PAPER *Special Section on Digital Signal Processing*

A Lifting Implementation of Variable-Coefficient Invertible Deinterlacer with Embedded Motion Detector

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

SUMMARY In this paper, a lifting implementation of variable-coefficient invertible deinterlacer with embedded motion detector is proposed. As previous works, the authors have developed invertible deinterlacing that suppresses comb-tooth artifacts caused by field interleaving for interlaced scanning video, which affect the quality of intraframe-based codec such as Motion-JPEG2000. To improve the local adaptability for given pictures, its variable-coefficient processing with motion detection has also been proposed so that filters can be changed according to local properties of motion pictures, while maintaining the invertibility. In this paper, it is shown that the variable-coefficient invertible deinterlacing can be realized by a lifting-based simple hardware architecture, and motion detector can also be embedded. Both of the motion detection and deinterlacing filters are shared by a special choice of their coefficients, and by adaptive selection of deinterlacing filters. The significance of our proposed architecture is verified by showing synthesis results from the VHDL models. The proposed implementation with embedded motion detector achieves about 28% reduction of the gate count compared with the corresponding separate implementation.

key words: *invertible deinterlacing, variable-coefficient filtering, motion-JPEG2000, lifting implementation*

1. Introduction

Interlaced scanning and progressive scanning are known as record and display formats of motion pictures [1]–[3]. For interlaced pictures, such as NTSC signals, its intraframe-based coding requires field interleaving so that some still picture coding is directly applicable. Unfortunately this process causes horizontal comb-tooth artifacts at edges of the moving objects [4]. In the case of scalable transform-based coding such as Motion-JPEG2000 (MJP2), quantization errors yielded by discarding vertical high frequency components of the comb-tooth artifacts become recognizable for low bit-rate decoding. To suppress these artifacts, the intraframe-based coding system with a pre-filter was proposed [4]. This technique is shown to be effective for a target bit-rate decoding. Especially, it is effective for low bit-rate applications.

In high bit-rate applications, the resolution of a filtered picture is not satisfactory because of the low-pass characteristics. As a previous work, to solve this problem, we developed invertible deinterlacing with sampling density preservation as a preprocess of scalable intraframe-based coding [5]–[9]. With this technique, we can suppress the comb-tooth artifacts while maintaining the resolution recovery. However, the first stages of invertible deinterlacing were not necessarily suitable for the local properties of a given picture since the coefficients of the deinterlacing filters were fixed. Later, we proposed invertible deinterlacing with variable coefficients to make it adaptive to the local properties of given pictures [10], [11]. As a result, the invertible deinterlacing was improved in terms of the resolution recovery, while maintaining comb-tooth suppression ability. In the articles [10] and [11], we considered switching between temporal and vertical-temporal (VT) filters according to the output of a motion-detection highpass filter. We also verified the significance by showing comb-tooth-artifact suppression ability as a pre-filter and the coding efficiency.

This work concerns how to implement the adaptive invertible deinterlacer for practical use. So far, we have already developed a lifting-based implementation of invertible deinterlacer with fixed coefficients [12], [13]. It was verified that the hardware architecture can be realized efficiently [12], [13]. For the adaptive invertible deinterlacer, a new problem arises that how we can save hardware resources for the comb-tooth detector, keeping an efficient lifting-based implementation.

In this paper, an implementation issue of variable-coefficient invertible deinterlacer with embedded motion detector is concerned, and a lifting-based architecture is proposed. We show that both of the deinterlacing and motion-detection coefficients can be shared by choosing a special set of filter coefficients. As a result, the whole process is shown to be implemented by a lifting-based simple and efficient hardware architecture. To verify the significance of our proposed implementation, we evaluate the synthesis results obtained from VHDL models of several architectures.

This paper is organized as follows. Section 2 outlines invertible deinterlacer with sampling-density preservation, and describes an adaptive deinterlacing with a motion-detection filter, and summarizes performances in terms of the image qualities. Section 3

Manuscript received November 21, 2002.
Manuscript revised February 28, 2003.
Final manuscript received April 14, 2003.

