Invertible Deinterlacing With Sampling-Density Preservation: Theory and Design

Shogo Muramatsu, *Member, IEEE*, Takuma Ishida, and Hisakazu Kikuchi, *Member, IEEE*

***Abstract*—A novel class of deinterlacing for intra-frame/field-based motion-picture coding, such as Motion JPEG2000, as well as an inter-frame-based coding without the support of interlaced scanning video, such as MPEG-1, is developed. This technique has two features: sampling-density preservation and invertibility. These features mean that the amount of deinterlaced pictures is not increased, and the original pictures can be perfectly reconstructed. This deinterlacing technique is a kind of sampling-lattice alteration and is regarded as a generalization of conventional field interleaving and field separation. With the help of multidimensional (M-D) multirate theory, it is shown that the design problem of such a system, that is, *invertible deinterlacer with sampling-density preservation*, can be replaced to finding a 2×2 multivariable polynomial matrix with a monomial determinant. This problem resembles the design of two-channel M-D maximally decimated perfect-reconstruction finite-impulse-response (FIR) filterbanks. The inverse system, which is referred to as a *reinterlacer*, is given as FIR when the dual deinterlacing system is FIR. A practical design procedure is provided by suggesting three constraints considered to be preferable: normalization, regularity, and vertical symmetry. The significance of the procedure is verified by showing some design examples of deinterlacing and reinterlacing filters. Simulation results show that the developed method causes fewer comb-shaped artifacts than conventional field interleaving.**

***Index Terms*—Deinterlacing, intra-frame/field-based coding, perfect reconstruction, sampling lattice alteration.**

I. Introduction

RECENT convergence of television, movie, and computer technologies demands that video signals are converted from one format to another. Most broadcasted television systems employ an interlaced scanning format, whereas computer systems usually handle video signals in a progressive scanning format [1]–[3]. Thus, many techniques for converting an interlaced scanning video to a progressive one, as well as spatial-resolution conversion and frame-rate conversion, have been developed so far [1], [2], [4]–[7]. Such techniques are referred to as deinterlacing. Deinterlacing has traditionally been used at advanced television receivers and personal computers to reduce flickering artifacts by doubling the vertical-temporal sampling density. This kind of deinterlacing has been and would be a major problem in the area of visual communication. This paper, however, does not deal with this class of deinterlacing.

There is another class of deinterlacing, such as field interleaving for constructing a frame picture from two fields and field separation for dividing a frame into two field pictures. Although traditional deinterlacing is mainly used as post-processing for displaying video signals, the deinterlacing in question is required prior to or during encoding of an interlaced-scanning video as preprocessing [1], [2], [8]. Unlike traditional post-deinterlacing, the sampling density is preserved, and the original can be perfectly reconstructed. Until now, field interleaving and field separation have not necessarily been regarded as deinterlacing. However, their interpretation as scanning-format conversion brings us a rich amount of novel solutions to improve their performance, which will be dealt with in this paper.

The international standard for motion picture coding MPEG-1 does not support interlaced scanning video. When applying MPEG-1 to such a video sequence, therefore, either field interleaving or field separation is required to construct frame or field pictures before encoding [1], [2], [8]. However, the intra-frame/field preprocessing suffers from vertical-temporal aliasing [3]. In particular, comb-shaped (or serration) artifacts appear at the edges of motion objects and degrade the image quality when intra-frame preprocessing is applied. The more recent standards MPEG-2 and MPEG-4 support interlaced scanning video, and several options concerning the frame/field-prediction for the motion compensation (MC) are defined. As a result, the annoying artifacts are significantly suppressed [1], [2], [8].

As well as MPEG-1, serious problems are caused by intra-frame/field coding techniques, such as Motion JPEG2000 (MJP2) [9]–[12], which originally motivated this work. Intra-frame/field-based motion-picture coding is known to be superior to the inter-frame/field-based coding such as MPEG-1/2/4 in terms of easiness of editing, system complexity, and so forth. The intra-frame/field-based approach requires field interleaving or separation so that still-picture coding becomes directly applicable to an interlaced scanning video signal. This process also causes degradation around the edges of motion objects due to the vertical-temporal aliasing [13]. For example, comb-shaped artifacts consist of vertical high-frequency components, and their quantization in some transform domain, e.g., wavelet transform domain, generates flickering around the edges of motion objects. Here, note that some applications of the intra-frame/field-based approach prefer a simple technique for deinterlacing to an MC technique. For example, digital still cameras are desired to have a single

Manuscript received April 30, 2002; revised February 20, 2003. This work was supported in part by Grant-in-Aid for Scientific Research 14750283 from the Society for the Promotion of Science and Culture of Japan. The associate editor coordinating the review of this paper and approving it for publication was Dr. Helmut Boelcskei.

The authors are with the Department of Electrical and Electronic Engineering, Niigata University, Niigata 950-2181, Japan (e-mail: shogo@eng.niigata-u.ac.jp).

Digital Object Identifier 10.1109/TSP.2003.815368

codec for both still pictures and motion pictures. Thus, it is of interest to use a simple solution to suppress the annoying artifacts.

