\mathrm{H} \end{pmatrix}} KH\left(W_2^{(0\ 1\ 0)}\mathbf{z}\right), \quad (4)$$

where K is an arbitrary non-zero constant, S_T, S_V and S_H are integers. $H\left(W_2^{(0\ 1\ 0)}\mathbf{z}\right)$ means the vertically π-modulated version of $H(\mathbf{z})$.

- Distortion Free Condition

Under the alias free condition, perfect reconstruction is achieved if $T(\mathbf{z})$ is pure delay. The condition that $T(\mathbf{z})$ becomes pure delay is as follows:

$$K\left\{E_{\mathbf{m}_0}(\mathbf{z})\,E_{\mathbf{m}_1}(\mathbf{z}) - \mathbf{z}^{-\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}} E_{\mathbf{m}_2}(\mathbf{z})\,E_{\mathbf{m}_3}(\mathbf{z})\right\} = \mathbf{z}^{-\mathbf{d}}, \quad (5)$$

where $\mathbf{d}$ is an arbitrary 3×1 integer vector. $E_{\mathbf{m}_k}(\mathbf{z})$ is the k-th type-I polyphase component of $H(\mathbf{z})$ with factor $\mathbf{S} = \mathrm{diag}(2, 2, 1)$ such that

$$H(\mathbf{z}) = \sum_{k=0}^{3} \mathbf{z}^{-\mathbf{m}_k} E_{\mathbf{m}_k}\left(\mathbf{z}^{\mathbf{S}}\right),$$
$$\mathbf{m}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \mathbf{m}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \mathbf{m}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \mathbf{m}_3 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}. \quad (6)$$

3. Efficient Implementation with In-Place Operations

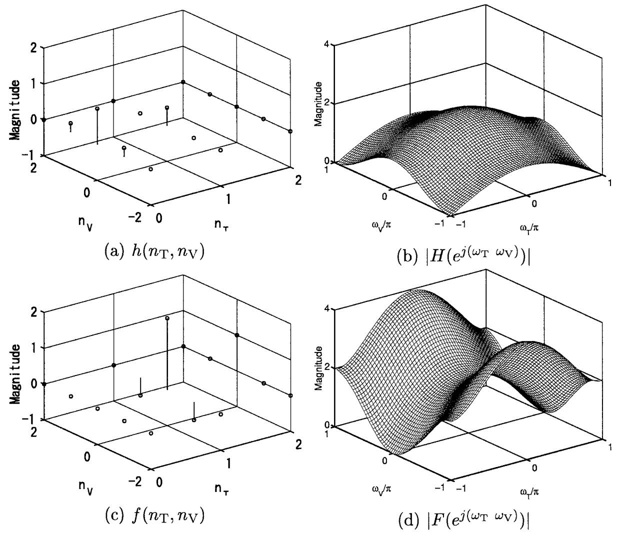
Let us show a special pair of invertible deinterlacing and reinterlacing filters:

$$H(\mathbf{z}) = 1 + \frac{1}{2}z_\mathrm{T}^{-1} + \frac{1}{4}(z_\mathrm{V}^{1} + z_\mathrm{V}^{-1}) \quad (7)$$

$$F(\mathbf{z}) = 2 + z_\mathrm{T}^{-1} - \frac{1}{2}(z_\mathrm{V}^{1} + z_\mathrm{V}^{-1}) \quad (8)$$

These filters are designed for satisfying the perfect reconstruction conditions shown in Eqs. (4) and (5) [14], [16], [17]. In addition, the following constraints are applied in order to give some preferable properties.