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

[††] The author is with the Science & Technical Research Laboratories, NHK, Tokyo, 157-8510 Japan.

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

proposes an architecture of the adaptive deinterlacer with embedded motion detector. Section 4 estimates the performance of our proposed architectures by using the synthesis results obtained from VHDL models, followed by our conclusions in Sect. 5.

2. Review of Invertible Deinterlacing

As a previous work, we have proposed a deinterlacing technique that preserves sampling density and has invertibility [5]–[7]. In this section, let us briefly review the invertible deinterlacing with variable coefficients as a preliminary [10], [11]. Here, we use the following notations: $\mathbf{z}$ denotes 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 arrays, we express $\mathbf{z}$ as $(z_\mathrm{T}, z_\mathrm{V}, z_\mathrm{H})^T$.

Assume that the input array $X(\mathbf{z})$ is given by a sampling lattice $\mathcal{L}(\mathbf{V})$ shown in Fig. 1, where $\mathbf{V} = \begin{pmatrix} P_\mathrm{T} & P_\mathrm{T} & 0 \\ -P_\mathrm{V} & P_\mathrm{V} & 0 \\ 0 & 0 & P_\mathrm{H} \end{pmatrix}$. P_T, P_V and P_H are the temporal period between successive fields, the vertical and horizontal sampling periods in a frame, respectively. $\mathcal{L}(\mathbf{V})$ denotes the set of sample points given by $\mathbf{Vn}$ for all 3×1 integer vectors $\mathbf{n}$, where $\mathbf{V}$ is a 3×3 non-singular matrix [14], [15].

2.1 Deinterlacing with Sampling Density Preservation

Figure 2 shows a basic structure of a deinterlacer with sampling density preservation, where the circles including $\uparrow\mathbf{Q}$ and $\downarrow\mathbf{R}$ denote the upsampler with a factor $\mathbf{Q}$ and downsampler with a factor $\mathbf{R}$, respectively [1], [2], [14]–[16]. The upsampler converts the interlaced video array $X(\mathbf{z})$ into a non-interlaced one. For sampling matrix $\mathbf{V}$, factor $\mathbf{Q}$ has to be $\begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. $H(\mathbf{z})$ is a 3-D filter, which removes the imaging caused by the upsampler and avoids the aliasing to be caused by the downsampler. The ratio has to be two to preserve the same density as the original. To keep the lattice orthogonality, one choice of $\mathbf{R}$ is given by $\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. In this case, $V(\mathbf{z})$ is temporally downsampled. Figure 3 shows the sampling lattice $\mathcal{L}(\mathbf{V}')$ of the deinterlaced array $Y(\mathbf{z})$, where $\mathbf{V}' = \mathbf{VQ}^{-1}\mathbf{R} = \begin{pmatrix} 2P_\mathrm{T} & 0 & 0 \\ 0 & P_\mathrm{V} & 0 \\ 0 & 0 & P_\mathrm{H} \end{pmatrix}$.

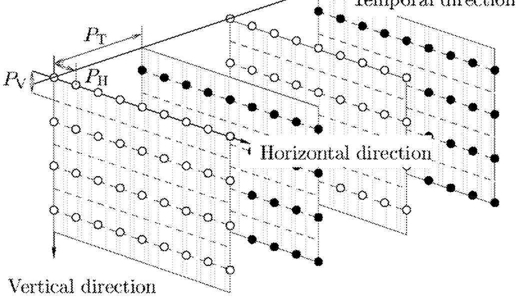


Fig. 1 Interlaced scanning with the sampling lattice $\mathcal{L}(\mathbf{V})$. The white and black circles are sample points on top and bottom fields, respectively.

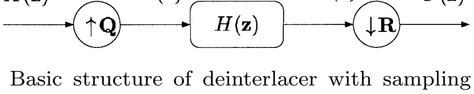


Fig. 2 Basic structure of deinterlacer with sampling density preservation.