As mentioned above, field interleaving and separation can be regarded as a scanning format conversion, that is, sampling-lattice alteration (SLA) [1], [2], [14]–[18]. From the theoretical viewpoint, these techniques employ Haar-type linear FIR filters. This fact implies that the ability of filters to suppress the aliasing such as comb-shaped artifacts is quite poor. To solve the aliasing problem, the pre-filtering approach was investigated [13]. We consider, however, that this approach is not accepted without invertibility when spatial resolution lost by the preprocessing has to be recovered. Let us assume embedded coding with scalable functionalities, such as MJP2. Some applications require to decode high-quality video, while low-bit rate applications, such as mobile devices and thumbnail representation of video retrieval systems, require to suppress the annoying artifacts. To achieve both purposes simultaneously, it is worth discussing to find an invertible deinterlacer with the effect as a prefilter. Note that for coding applications, preservation of sampling density is indispensable.

The purpose of the current study is to develop a novel class of deinterlacing, which preserves sampling density and has invertibility. Theory on invertibility and filter design problem will be discussed. The organization of this paper is as follows: Section II reviews interlaced scanning and introduces deinterlacing with sampling-density preservation, which generalizes field interleaving and separation. Section III discusses theorems with regard to the invertibility of SLA systems. Section IV clarifies how to construct a reinterlacer for a given invertible deinterlacer. Section V describes practical design procedures and verifies the significance by showing some design examples of filters and simulation results of intra-frame processing, followed by the conclusions in Section VI. An Appendix is also given so that the reader can follow the derivation of our proposed design procedure.

Throughout this paper, most of the following notations are inherited from [14]–[16] and [19]–[23]:

D — Number of dimensions.

$\mathcal{N}$ — Set of all $D \times 1$ integer vectors.

$\mathbf{z}$ — $D \times 1$ vector that consists of variables in a D-dimensional z-domain, that is, $\mathbf{z} = (z_0, z_1, z_2, \ldots, z_{D-1})^T$. For 3-D progressive arrays, z_0, z_1, and z_2 denote the variables for the temporal, vertical, and horizontal directions, respectively. For these arrays, we express $\mathbf{z}$ as

$$\begin{pmatrix} z_{\mathrm{T}} \\ z_{\mathrm{V}} \\ z_{\mathrm{H}} \end{pmatrix}.$$

$\mathbf{z}^{\mathbf{n}}$ — Product defined by $\mathbf{z}^{\mathbf{n}} = z_0^{n_0} z_1^{n_1} z_2^{n_2} \cdots z_{D-1}^{n_{D-1}}$, where $\mathbf{n}$ is a $D \times 1$ integer vector, and n_k denotes the kth element of $\mathbf{n}$.

For nonsingular real matrix $\mathbf{V}$

$\mathcal{L}(\mathbf{V})$ — Set of all vectors of the form $\mathbf{Vn}, \mathbf{n} \in \mathcal{N}$.

$\rho(\mathbf{V})$ — Inverse of absolute determinant of $\mathbf{V}$, that is, $1/|\det(\mathbf{V})|$. This is equal to the sampling density of $\mathcal{L}(\mathbf{V})$.

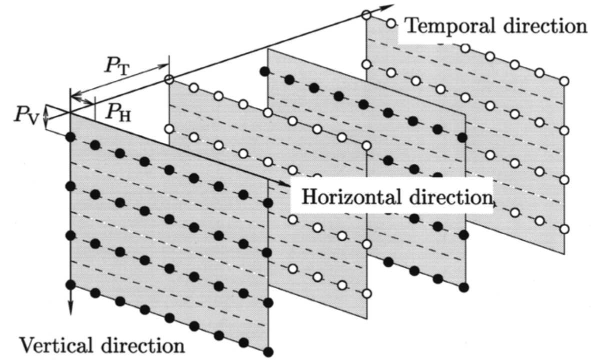


Fig. 1. Interlaced scanning with the sampling lattice $\mathcal{L}(\mathbf{V})$ generated by (1). The white and black circles are the sample points on top (even) and bottom (odd) fields, respectively.

For nonsingular integer matrix $\mathbf{Q}$

$\mathcal{N}(\mathbf{Q})$ — Set of integer vectors of the form $\mathbf{Qx}, \mathbf{x} \in [0,1)^D$, where $[0,1)^D$ is the set of $D \times 1$ real vectors $\mathbf{x}$ with components x_i in the range $0 \le x_i < 1$.

$J(\mathbf{Q}) = |\det \mathbf{Q}|$ — Absolute determinant of $\mathbf{Q}$. This is also equal to the number of elements in $\mathcal{N}(\mathbf{Q})$.

The z-transform of a D-dimensional array $x(\mathbf{n})$ is defined by

$$X(\mathbf{z}) = \sum_{\mathbf{n} \in \mathcal{N}} x(\mathbf{n}) \mathbf{z}^{-\mathbf{n}}.$$

II. Review of Deinterlacing

As a preliminary, interlaced scanning is reviewed here, and a deinterlacing technique with sampling-density preservation is introduced. We consider two types of processing: temporal and vertical downsampling. These techniques can be regarded as a generalization of field interleaving and separation, respectively [1], [2], [8].

A. Interlaced Scanning

In this work, it is assumed that the input array $X(\mathbf{z})$ is sampled on the lattice $\mathcal{L}(\mathbf{V})$ generated by the following sampling matrix [1], [2], [14], [24]:

$$\mathbf{V} = \begin{pmatrix} P_{\mathrm{T}} & P_{\mathrm{T}} & 0 \\ -P_{\mathrm{V}} & P_{\mathrm{V}} & 0 \\ 0 & 0 & P_H \end{pmatrix} \tag{1}$$

where P_{T} is the temporal period between successive fields, and P_{V} and P_H are the vertical and horizontal sampling periods in a frame, respectively. Fig. 1 shows the sampling lattice. Motion pictures are regarded as a 3-D array; thus, the size of the sampling matrix is 3×3. Sampling density $\rho(\mathbf{V})$ is given as $1/2P_{\mathbf{T}}P_{\mathrm{V}}P_H$.