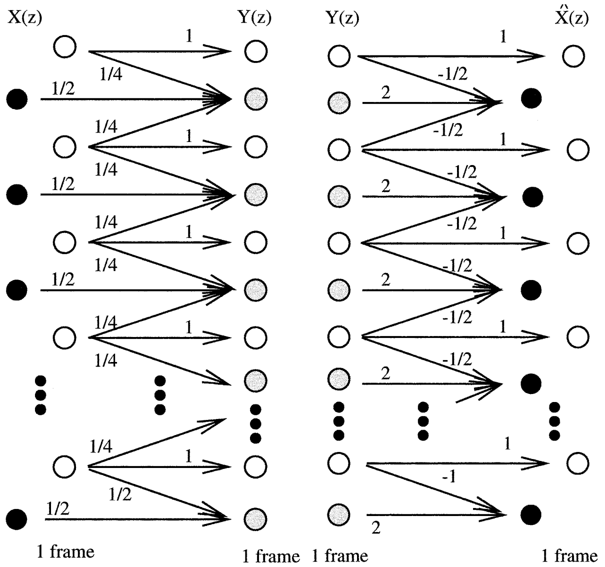
- Normalized amplitude : to keep the luminance of pictures.



(a) $h(n_\mathrm{T}, n_\mathrm{V})$ (b) $|H(e^{j(\omega_\mathrm{T}\ \omega_\mathrm{V})})|$

(c) $f(n_\mathrm{T}, n_\mathrm{V})$ (d) $|F(e^{j(\omega_\mathrm{T}\ \omega_\mathrm{V})})|$

Fig. 5 Impulse and frequency responses of deinterlacing and reinterlacing filters in Eqs. (7) and (8).



(a) Deinterlacing (b) Reinterlacing

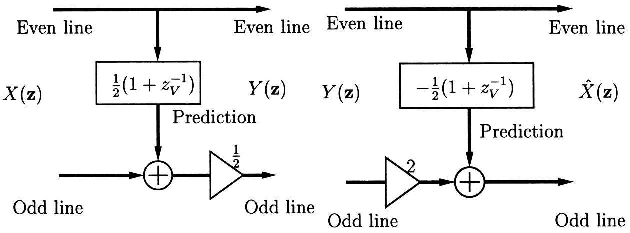
Fig. 6 Efficient operations with symmetric border extension method, where the white, black and gray circles denote even line, odd line of $X(\mathbf{z})$ and odd line of $Y(\mathbf{z})$, respectively.

- Regularity : to avoid the checker board effect by interpolation [21].
- Vertical symmetry : to apply the symmetric border extension method [22].

It is also worth noting that all of the coefficients in Eqs. (7) and (8) are powers of two.

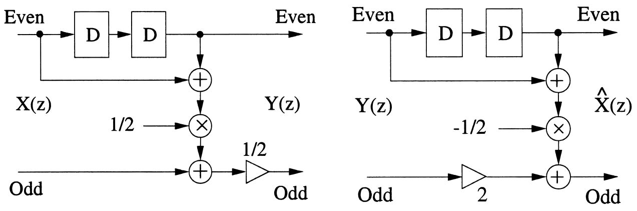
Figure 5 shows impulse and frequency responses of $H(\mathbf{z})$ and $F(\mathbf{z})$, respectively, where n_T, n_V, ω_T and ω_V represent temporal variable, vertical variable, temporal and vertical normalized angular frequencies, respectively.

Efficient deinterlacing and reinterlacing operations



(a) Deinterlacing (b) Reinterlacing

Fig. 7 Lifting implementation of deinterlacing and reinterlacing.



(a) Deinterlacing (b) Reinterlacing

Fig. 8 Lifting architectures for deinterlacer and reinterlacer.