2.2 Reinterlacing

For an intraframe-based codec system, the deinterlaced array $Y(\mathbf{z})$ is encoded, transmitted and then decoded frame by frame. Especially for high bit-rate decoding, the interlaced video source is expected to be reconstructed. To achieve this, we have introduced the inverse converter, i.e. *reinterlacer*. Figure 4 illustrates the basic structure of the reinterlacer. In the articles [5], [6] and [7], we showed that the input array $X(\mathbf{z})$ can be perfectly reconstructed from the deinterlaced array $Y(\mathbf{z})$ if $H(\mathbf{z})$ and $F(\mathbf{z})$ satisfy some conditions.

2.3 Intraframe-Based CODEC with Deinterlacer

Let us suggest our scenario of intraframe-based scalable coding such as MJP2 [8], [9]. Figure 5 shows an outline of our suggested codec system. This system uses an invertible deinterlacer as a pre-filter. The comb-tooth artifacts can be suppressed for low bit-rate decoding,

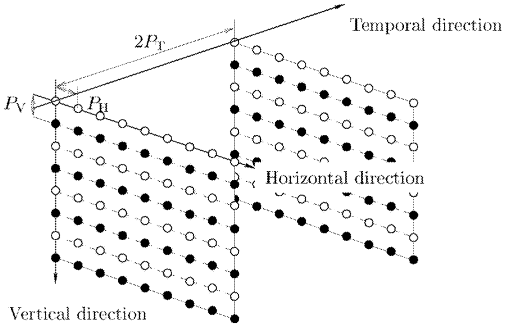


Fig. 3 Sampling lattice $\mathcal{L}(\mathbf{V}')$ of deinterlaced array with temporal decimation.

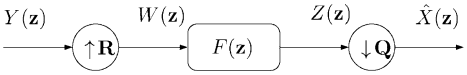


Fig. 4 Basic structure of reinterlacer.

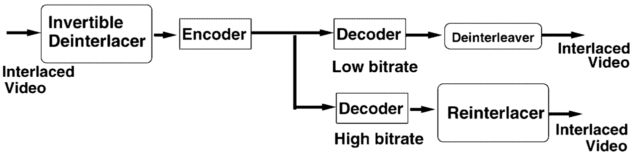
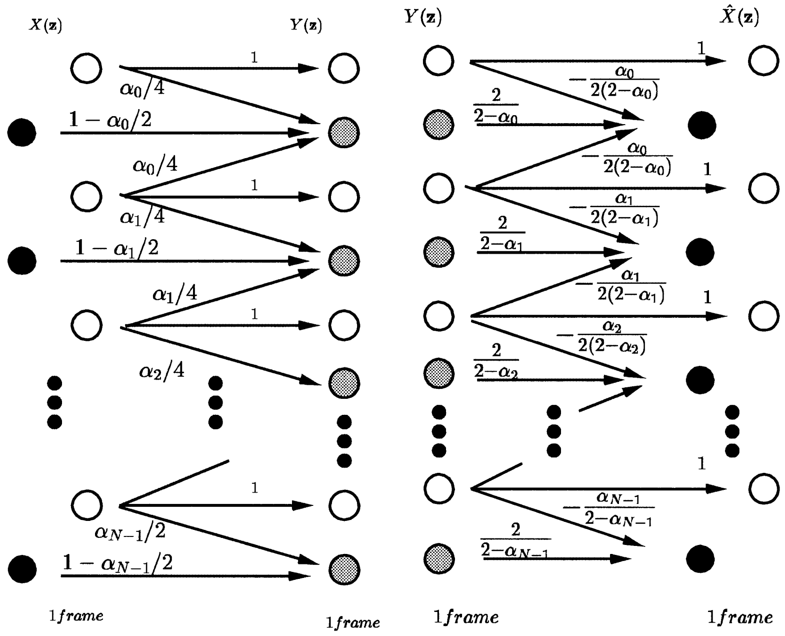


Fig. 5 Intraframe-based coding system with deinterlacer.