B. Sampling-Density Preservation

Deinterlacing with sampling-density preservation is a kind of 3-D SLA [1], [2], [14]–[18]. Fig. 2 shows the basic structure of SLA systems, where the circles including $\uparrow \mathbf{Q}$ and $\downarrow \mathbf{R}$ denote the upsampler with a factor $\mathbf{Q}$ and the downsampler with a factor $\mathbf{R}$ [14]–[18], [21], [24], [25]. $H(\mathbf{z})$ is a D-dimensional filter that removes the imaging caused by the upsampler and avoids the aliasing caused by the downsampler.

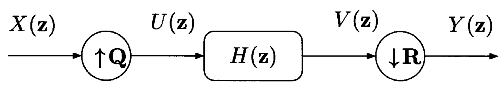


Fig. 2. SLA system, where $\mathbf{Q}$ is given by (2), and $\mathbf{R}$ is given by (3) or (5) for a deinterlacer with sampling-density preservation.

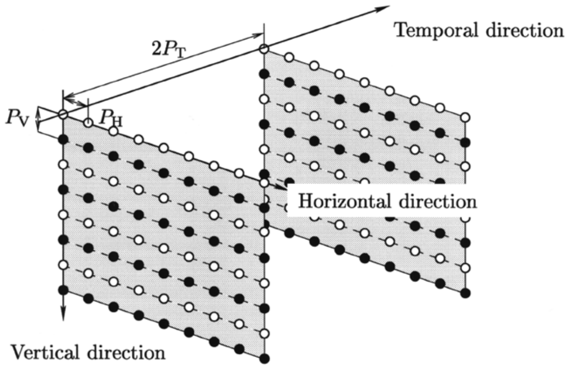


Fig. 3. Sampling lattice $\mathcal{L}(\mathbf{V}')$ of deinterlaced array $Y(\mathbf{z})$ with temporal decimation.

As a deinterlacer, the upsampler with factor $\mathbf{Q}$ converts the interlaced scanning video array $X(\mathbf{z})$ into a progressive one. Under the assumption that the sampling matrix of $X(\mathbf{z})$ is given by (1), factor $\mathbf{Q}$ has to be

$$\mathbf{Q} = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

Then, the upsampled array $U(\mathbf{z})$ is filtered with $H(\mathbf{z})$ to produce the interpolated array $V(\mathbf{z})$ (and to avoid the aliasing caused by the following downsampling). The corresponding sampling matrix is given by $\mathbf{VQ}^{-1} = \mathrm{diag}(P_{\mathbf{T}}, P_{\mathrm{V}}, P_H)$, and the sampling density becomes twice as much as the original $\rho(\mathbf{V})$ because the upsampling ratio $J(\mathbf{Q})$ equals two.

The downsampling is applied to $V(\mathbf{z})$ with factor $\mathbf{R}$ so that the overall output array $Y(\mathbf{z})$ has the same density as the original, where ratio $J(\mathbf{R})$ has to be two, and the lattice orthogonality has to be kept. Two choices of $\mathbf{R}$ are discussed in the followings.

1) Temporal Decimation: One choice of $\mathbf{R}$ is given by

$$\mathbf{R} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3}$$

This factor implies that $V(\mathbf{z})$ is temporally downsampled. Then, the sampling matrix of $Y(\mathbf{z})$ becomes

$$\mathbf{V}' = \mathbf{VQ}^{-1}\mathbf{R} = \begin{pmatrix} 2P_{\mathrm{T}} & 0 & 0 \\ 0 & P_{\mathrm{V}} & 0 \\ 0 & 0 & P_H \end{pmatrix}. \tag{4}$$

The diagonal form implies the orthogonality of the sampling lattice $\mathcal{L}(\mathbf{V}')$. The sampling density $\rho(\mathbf{V}')$ results in $1/2P_{\mathbf{T}}P_{\mathrm{V}}P_H$, which is identical to the original. The corresponding sampling lattice is shown in Fig. 3. When $H(\mathbf{z}) = 1 + z_{\mathrm{T}}^{-1}$, the system becomes the conventional field-interleaver.

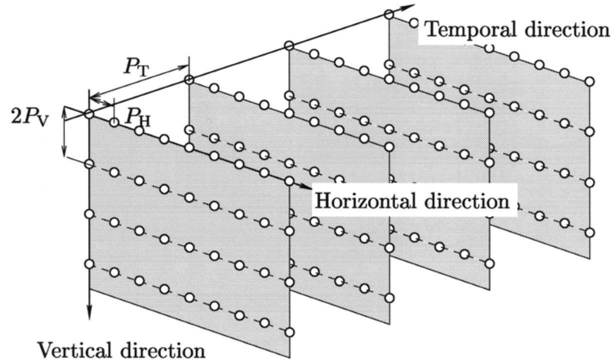


Fig. 4. Sampling lattice $\mathcal{L}(\mathbf{V}')$ of deinterlaced array $Y(\mathbf{z})$ with vertical decimation.