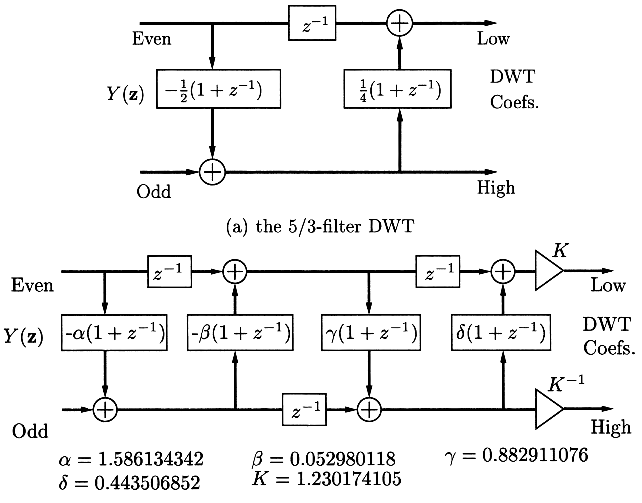
using the filters in Eqs. (4) and (5) are shown in Fig. 6, which allow us to compute them with in-place operations. In Fig. 6, the white, black and gray circles denote even line, odd line of $X(\mathbf{z})$ and odd line of $Y(\mathbf{z})$, respectively. The symmetric border extension method is applied to the vertical direction. The computation can be implemented by bit-shift operations without multipliers, since all of the filter coefficients are powers of two. Additionally, memories can be efficiently used due to the in-place operations, and most of hardware resources can be shared by deinterlacer and reinterlacer, since their structures are symmetric each other. Figure 7 shows structures to realize the operations shown in Fig. 6. Each structure consists of one predictor with Haar-type 2-tap FIR filter. The deinterlacing process can be regarded as vertical lifting calculation for frame pictures after field interleaving. Figure 8 shows hardware architectures for deinterlacing and reinterlacing, respectively, where boxes including 'D' denote delay elements.

4. Lifting Implementation with DWT

In this section, we propose two deinterlacer-embedded DWT lifting architectures, single-multiplier-based (SMUL) and dual-multiplier-based (DMUL) architectures. The resulting architectures allow us to apply deinterlacing to Motion-JPEG2000 (JPEG2000 Part-3) with little overhead.

4.1 DWT of JPEG2000

JPEG2000 Part-1 defines two types of DWTs, the 5/3-filter DWT and 9/7-filter DWT. Their convolution is implemented by lifting computation [7], [10]. Figure 9 shows the structures for the 5/3-filter and 9/7-filter DWTs. The 5/3-filter DWT is constructed by a prediction filter and an update filter. On the other hand, the 9/7-filter DWT is composed of two steps of the prediction and update lifting. Table 1 shows filter coefficients used for the 9/7-filter DWT in the article [18]. These coefficients are represented as the decimal numbers by taking 12-bits for their fractional parts.



(a) the 5/3-filter DWT

(b) the 9/7-filter DWT

Fig. 9 Lifting-based DWT.

Table 1 Filter coefficients of the 9/7-filter DWT, where binary 12-bits are assigned for their fractional parts, and values are represented as decimal numbers.

α	1.5861816	γ	0.8828125
β	0.0529785	δ	0.4436035

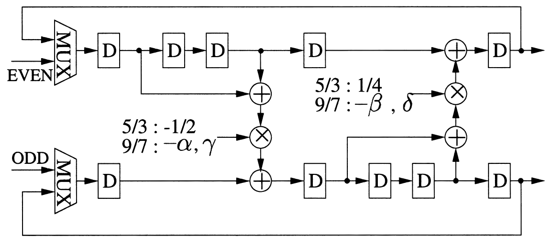


Fig. 10 Dual-multiplier (DMUL) lifting DWT architecture.

Figure 10 shows the conventional architecture for both of the 5/3-filter and 9/7-filter DWTs proposed in the article [18]. This architecture realizes the 5/3-filter DWT as well as the 9/7-filter DWT with the feed back loops. With the architecture shown in Fig. 10, the ratio of resource utilization becomes 100%, and the hardware costs are saved. In the followings, deinterlacer-embedded DWT lifting architecture will be proposed by modifying this architecture.

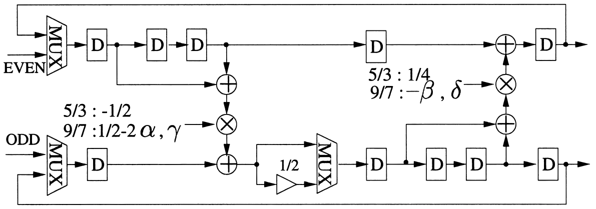


Fig. 11 Deinterlacer-embedded dual-multiplier (DMUL) lifting DWT architecture.

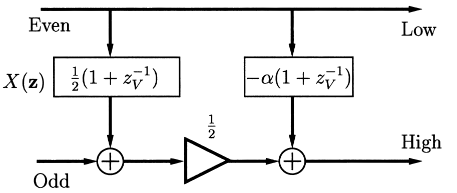


Fig. 12 Lifting implementation of deinterlacer and prediction filter.