(a) Deinterlacer (b) Reinterlacer

Fig. 6 Efficient implementation of deinterlacing with variable coefficients, where the symmetric extension method is applied. The white, black and gray circles denote top line, bottom line of $X(\mathbf{z})$ and odd line of $Y(\mathbf{z})$, respectively.

whereas the resolution can be maintained with reinterlacer for high bit-rate decoding. An example of deinterlacing and reinterlacing filter pair is shown in the followings [5]–[7]:

$$H(\mathbf{z}) = 1 + \frac{1}{2}z_T^{-1} + \frac{1}{4}(z_V^1 + z_V^{-1}). \tag{1}$$

$$F(\mathbf{z}) = z_T^{-1}\left\{2 + z_T^{-1} - \frac{1}{2}(z_V^1 + z_V^{-1})\right\}. \tag{2}$$

These filters have the following properties, i) normalized amplitude to keep brightness, ii) regularity to avoid the checkerboard effect [17], [18] and iii) vertical symmetry to afford the symmetric border extension [17], [19]. We have verified that the comb-tooth artifacts can be avoided for low bit-rate decoding with the deinterlacing filter $H(\mathbf{z})$. However, the quality of still parts is degraded by the process with the filter in Eq. (1). Actually, a simple temporal filter is preferable for still parts.

2.4 Variable-Coefficient Filters

To improve the local adaptability, we also have proposed variable-coefficient invertible filters of deinterlacing and reinterlacing as follows [10], [11]:

$$H_n(\mathbf{z}) = 1 + (1 - \frac{\alpha_n}{2})z_T^{-1} + \frac{\alpha_n}{4}(z_V^1 + z_V^{-1}). \tag{3}$$

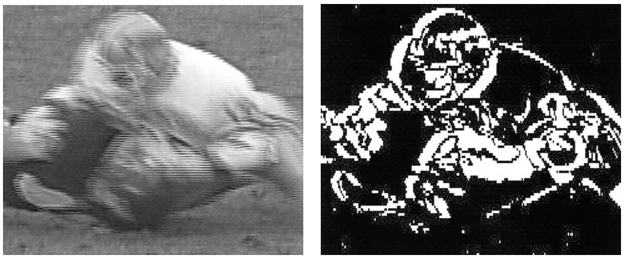
$$F_n(\mathbf{z}) = z_T^{-1}\left\{\frac{2}{2-\alpha_n} + z_T^{-1} - \frac{\alpha_n}{2(2-\alpha_n)}(z_V^1 + z_V^{-1})\right\}. \tag{4}$$

where α_n is a parameter. The characteristics of these filters are controlled among temporal, vertical-temporal (VT) and vertical filters in the range of $0 \leq \alpha_n < 2$. In particular, in the case of $\alpha_n = 1$, the transfer function of the variable-coefficient method is the same as one given by Eq. (1). In addition, the deinterlacer becomes simple field interleaver in the case of $\alpha_n = 0$, so that resolution can be maintained.

Figure 6 illustrates efficient deinterlacing and reinterlacing operations with the filters in Eqs.(3) and (4). The symmetric extension method is applied to the vertical direction. The white, black and gray circles indicate pixels on top line, bottom line of $X(\mathbf{z})$ and odd line of deinterlaced frame $Y(\mathbf{z})$, respectively. The odd line of deinterlaced frame pictures can be obtained by weighted sum of three lines with weights beside arrows. It is verified that our variable-coefficient method can be computed through the in-place implementation shown in Fig. 6. Even if the value of α_n is varied for each sample on odd line, $X(\mathbf{z})$ can be recovered from $Y(\mathbf{z})$ by changing α_n in the same manner as shown in Fig. 6. Note that the property of perfect reconstruction can be kept for this implementation independently from the choice of α_n.

2.5 Adaptive Control Method

In the variable-coefficient deinteralcer, we have to take suitable coefficients according to local properties of given pictures. Problems here are how to choose the coefficients and how to maintain the invertibility. We



(a) Original frame picture. (b) Motion detection result ($T = 16$).

Fig. 7 Example of motion detection.

Table 1 Performances of invertible deinterlacing [10], [11].