2) Vertical Decimation: Another choice of $\mathbf{R}$ is given by

$$\mathbf{R} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{5}$$

With this choice of $\mathbf{R}$, $V(\mathbf{z})$ is vertically downsampled, and the sampling matrix of the result becomes

$$\mathbf{V}' = \mathbf{VQ}^{-1}\mathbf{R} = \begin{pmatrix} P_{\mathrm{T}} & 0 & 0 \\ 0 & 2P_{\mathrm{V}} & 0 \\ 0 & 0 & P_H \end{pmatrix}. \tag{6}$$

The orthogonality is preserved and the sampling density $\rho(\mathbf{V}')$ becomes identical to the original. The corresponding sampling lattice is shown in Fig. 4. When $H(\mathbf{z}) = 1 + z_{\mathrm{V}}^{1}$, the system corresponds to the conventional field-separator.

Note that the sampling lattice alteration with sampling density preservation is meaningless in one dimension. Thus, the invertible problem to be considered is worth discussing only in the multidimensional case.[1]

III. Invertibility of SLA Systems

For motion-picture coding applications, the deinterlaced array $Y(\mathbf{z})$ is encoded, transmitted, and then decoded. Naturally, the interlaced video source is expected to be reconstructed. To do so, an inverse converter for the deinterlacer is required.

In this section, we investigate the inverse converter and consider the condition for perfect reconstruction (PR), assuming that no process is applied to the deinterlaced array $Y(\mathbf{z})$. To achieve PR, we want to find the conditions under which the relation between $X(\mathbf{z})$ and $\hat{X}(\mathbf{z})$ is reduced to

$$\hat{X}(\mathbf{z}) = c\mathbf{z}^{-\mathbf{d}}X(\mathbf{z}) \tag{7}$$

for $c \neq 0$ and $D \times 1$ integer vector $\mathbf{d}$. In this case, $\hat{X}(\mathbf{z})$ becomes identical to $X(\mathbf{z})$, except for the scaling and delay.

Fig. 5 shows the basic structure of an inverse SLA system. This converter enables us to obtain an array $\hat{X}(\mathbf{z})$ on the same sampling lattice as the original $\mathbf{V}$ because $\mathbf{V}'\mathbf{R}^{-1}\mathbf{Q} = \mathbf{V}$.

[1]Suppose a 1-D multirate system consisting of a succeeding upsampler with factor M, linear time-invariant system $H(z)$, and downsampler with factor M. This system is a 1-D counterpart of the deinterlacer with sampling-density preservation, as shown in Fig. 2. Let $H_k(z)$ be the kth type-I polyphase component of $H(z)$ with factor M [14], [26]. Then, the system function results in the linear time-invariant system $H_0(z)$. This fact implies that the inverse system is given as $H_0^{-1}(z)$, and any multirate theory is not required at all. Actually, if $\mathbf{Q} = \mathbf{R}$, the same result is met even in the MD case ([21], ch. IV).

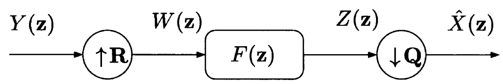


Fig. 5. Inverse SLA system, where $\mathbf{Q}$ and $\mathbf{R}$ have to be identical to those of the corresponding forward SLA system shown in Fig. 2.

Note that SLA systems, as shown in Figs. 2 and 5, are allowed to have any combination of nonsingular integer matrices $\mathbf{Q}$ and $\mathbf{R}$. Sampling lattice $\mathbf{V}$ and dimension D are also arbitrary. In this section, for a general presentation, matrices $\mathbf{V}, \mathbf{Q}$, and $\mathbf{R}$ are not restricted to those given in the previous section. We just assume that matrices $\mathbf{Q}$ and $\mathbf{R}$ are chosen such that i) $J(\mathbf{Q}) = J(\mathbf{R}) = 2$ and ii) $\mathcal{L}(\mathbf{Q}) \neq \mathcal{L}(\mathbf{R})$.

A. Properties of Integer Lattices

Let us review some properties of integer lattices and define some vectors that play an important role through this work. Assumptions i) and ii) imply that $\mathbf{Q}$ and $\mathbf{R}$ are left coprime because,if they had a common left divisor, $\mathbf{D}$, $\mathbf{Q}$, and $\mathbf{R}$ are represented by $\mathbf{Q} = \mathbf{D}\mathbf{U}_{\mathrm{Q}}$ and $\mathbf{R} = \mathbf{D}\mathbf{U}_{\mathrm{R}}$, respectively. In this case, $J(\mathbf{D})$ must be two from the assumption i), and then, $J(\mathbf{U}_{\mathrm{Q}})$ and $J(\mathbf{U}_{\mathrm{R}})$ must be unity. In other words, $\mathbf{U}_{\mathrm{Q}}$ and $\mathbf{U}_{\mathrm{R}}$ must be unimodular. This conflicts with the assumption ii) [14], [15], [24], [25].

In the following, we consider the index space of the sampling lattice for upsampled progressive signals $U(\mathbf{z}), V(\mathbf{z}), W(\mathbf{z})$, and $Z(\mathbf{z})$ in Figs. 2 and 5, that is, the integer vector space $\mathcal{N}$. Integer lattices $\mathcal{L}(\mathbf{Q})$ and $\mathcal{L}(\mathbf{R})$ are clearly sublattices of their ambient space $\mathcal{N}$ and are of index two in $\mathcal{N}$. $\mathcal{L}(\mathbf{Q})$ corresponds to sample points before upsampling with factor $\mathbf{Q}$, and $\mathcal{L}(\mathbf{R})$ represents sample points to be retained after downsampling with factor $\mathbf{R}$.