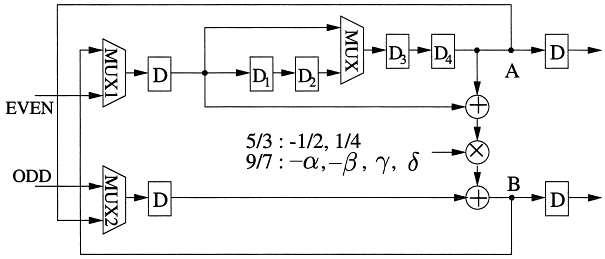


Fig. 13 Single-multiplier (SMUL) lifting DWT architecture.

4.2 Lifting Implementation with DWT

Figure 11 shows the proposed architecutre with dual-multipliers. The proposed architecture provides simultaneous implementation of deinterlacing and lifting-based DWT by embedding the coefficients of deinterlacing into the DWT's prediction filter. In Figs. 10 and 11, the 5/3-filter DWT uses the coefficients −1/2 and 1/4, and the 9/7-filter DWT uses α, β, γ and δ.

The coefficient $1/2 - 2\alpha$ is derived as follows. In Fig. 12,

$$\begin{aligned} High &= \frac{1}{2}\left\{\frac{1}{2}Even\left(1+z_V^{-1}\right)+Odd\right\}-\alpha\left(1+z_V^{-1}\right) \\ &= \frac{1}{2}\left\{\left(\frac{1}{2}-2\alpha\right)Even\left(1+z_V^{-1}\right)+Odd\right\}. \qquad (9) \end{aligned}$$

Additionally, since the architectures of the prediction filter and the update filter are the same as each other, SMUL architecture is also possible by sharing one multiplier and some adders. Figure 13 shows this architecture for lifting DWT. Hardware costs are reduced by saving the number of multipliers. Figure 14 shows the corresponding deinterlacer-embedded lifting DWT architecture. In the SMUL architecture, the number of delay elements working for the 5/3-filter DWT differs from that for the 9/7-filter DWT.

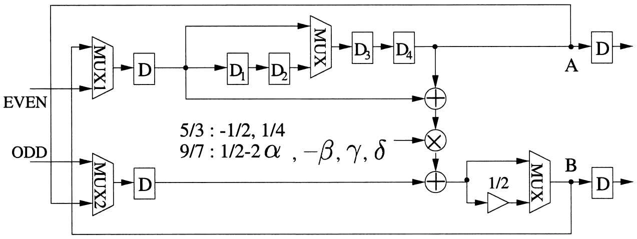
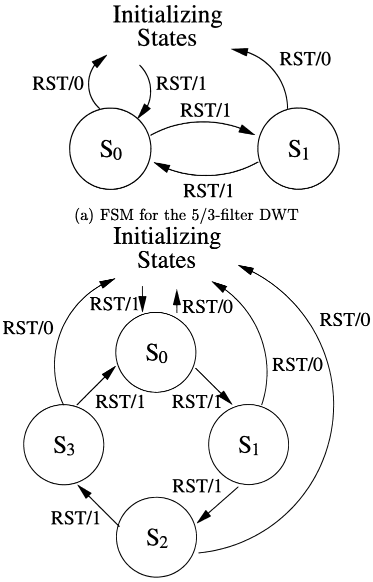


Fig. 14 Deinterlacer-embedded single-multiplier (SMUL) DWT lifting architecture.



(a) FSM for the 5/3-filter DWT

(b) FSM for the 9/7-filter DWT

Fig. 15 Simplified FSM for lifting architectures, where RST/0 and RST/1 denote the states of reset signal RST. (a) FSM for the 5/3-filter DWT (b) FSM for the 9/7-filter DWT

Table 2 Selection of signals for the 5/3-filter DWT in Figs. 13 and 14.