	Comb-tooth suppression (Low bit-rate)	Coding efficiency (High bit-rate)
Field interleaving	Poor	Good
Variable-coefficient	Good	Moderate
Fixed-coefficient	Good	Not good

show a switching control method. The parameter α_n can arbitrarily be selected in the range of $0 \leq \alpha_n < 2$. It is, however, necessary to provide the information of α_n to decoders for reinterlacing in the case of high bit-rate decoding. Thus, it is important to reduce the possible quantity of α_n. The reduction of computational complexities is also desired. To achieve these, we proposed to switch the value of α_n in $\{0, 1\}$ in the articles [10] and [11].

In order to detect the comb-tooth artifacts, we considered utilizing a vertical highpass filter $D(\mathbf{z}) = \frac{1}{2}z_T^{-1} - \frac{1}{4}(z_V^{1} + z_V^{-1})$. Note that the coefficients are identical to those in Eq. (1) except for their signs. Therefore, the intermediate values calculated to detect the comb-tooth artifacts with filter $D(\mathbf{z})$ can be shared by the deinterlacing filter. We provide the procedure of motion detection in the followings:

1 Obtain an output frame $I_{x,y}$ of filter $D(\mathbf{z})$.
2 If $|I_{x,2n+1}| \leq$ threshold T, set $\alpha_n = 0$.
Otherwise, set $\alpha_n = 1$.

When $T = 0$, this system results in deinterlacing with fixed coefficients. On the other hand, the calculation is reduced to simple interleaving when $T > 128$ for 8-bit gray scale images. Figures 7(a) and (b) show each frame of original picture and the result of the detected comb-tooth artifacts. The black and white regions show still and motion parts, respectively.

2.6 Performances of Invertible Deinterlacing

In order to show the significance of the adaptive deinterlacing, let us summarize the performances evaluated in the articles [10] and [11].

Table 1 compares the performances in terms of image quality among the adaptive invertible deinterlacing, simple field interleaving and fixed-coefficient deinterlacing. In Table 1, two items are given. One is the comb-tooth-artifact suppression ability, and the other is the coding efficiency.

The former relates to the image quality at low bit-rate decoding when invertible deinterlacers are applied to scalable coding for interlaced video. Field interleaving shows poor result in suppressing comb-tooth artifacts because it corresponds to temporal filtering and has no ability to remove vertical high frequency components. On the other hand, both of the fixed- and variable-coefficient deinterlacing show good results because they remove vertical high frequency components so that the flickering due to comb-tooth artifacts is significantly suppressed.

The latter, coding efficiency, relates to the image quality at high bit-rate decoding. This performance is evaluated via MJP2 in the articles [10] and [11]. Field interleaving is superior to any other deinterlacing in terms of PSNR at the same bit rate because no recovery process is required. Contrary, the fixed-coefficient deinterlacing requires a recovery process, that is reinterlacing, for whole frame pictures. As a result, PSNR becomes worse than that of field interleaving. The variable-coefficient deinterlacing, however, yields moderate results because only some parts prone to be comb-tooth artifacts requires the recovery process, and still parts, e.g. backgrounds, remain without any process after decoding. Experimental results with *Football* sequence show that the field interleaving scores about 1.5 dB in PSNR over the fixed-coefficient deinterlacing at 2.0 bpp decoding for 8-bit grayscale pictures. The difference is shown to be reduced to about 0.5 dB by using the adaptive deinterlacing.

Summarizing, the adaptive deinterlacing is suitable in terms both of the comb-tooth artifact suppression ability and the coding efficiency, and it gives us a compromise between the field interleaving and the fixed-coefficient deinterlacing. See the articles [10] and [11] for details.

3. Proposed Architecture

We propose an efficient lifting implementation of adaptive invertible deinterlacing. The point of this work is that the coefficients of motion detection filter $D(\mathbf{z})$ can be shared with deinterlacing filter in Eq. (1). With this idea, a lifting implementation with motion detector is obtained for our proposed deinterlacer.