The intersection lattice $\mathcal{L}(\mathbf{Q}) \cap \mathcal{L}(\mathbf{R})$ is known to be given as $\mathcal{L}(\mathrm{lcrm}(\mathbf{Q}, \mathbf{R}))$, where "lcrm" is the abbreviation of "least common right multiples" [15], [16], [24], [25]. Let one $\mathrm{lcrm}(\mathbf{Q}, \mathbf{R})$ be $\mathbf{S}$. Then, $J(\mathbf{S})$ results in $J(\mathbf{Q})J(\mathbf{R}) = 4$ since $\mathbf{Q}$ and $\mathbf{R}$ are left coprime (see [15, Fact 1, Lemma 5]).

$\mathcal{L}(\mathbf{S})$ is of index four in the integer vector space $\mathcal{N}$. Thus, the space $\mathcal{N}$ can be represented as the union of four cosets of the lattice $\mathcal{L}(\mathbf{S})$, that is

$$\mathcal{N} = \{\mathbf{n} + \mathbf{m} \,|\, \mathbf{n} \in \mathcal{L}(\mathbf{S}), \mathbf{m} \in \mathcal{N}(\mathbf{S})\}$$

where $\mathcal{N}(\mathbf{S})$ has four different elements. The ordering of elements in $\mathcal{N}(\mathbf{S})$ is arbitrary.

Definition: Let us define the ordering of elements in $\mathcal{N}(\mathbf{S})$ such that we have the following.

- $\mathbf{m}_0$ is the zero vector $\mathbf{0}$, which is on the lattice $\mathcal{L}(\mathbf{S}) = \mathcal{L}(\mathbf{Q}) \cap \mathcal{L}(\mathbf{R})$ in $\mathcal{N}(\mathbf{S})$.
- $\mathbf{m}_1$ is the other element on $\mathcal{L}(\mathbf{Q})$ in $\mathcal{N}(\mathbf{S})$.
- $\mathbf{m}_2$ is the other element on $\mathcal{L}(\mathbf{R})$ in $\mathcal{N}(\mathbf{S})$.
- $\mathbf{m}_3$ is the rest in $\mathcal{N}(\mathbf{S})$, that is, on neither $\mathcal{L}(\mathbf{Q})$ nor $\mathcal{L}(\mathbf{R})$.

All through this work, we use the above definition for the ordering of the elements in $\mathcal{N}(\mathbf{S})$.

B. Invertibility of Forward SLA

The invertibility of a forward lattice alteration system, such as a deinterlacer, shown in Fig. 2, is required for constructing a PR pair of SLA systems. Here, let us show that the problem can be reduced to finding an invertible 2×2 multivariable polynomial matrix.

On the ambient space $\mathcal{N}$ of $\mathcal{L}(\mathbf{Q})$ and $\mathcal{L}(\mathbf{R})$, the following filter operation is processed:

$$V(\mathbf{z}) = H(\mathbf{z})U(\mathbf{z}) = H(\mathbf{z})X(\mathbf{z}^{\mathbf{Q}}). \tag{8}$$

Recall that $\mathbf{S}$ is one $\mathrm{lcrm}(\mathbf{Q}, \mathbf{R})$, and there exists an integer matrix $\mathbf{R}'$ such that $\mathbf{S} = \mathbf{Q}\mathbf{R}'$. Then, the relation in (8) can be rewritten as follows:

$$V(\mathbf{z}) = \left\{\sum_{k=0}^{3} \mathbf{z}^{-\mathbf{m}_k} H_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}})\right\} \left\{\sum_{\ell=0}^{1} \mathbf{z}^{\mathbf{Q}\mathbf{n}_\ell} X_{\mathbf{n}_\ell}(\mathbf{z}^{\mathbf{S}})\right\} \tag{9}$$

where $H_{\mathbf{m}_k}(\mathbf{z})$ is the kth type-I polyphase component of $H(\mathbf{z})$ with factor $\mathbf{S}$, and $X_{\mathbf{n}_\ell}(\mathbf{z})$ is the ℓth type-II polyphase component of $X(\mathbf{z})$ with factor $\mathbf{R}'$ [14]–[16], [23] such that

$$H(\mathbf{z}) = \sum_{k=0}^{3} \mathbf{z}^{-\mathbf{m}_k} H_{\mathbf{m}_k}\left(\mathbf{z}^{\mathbf{S}}\right), \quad \mathbf{m}_k \in \mathcal{N}(\mathbf{S}) \tag{10}$$

$$X(\mathbf{z}) = \sum_{\ell=0}^{1} \mathbf{z}^{\mathbf{n}_\ell} X_{\mathbf{n}_\ell}\left(\mathbf{z}^{\mathbf{R}'}\right), \quad \mathbf{n}_\ell \in \mathcal{N}(\mathbf{R}'). \tag{11}$$

$\mathcal{N}(\mathbf{R}')$ has two elements because $J(\mathbf{R}') = J(\mathbf{S})/J(\mathbf{Q}) = 2$. Let us define the ordering of the elements in $\mathcal{N}(\mathbf{R}')$ such that $\mathbf{n}_0$ is the zero vector $\mathbf{0}$, and $\mathbf{n}_1$ is the other element. Then, we have

$$V(\mathbf{z}) = \sum_{k=0}^{3} \mathbf{z}^{-\mathbf{m}_k} H_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}}) X_{\mathbf{n}_0}(\mathbf{z}^{\mathbf{S}}) + \sum_{k=0}^{3} \mathbf{z}^{-(\mathbf{m}_k - \mathbf{m}_1)} H_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}}) X_{\mathbf{n}_1}(\mathbf{z}^{\mathbf{S}}) \tag{12}$$

where we use the fact that $\mathbf{Q}\mathbf{n}_0 = \mathbf{m}_0 = \mathbf{0}$, and $\mathbf{Q}\mathbf{n}_1 = \mathbf{m}_1$. The division theorem for integer vectors allows us to combine the first and second terms [16].