State	Coefs.	Selection of signals	
		MUX1	MUX2
S_0	1/4	EVEN	ODD
S_1	−1/2	B	A

Figure 15(a) shows a simplified finite-state-machine (FSM) for controlling the 5/3-filter DWT shown in Figs. 13 and 14, and the corresponding signal selections are shown in Table 2. Figure 15(b) shows a simplified FSM for controlling the 9/7-filter DWT, and the corresponding signal selections are shown in Table 3.

For the 5/3-filter DWT, delay elements D_1 and D_2 are skipped. Multiplexer MUX1 selects signal EVEN and signal B by turns. Multiplexer MUX2 selects signal ODD and signal A by turns. For the 9/7-filter DWT, all delay elements are used. Multiplexer MUX1 se-

Table 3 Selection of signals for the 9/7-filter DWT in Figs. 13 and 14.

State	Coefs.	Selection of signals	
		MUX1	MUX2
S_0	δ	EVEN	ODD
S_1	$-\alpha$	B	A
S_2	$-\beta$	B	A
S_3	γ	B	A

lects signal EVEN every four cycles, and signal B in the other cases. Multiplexer MUX2 selects signal ODD every four cycles and signal A in the other cases. The SMUL architectures have only one-multiplier. For the 5/3-filter DWT, 2-tap filtering is carried out to obtain two wavelet coefficients, that is, low and high coefficients. Therefore, two delay elements are required to store data in the data path. Thus, D_1 and D_2 are skipped. On the other hand, for the 9/7-filter DWT, 2-tap filtering is carried out two times to obtain two wavelet coefficients. Therefore, four delay elements are required to store data in the data path. Thus, all delay elements are used.

The proposed architectures shown in Figs. 11 and 14 can also carry out only the lifting-based DWT by choosing the filter coefficients for the normal transform, as well as deinterlacing with DWT.

5. Performance Evaluation

Let us verify the significance of our proposed architectures. The performance will be evaluated by using several synthesis results obtained from VHDL models for the architectures. Here, the proposed architectures are compared with the conventional lifting-based DWT ones plus separate deinterlacer. The optimization processes are applied for minimizing the area in this evaluation.

In the following evaluation, Xilinx XCV300 is selected as a target device. The proposed architectures are modeled such that the number of input bits are set to 32, where the integer part is 16 bits and the fractional part is 16 bits. Synopsys Design Analyzer version 2000.05, which is licensed from VDEC, is used for the synthesis [23], and Xilinx Design Manager version v.3.08.i is used as a layout tool.

Table 4 shows the synthesis result of the architecture shown in Fig. 8. To save the hardware costs, the multiplier is modeled by a shift operator. Table 5 shows the synthesis results of the architectures shown in Figs. 10 and 11. These results also include the controller, whose simplified FSM is shown in Fig. 15. In the case of the deinterlacer-embedded architecture, the number of equivalent gates increases only 0.8% from that of the normal lifting-based DWT architecture. When deinterlacing is implemented as a separate module, the number of equivalent gates results in 1,670 as is seen in Table 4, whereas the proposed architecture requires only 217 gate-increase. Table 6 shows the synthesis results of the architectures shown in Figs. 13 and 14. The controllers are also taken into account. With the same way to the previous comparison, the number of equivalent gates is found to be 4.5% increase. Compared with the separate implementation of architectures shown in Figs. 8(a) and 13, the deinterlacer-embedded architecture shown in Fig. 14 is found to require less resources.

Table 4 Synthesis result of deinterlacer (Fig. 8).

The number of equivalent gates	1,670
Critical path [ns]	22.789
Maximum frequency [MHz]	43.881

Table 5 Synthesis results of lifting-based DWT (Figs. 10 and 11).

DMUL Architecture	Deinterlacer	
	Excluded	Embedded
The number of equivalent gates	27,587	27,804
Critical path [ns]	38.404	40.006
Maximum frequency [MHz]	26.039	24.996

Table 6 Synthesis results of single-multiplier architecture (Figs. 13 and 14).

SMUL Architecture	Deinterlacer	
	Excluded	Embedded
The number of equivalent gates	15,641	16,343
Critical path [ns]	38.348	43.154
Maximum frequency [MHz]	26.077	23.173

Table 7 Comparison of latency and throughput of DMUL and SMUL.