3.1 Architecture of Deinterlacer

3.1.1 Transfer Function Level Structure

A basic structure where the deinterlacer and the motion detector consists of separately is shown in Fig. 8. In Fig. 8, the components inside the dotted line are for

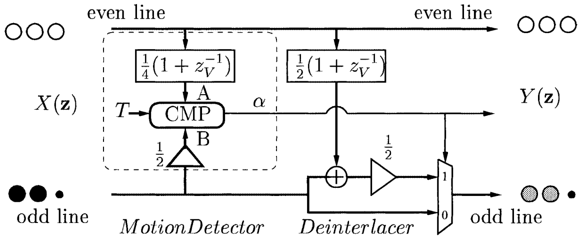


Fig. 8 Lifting structure of separate motion detector.

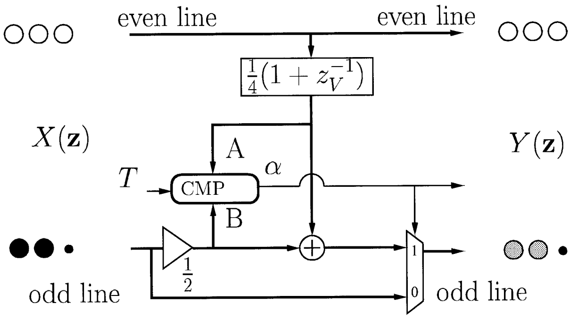


Fig. 9 Lifting structure of embedded motion detector.

motion detection, and the remaining parts are for deinteralcer. CMP indicates a comparator that calculates parameter α, which is actually the control parameter α_n, as follows:

$$\alpha = \begin{cases} 0, & |A-B| \leq T \\ 1, & |A-B| > T. \end{cases} \tag{5}$$

The trapezoid indicates the multiplexor that selects the output signal according to α.

This structure consists of two predictors, which are simple Haar-type 2-tap FIR filters. For these two filtering parts, several components can be implemented in common. By replacing the bit-shifter, i.e. half scaling, on odd line to ahead the adder on odd line in the deinterlacer part, these two predictors can be shared. Figure 9 illustrates our proposed structure. The deinterlacing process can be regarded as a vertical lifting calculation for frame pictures after field interleaving. Note that this efficient lifting structure can be implemented under the perfect reconstruction conditions and the three desirable properties which the fixed-coefficient filter in Eq. (1) possesses (see 2.3).

3.1.2 Architecture of Deinterlacer

Figure 10 illustrates a hardware architecture corresponding to the separated structure shown in Fig. 8. Similarly, Fig. 11 illustrates a hardware architecture corresponding to the structure with embedded motion detector shown in Fig. 9. In Figs. 10 and 11, the boxes including 'D' indicates a vertical delay element. The computation can be implemented only by bit-shift operations without multipliers to save the hardware costs, since all of the filter coefficients are powers of two.

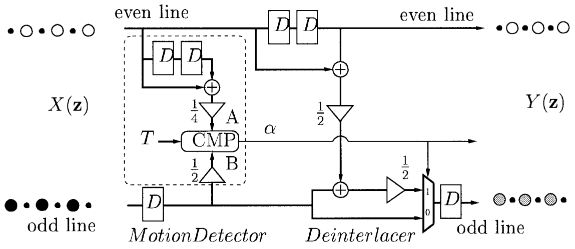


Fig. 10 Architecture of separate motion detector.

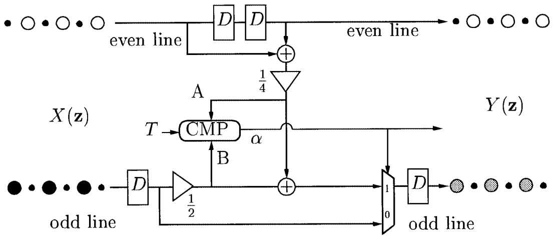


Fig. 11 Architecture of embedded motion detector.

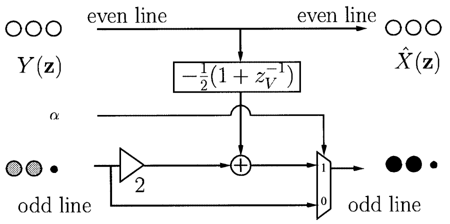


Fig. 12 Lifting structure of reinterlacer.