Definitions: For the integer vectors $\mathbf{m}_k \in \mathcal{N}(\mathbf{S})$, let us define a function $f_1(k)$ and integer vectors $\mathbf{g}_1(k) \in \mathcal{N}$ based on the division theorem such that

$$\mathbf{m}_k - \mathbf{m}_1 = \mathbf{S}\mathbf{g}_1(k) + \mathbf{m}_{f_1(k)}, \quad k = 0, 1, 2, 3$$

where $\mathbf{g}_1(k)$ is the quotient of $\mathbf{m}_k - \mathbf{m}_1$ with respect to $\mathbf{S}$, and $\mathbf{m}_{f_1(k)}$ is the remainder in $\mathcal{N}(\mathbf{S})$, i.e., $\mathbf{m}_{f_1(k)} = ((\mathbf{m}_k - \mathbf{m}_1))_{\mathbf{S}}$ (see [16, App. II]). Independently from $\mathbf{Q}$ and $\mathbf{R}$, $f_1(0) = 1$ and $f_1(1) = 0$ hold. Thus, we also have $f_1(2) = 3$ and $f_1(3) = 2$.

With the above definition, (12) results in the following polyphase decomposition form with factor $\mathbf{S}$:

$$V(\mathbf{z}) = \sum_{k=0}^{3} \mathbf{z}^{-\mathbf{m}_k} \left\{ H_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}}) X_{\mathbf{n}_0}(\mathbf{z}^{\mathbf{S}}) + \mathbf{z}^{-\mathbf{S}\mathbf{g}_1(f_1^{-1}(k))} H_{\mathbf{m}_{f_1^{-1}(k)}}(\mathbf{z}^{\mathbf{S}}) X_{\mathbf{n}_1}(\mathbf{z}^{\mathbf{S}}) \right\} \tag{13}$$

where $f_1^{-1}(k)$ is the inverse mapping of $f_1(k)$.

Since $V(\mathbf{z})$ is downsampled with factor $\mathbf{R}$, only the retained components on $\mathcal{L}(\mathbf{R})$ contribute to output $Y(\mathbf{z})$. Note that the components are identical to those of $W(\mathbf{z})$ on $\mathcal{L}(\mathbf{R})$ in Fig. 5. From this fact, let us derive the relation between $X(\mathbf{z})$ and $Y(\mathbf{z})$ in the polyphase representation.

Recall that there is the integer matrix $\mathbf{Q}'$ such that $\mathbf{S} = \mathbf{R}\mathbf{Q}'$. By using matrix $\mathbf{Q}'$, we define the type-I polyphase components $Y_{\mathbf{k}_n}(\mathbf{z})$ of $Y(\mathbf{z})$ such that

$$Y(\mathbf{z}) = \sum_{n=0}^{1} \mathbf{z}^{-\mathbf{k}_n} Y_{\mathbf{k}_n}(\mathbf{z}^{\mathbf{Q}'}), \quad \mathbf{k}_n \in \mathcal{N}(\mathbf{Q}') \tag{14}$$

where $\mathbf{k}_0 = \mathbf{0}$, and $\mathbf{k}_1$ is the other element in $\mathcal{N}(\mathbf{Q}')$, which has two elements because $J(\mathbf{Q}') = J(\mathbf{S})/J(\mathbf{R}') = 2$.

The relation $W(\mathbf{z}) = Y(\mathbf{z}^{\mathbf{R}})$ holds. Thus, from the fact that $\mathbf{R}\mathbf{k}_0 = \mathbf{m}_0$ and $\mathbf{R}\mathbf{k}_1 = \mathbf{m}_2$, we have

$$W(\mathbf{z}) = \mathbf{z}^{-\mathbf{m}_0} Y_{\mathbf{k}_0}(\mathbf{z}^{\mathbf{S}}) + \mathbf{z}^{-\mathbf{m}_2} Y_{\mathbf{k}_1}(\mathbf{z}^{\mathbf{S}}). \tag{15}$$

Compared with (13) on $\mathcal{L}(\mathbf{R})$, (15) reveals the relation between $X_{\mathbf{n}_\ell}(\mathbf{z})$ and $Y_{\mathbf{k}_n}(\mathbf{z})$ as follows:

$$\begin{pmatrix} Y_{\mathbf{k}_0}(\mathbf{z}) \\ Y_{\mathbf{k}_1}(\mathbf{z}) \end{pmatrix} = \mathbf{H}(\mathbf{z}) \begin{pmatrix} X_{\mathbf{n}_0}(\mathbf{z}) \\ X_{\mathbf{n}_1}(\mathbf{z}) \end{pmatrix} \tag{16}$$

where

$$\mathbf{H}(\mathbf{z}) = \begin{pmatrix} H_{\mathbf{m}_0}(\mathbf{z}), & \mathbf{z}^{-\mathbf{g}_1(1)} H_{\mathbf{m}_1}(\mathbf{z}) \\ H_{\mathbf{m}_2}(\mathbf{z}), & \mathbf{z}^{-\mathbf{g}_1(3)} H_{\mathbf{m}_3}(\mathbf{z}) \end{pmatrix}. \tag{17}$$