	DMUL		SMUL	
	5/3	9/7	5/3	9/7
Latency [cycles]	3	6	3	6
Throughput [cycles]	1	1	1	2

Table 7 shows the comparison of latency and throughput of DMUL and SMUL. For the 5/3-filter DWT, each of DMUL and SMUL is the same. For the 9/7-filter DWT, the throughput of DMUL is half of that of SMUL. It is noticed that there is a tradeoff between the number of gates and throughput.

6. Conclusion

In this paper, invertible deinterlacing was implemented by some lifting architectures, and the performance was evaluated by using the synthesis results obtained from their VHDL models. It was shown that our proposed architectures can carry out both of deinterlacing and DWT lifting as almost the same cost as the usual lifting DWT architecture. The hardware costs can further be saved by employing single-multiplier architecture. Reinterlacer-embedded lifting-based inverse DWT is also implemented by modifying scaling and filter coefficients of normal lifting IDWT architecture.

References

[1] A.M. Tekalp, Digital Video Processing, Prentice-Hall, 1995.

[2] Y. Wang, J. Ostermann, and Y.Q. Zhang, Video Processing and Communications, Signal Processing Series, Prentice-Hall, 2002.

[3] R.A. Beuker and I.A. Shah, "Analysis of interlaced video signals and its applications," IEEE Trans. Image Process., vol.3, no.5, pp.501–512, May 1994.

[4] K. Self, "Prolog to deinterlacing—An overview," Proc. IEEE, vol.9, no.86, pp.1837–1838, Sept. 1998.

[5] G. de Haan and E.B. Bellers, "Deinterlacing—An overview," Proc. IEEE, vol.9, no.86, pp.1839–1857, Sept. 1998.

[6] T. Kuge, "An overview of the new moving picture coding standard motion-JPEG2000," ITE, vol.54, no.12, pp.35–42, Dec. 2000.

[7] D.S. Taubman and M.W. Marcellin, JPEG2000: Image compression fundamentals, standards and practice, Kluwer Academic Publishers, 2002.

[8] ISO/IEC JTC 1/SC 29/WG1 N2117, "Motion JPEG2000 final committee draft 1.0," March 2001.

[9] T. Fukuoka, K. Katoh, S. Kimura, K. Hosaka, and A. Leung, "Motion-JPEG2000 standardizationand target market," Proc. IEEE ICIP, Sept. 2000.

[10] A. Skodras, C. Christopoulis, and T. Ebrahimi, "The JPEG2000 still images compression standard," Proc. IEEE Signal Magazine, vol.18, pp.36–58, Sept. 2001.

[11] T. Kuge and H. Hoshino, "Investigation of HDTV motion picture codec with JPEG2000," Proc. IEICE Gen. Conf. 2001, D-11-18, p.18, March 2001.

[12] T. Kuge, "Several considerations on the visual distortion by wavelet picture coding," IEICE Technical Report, HIR2001-97, Nov. 2001.

[13] T. Kuge, "Wavelet picture coding and its several problems of the application to the interlace HDTV and the ultra-high definition images," Proc. IEEE ICIP 2002.

[14] S. Muramatsu, S. Sasaki, Z. Jie, and H. Kikuchi, "De-interlacing with perfect reconstruction property," Proc. IEICE DSP Symposium, pp.427–432, Nov. 2001.

[15] T. Ishida, S. Muramatsu, Z. Jie, S. Sasaki, and H. Kikuchi, "Intra-frame/field-based motion picture coding with perfect reconstruction de-interlacing," IEICE Technical Report, CAS2001-80, Nov. 2001.

[16] S. Muramatsu, T. Ishida, and H. Kikuchi, "A design method of invertible de-interlacer with sampling density preservation," Proc. IEEE ICASSP, no.IMDSP-L03.01, May 2002.

[17] S. Muramatsu, T. Ishida, and H. Kikuchi, "Invertible deinterlacing with sampling-density preservation: Theory and design," submitted to IEEE Trans. Signal Process., 2002.

[18] C.J. Lian, K.F. Chen, H.H. Chen, and L.G. Chen, "Lifting based discrete wavelet transform architecture for JPEG2000," Proc. IEEE ISCAS, pp.445–448, 2001.