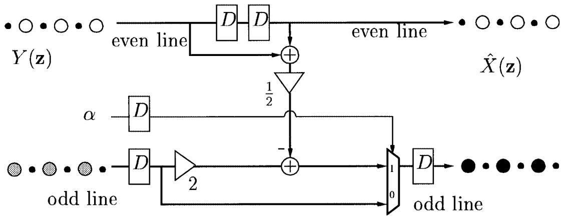


Fig. 13 Architecture of reinterlacer.

3.2 Architecture of Reinterlacer

We have introduced the implementation of reinterlacer with fixed coefficients [12], [13]. For variable-coefficient processing, reinterlacer is also realized by switching reinterlacing filters according to the received α. Figure 12 illustrates a structure of reinterlacer with variable coefficients, and Fig. 13 shows a hardware architecture. From the reinterlacer with fixed coefficients, the adaptive reinterlacer can be realized by adding one multiplexor and input signal α. A simple unified implementation is also obtained by embedding the reinterlacer into the adaptive deinterlacer. Figure 14 illustrates the architecture of the deinterlacer with embedded reinterlacer, where 'MUX0' indicates selector of

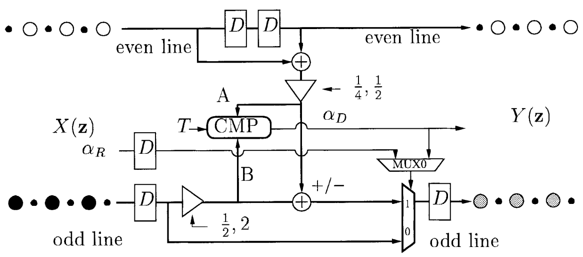


Fig. 14 Architecture of adaptive deinterlacer embedded reinterlacer.

the control-parameters α_D for deinterlacing and α_R for reinterlacing. To save hardware costs, most of hardware resources can be shared by deinterlacer and reinterlacer.

4. Performance Estimation

Let us verify the significance of our proposed architectures. The performances will be evaluated by using several synthesis results obtained from VHDL models. Here, the proposed architectures are compared with the separate architecture. For reference, it is also shown that the synthesis results of deinterlacer and reinterlacer with fixed coefficients. To implement several architectures, XILINX XCV300-5 is selected as a target FPGA device. The optimization processes are applied for minimizing the area in this evaluation.

We assume that the input bit-width is of 32 bits, which includes integer part of 16 bits and fractional part of 16 bits. Synopsys Design Compiler Version 2000.11 licensed from VDEC [20], and XILINX Design Manager Version v.4.1.03i as an FPGA layout tool are used in order to evaluate the equivalent gate count and critical path.

Table 2 shows the synthesis results of the architectures shown in Figs. 10 and 11. The latency is two clocks for all of separate, embedded and fixed-coefficients architectures. Also, the throughput is two clocks for each odd and even line. Thus, for the proposed architecture, the throughput in total results in one clock. When the deinterlacer is implemented with a separate motion detector, the number of equivalent gates increases $2,098$ from the fixed-coefficient one as shown in Table 2, whereas the proposed architecture requires only 803 gate increase. In terms of the number of equivalent gate count, the proposed architecture with embedded motion detector achieves about 28% reduction compared with the separated architecture. As for critical path, the proposed architecture possesses almost the same performance of the separate architecture. For reference, Table 3 shows the synthesis result of the architecture of reinterlacer shown in Fig. 13. The architecture requires only 1.3% gate-increase compared with fixed-coefficient reinterlacer.

Table 2 Synthesis results of deinterlacer on XCV300-5, where E.G.C. means equivalent gate count.

		Variable-coefs	
	Fixed-coefs	Separated	Embedded
E.G.C.	2,558	4,656	3,361
Critical path[ns]	22.789	26.794	28.280

Table 3 Synthesis results of reinterlacer on XCV300-5.