Consequently, we have the following theorem:

Theorem 1: Suppose that $\mathbf{Q}$ and $\mathbf{R}$ are integer matrices such that $J(\mathbf{Q}) = J(\mathbf{R}) = 2$ and $\mathcal{L}(\mathbf{Q}) \neq \mathcal{L}(\mathbf{R})$. Then, an SLA system shown in Fig. 2 with factors $\mathbf{Q}$ and $\mathbf{R}$ is invertible if and only if the matrix $\mathbf{H}(\mathbf{z})$ defined by (17), which consists of polyphase components decomposed by an lcrm$(\mathbf{Q}, \mathbf{R})$, is invertible.

Proof: Obviously, from (16), $X_{\mathbf{n}_0}$ and $X_{\mathbf{n}_1}$ are perfectly reconstructed from $Y_{\mathbf{k}_0}$ and $Y_{\mathbf{k}_1}$ if and only if there exits $\mathbf{H}^{-1}(\mathbf{z})$. ∎

C. Inverse Process of SLA

For a forward SLA system satisfying the invertibility, we can construct an inverse system. In the following, we clarify how to relate the inverse filter $F(\mathbf{z})$ with $H(\mathbf{z})$. This problem resembles the relation between analysis and synthesis systems of PR filterbanks. Overall system function from $X(\mathbf{z})$ to $\hat{X}(\mathbf{z})$ is also discussed.

Output $Z(\mathbf{z})$ of filter $F(\mathbf{z})$ in the inverse process shown in Fig. 5 is represented by

$$Z(\mathbf{z}) = F(\mathbf{z})W(\mathbf{z}) = F(\mathbf{z})Y(\mathbf{z}^{\mathbf{R}}). \tag{18}$$

It can be verified that the above relation is rewritten in the polyphase representation as follows:

$$Z(\mathbf{z}) = \left\{ \sum_{k=0}^{3} \mathbf{z}^{\mathbf{m}_k} F_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}}) \right\} \left\{ \sum_{n=0}^{1} \mathbf{z}^{-\mathbf{R}\mathbf{k}_n} Y_{\mathbf{k}_n}(\mathbf{z}^{\mathbf{S}}) \right\} \tag{19}$$

where (14) is used, and $F_{\mathbf{m}_k}$ is the type-II polyphase component of $F(\mathbf{z})$ with factor $\mathbf{S}$ such that

$$F(\mathbf{z}) = \sum_{k=0}^{3} \mathbf{z}^{\mathbf{m}_k} F_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}}),\ \mathbf{m}_k \in \mathcal{N}(\mathbf{S}). \tag{20}$$

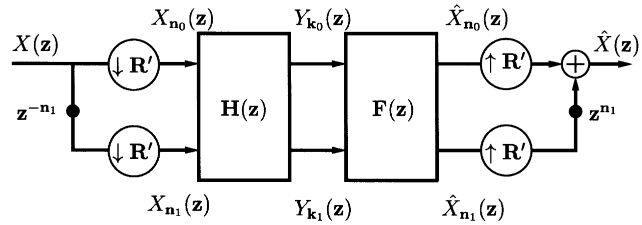


Fig. 6. Polyphase representation of forward and inverse SLA systems.

By using the fact that $\mathbf{R}\mathbf{k}_0 = \mathbf{m}_0$ and $\mathbf{R}\mathbf{k}_1 = \mathbf{m}_2$, (19) can be reduced to the following form in the similar way to the derivation of (13):

$$\begin{aligned} Z(\mathbf{z}) = \sum_{k=0}^{3} \mathbf{z}^{\mathbf{m}_k} \Big\{ & F_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}}) Y_{\mathbf{k}_0}(\mathbf{z}^{\mathbf{S}}) \\ & + \mathbf{z}^{\mathbf{S}\mathbf{g}_2(f_2^{-1}(k))} F_{\mathbf{m}_{f_2^{-1}(k)}}(\mathbf{z}^{\mathbf{S}}) Y_{\mathbf{k}_1}(\mathbf{z}^{\mathbf{S}}) \Big\} \end{aligned} \tag{21}$$

where the following definitions are used.

Definitions: For the integer vectors $\mathbf{m}_k \in \mathcal{N}(\mathbf{S})$, let us define a function $f_2(k)$ and integer vectors $\mathbf{g}_2(k) \in \mathcal{N}$ based on the division theorem such that

$$\mathbf{m}_k - \mathbf{m}_2 = \mathbf{S}\mathbf{g}_2(k) + \mathbf{m}_{f_2(k)}, \quad k = 0, 1, 2, 3$$

where $\mathbf{g}_2(k) \in \mathcal{N}$ and $\mathbf{m}_{f_2(k)} \in \mathcal{N}(\mathbf{S})$. $\mathbf{g}_2(k)$ is the quotient of $\mathbf{m}_k - \mathbf{m}_2$ with respect to $\mathbf{S}$, and $\mathbf{m}_{f_2(k)}$ is the remainder, i.e., $\mathbf{m}_{f_2(k)} = ((\mathbf{m}_k - \mathbf{m}_2))_{\mathbf{S}}$ for $k = 0, 1, 2, 3$. $f_2^{-1}(k)$ is the inverse mapping of $f_2(k)$. Obviously, $f_2(0) = 2$ and $f_2(2) = 0$ hold. Thus, we have $f_2(1) = 3$ and $f_2(3) = 1$.