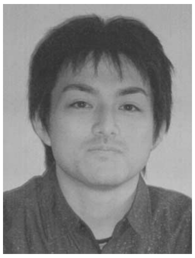
[19] P.P. Vaidyanathan, Multirate Systems and Filter Banks, Prentice-Hall, Englewood Cliffs, 1993.

[20] S. Muramatsu and H. Kiya, "Parallel processing techniques for multidimensional sampling lattice alteration based on overlap-add and overlap-save methods," IEICE Trans. Fundamentals, vol.E78-A, no.8, pp.934–943, Aug. 1995.

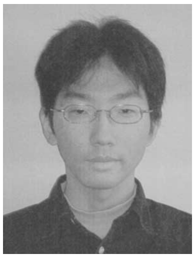
[21] Y. Harada, S. Muramatsu, and H. Kiya, "Multidimensional multirate filter and filter bank without checkerboard effect," IEICE Trans. Fundamentals, vol.E81-A, no.8, pp.1607–1615, Aug. 1998.

[22] H. Kiya, K. Nishikawa, and M. Iwahashi, "A development of symmetric extension method for subband image coding," IEEE Trans. Image Process, vol.3, no.1, pp.78–81, Jan. 1994.

[23] VDEC, http://www.vdec.u-tokyo. ac.jp

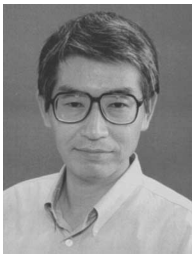


Tatsuumi Soyama was born in Nagano, Japan, in 1979. He received B.E. degree from Niigata University, Niigata, in 2002. He is currently a candidate for the M.E. degree at Niigata University. His research interests are in VLSI architecture and image processing.



Takuma Ishida was born in Niigata, Japan, in 1978. He received B.E. degree from Niigata University, Niigata, in 2001. He is currently a candidate for the M.E. degree at Niigata University. His research interests are in image processing and VLSI architecture.

Shogo Muramatsu was born in Tokyo, Japan, in 1970. He received B.E., M.E., and D.E. degrees in electrical engineering from Tokyo Metropolitan University in 1993, 1995, and 1998, respectively. From 1997 to 1999, he worked at Tokyo Metropolitan University. In 1999, he joined Niigata University, where he is currently an associate professor of Electrical and Electric Engineering, Faculty of Technology. His research interests are in digital signal processing, multirate systems, image processing and VLSI architecture. Dr. Muramatsu is a member of IEEE (Institute of Electrical and Electronics Engineers, Inc.), IPSJ (Information Processing Society of Japan) and ITE (Institute of Image Information and Television Engineers).



Hisakazu Kikuchi received B.E. and M.E. degrees from Niigata University, Niigata, Japan, in 1974 and 1976, respectively, and Dr.Eng. degree in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan in 1988. From 1976 to 1979 he worked at Information Processing Systems Laboratory, Fujitsu, Ltd., Tokyo. Since 1979 he has been with Niigata University, where he is a Professor at Department of Electrical Engineering. During a year of 1992 to 1993, he was a visiting scientist at the California, Los Angeles. His research interests include digital signal processing, image/video processing, and wavelets as well as spread spectrum communication systems. Dr. Kikuchi is a member of IEEE (Institute of Electrical and Electronics Engineers, Inc.), ITE (Institute of Image Information and Television Engineers of Japan), and Japan Society for Industrial and Applied Mathematics, Research Institute of Signal Processing, and SPIE.

Tetsuro Kuge received M.S. degree in electrical engineering from Doshisha University in 1978. Since 1984 he was engaged in research of signal processing and picture coding of HDTV-VTR and stereoscopic HDTV at NHK Science & Technical Research Laboratories. Scince 1995 he was engaged in development of election reporting system at News Department and in development of information processing systems for program production at Multimedia Department. Since 2000 he has been researching JPEG2000 related picture coding for HDTV program production at Science & Technical Research Laboratories. He is a member of ITE Japan, and SC29/WG1 Japan committee.