	Fixed-Coefs	Variable-Coefs
E.G.C.	2,567	2,601
Critical path[ns]	15.112	14.750

5. Conclusion

In this paper, a lifting implementation of variable-coefficient invertible deinteralcer with embedded motion detector was proposed. It was shown that a high-pass filter of motion detector can be embedded into deinterlacing filter in a special case. We verified the significance of our proposed architecture by showing the synthesis results obtained from VHDL models and by comparing it with the separate implementation.

Acknowledgment

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 VHDL Design and Education Center (VDEC), the University of Tokyo. We would like to appreciate their cooperation.

References

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

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

[3] G. de Haan and E.B. Bellers, "Deinterlacing—An overview," Proc. IEEE, vol.86, pp.1837–1857, 1998.

[4] T. Kuge, "Wavelet picture coding and its several problems of the application to the interlace HDTV and the ultra-high definition images," IEEE Proc. ICIP, WA-P2.1, 2002.

[5] S. Muramatsu, S. Sasaki, Z. Jie, and H. Kikuchi, "Deinterlacing with perfect reconstruction property," Proc. IEICE DSP Symposium, pp.427–432, 2001.

[6] S. Muramatsu, T. Ishida, and H. Kikuchi, "A design method of invertible de-interlacer with sampling density preservation," IEEE Proc. ICASSP, vol.4, pp.3277–3280, 2002.

[7] S. Muramatsu, T. Ishida, and H. Kikuchi, "Invertible deinterlacer with sampling-density preservation: Theory and design," IEEE Trans. Signal Process., 2003 (to be appeared).

[8] H. Kikuchi, S. Muramatsu, T. Ishida, and T. Kuge, "Reversible conversion between interlaced and progressive scan formats and its efficient implementation," Proc. EUSIPCO, no.448, 2002.

[9] 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, 2001.

[10] T. Ishida, S. Muramatsu, H. Kikuchi, and T. Kuge, "Invertible deinterlacing with variable coefficients," IEICE Technical Report, CAS2002-33, 2002.

[11] T. Ishida, S. Muramatsu, H. Kikuchi, and T. Kuge, "Invertible deinterlacer with variable coefficients and its lifting implementation," IEEE Proc. ICASSP, IMSP-L5.3 (III-105), 2003 (to be presented in ICME2003).

[12] T. Ishida, T. Soyama, S. Muramatsu, H. Kikuchi, and T. Kuge, "Lifting implementation of reversible deinterlacer," Proc. ITC-CSCC 2002, pp.90–93, 2002.

[13] T. Soyama, T. Ishida, S. Muramatsu, H. Kikuchi, and T. Kuge, "Lifting implementation of invertible deinterlacing," IEICE Trans. Fundamentals, vol.E86-A, no.4, pp.779–786, April 2003.

[14] P.P. Vaidyanathan, Multirate Systems and Filter Banks, Prentice-Hall, 1993.

[15] S. Muramatsu and H. Kiya, "Multidimensional parallel processing methods for rational sampling lattice alteration," Proc. IEEE ISCAS, vol.1, no.3, pp.756–759, 1995.

[16] T. Chen and P. Vaidyanathan, "Recent developments in multidimensional multirate system," IEEE Trans. Circuits Syst. Video Technol., vol.3, no.2, pp.116–137, 1993.

[17] G. Strang and T. Nguyen, Wavelets and Filter Banks, Wellesley-Cambridge Press, 1996.

[18] Y. Harada, S. Muramatsu, and H. Kiya, "Multidimensional multirate filter and filter bank without checkerboard effect," Proc. EUSIPCO, pp.1881–1884, 1998.

[19] 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, 1994.

[20] "http://vdec.u-tokyo.ac.jp/," URL.

Takuma Ishida was born in Niigata, Japan, in 1978. He received B.E. and M.E. degrees from Niigata University, Niigata, in 2001 and 2003. He is currently a candidate for the D.E. degree at Niigata University. His research interests are in video processing, picture coding and VLSI architecture.

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.

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. Since 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.