In the same way as (11), the type-II polyphase representation of $\hat{X}(\mathbf{z})$ with factor $\mathbf{R}'$ is given as follows:

$$\hat{X}(\mathbf{z}) = \sum_{\ell=0}^{1} \mathbf{z}^{\mathbf{n}_\ell} \hat{X}_{\mathbf{n}_\ell}(\mathbf{z}^{\mathbf{R}'}), \quad \mathbf{n}_\ell \in \mathcal{N}(\mathbf{R}'). \tag{22}$$

Note that samples of $\hat{X}(\mathbf{z}^{\mathbf{Q}})$ on $\mathcal{L}(\mathbf{Q})$ is identical to those of $Z(\mathbf{z})$. Thus, compared with (21) on $\mathcal{L}(\mathbf{Q})$, the following equation reveals the relation between $Y_{\mathbf{k}_n}(\mathbf{z})$ and $\hat{X}_{\mathbf{n}_\ell}(\mathbf{z})$:

$$\begin{aligned} \hat{X}(\mathbf{z}^{\mathbf{Q}}) &= \sum_{n=0}^{1} \mathbf{z}^{\mathbf{Q}\mathbf{n}_\ell} \hat{X}_{\mathbf{n}_\ell}(\mathbf{z}^{\mathbf{Q}\mathbf{R}'}) \\ &= \mathbf{z}^{\mathbf{m}_0} \hat{X}_{\mathbf{n}_0}(\mathbf{z}^{\mathbf{S}}) + \mathbf{z}^{\mathbf{m}_1} \hat{X}_{\mathbf{n}_1}(\mathbf{z}^{\mathbf{S}}) \end{aligned} \tag{23}$$

where the fact that $\mathbf{Q}\mathbf{n}_0 = \mathbf{m}_0$ and $\mathbf{Q}\mathbf{n}_1 = \mathbf{m}_1$ is used.

As a result, we have

$$\begin{pmatrix} \hat{X}_{\mathbf{n}_0}(\mathbf{z}) \\ \hat{X}_{\mathbf{n}_1}(\mathbf{z}) \end{pmatrix} = \mathbf{F}(\mathbf{z}) \begin{pmatrix} Y_{\mathbf{k}_0}(\mathbf{z}) \\ Y_{\mathbf{k}_1}(\mathbf{z}) \end{pmatrix} \tag{24}$$

where

$$\mathbf{F}(\mathbf{z}) = \begin{pmatrix} F_{\mathbf{m}_0}(\mathbf{z}), & \mathbf{z}^{\mathbf{g}_2(2)} F_{\mathbf{m}_2}(\mathbf{z}) \\ F_{\mathbf{m}_1}(\mathbf{z}), & \mathbf{z}^{\mathbf{g}_2(3)} F_{\mathbf{m}_3}(\mathbf{z}) \end{pmatrix}. \tag{25}$$

In summary, from (11), (16), (23), and (24), the overall system function of the forward and inverse SLA processes are modeled as two-channel maximally decimated filterbanks, as shown in Fig. 6 [14], [26]. In the same way as the multirate filterbanks, the perfect reconstruction can be achieved if and only the product of $\mathbf{F}(\mathbf{z})\mathbf{H}(\mathbf{z})$ has the following form:

$$\mathbf{G}(\mathbf{z}) = \mathbf{F}(\mathbf{z})\mathbf{H}(\mathbf{z}) = c\mathbf{z}^{-\mathbf{d}_G} \mathbf{I}_r(\mathbf{z}) \tag{26}$$

for $c \neq 0$ and some integer vector $\mathbf{d}_{\mathrm{G}}$, where $\mathbf{I}_r(\mathbf{z})$ is the 2×2 matrix defined for $r = 0, 1$ by

$$\mathbf{I}_r(\mathbf{z}) = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, & r = 0 \\ \begin{pmatrix} 0 & \mathbf{z}^{2\mathbf{R}'^{-1}\mathbf{n}_1} \\ 1 & 0 \end{pmatrix}, & r = 1 \end{cases}$$

where $\mathbf{n}_1$ is the element in $\mathcal{N}(\mathbf{R}')$ other than $\mathbf{n}_0 = \mathbf{0}$. Note that $2\mathbf{R}'^{-1}$ is guaranteed to be integer since $J(\mathbf{R}') = J(\mathbf{S})/J(\mathbf{Q}) = 2$. It can be verified that if (26) holds, the relation between $\hat{X}(\mathbf{z})$ and $X(\mathbf{z})$ results in

$$\hat{X}(\mathbf{z}) = c\mathbf{z}^{-(\mathbf{R}'\mathbf{d}_{\mathrm{G}} - r\mathbf{n}_1)} X(\mathbf{z}) = c\mathbf{z}^{-\mathbf{d}} X(\mathbf{z}). \tag{27}$$

In other words, (7) can be achieved. From (26), we have $\mathbf{F}(\mathbf{z}) = c\mathbf{z}^{-\mathbf{d}_{\mathrm{G}}}\mathbf{I}_r(\mathbf{z})\mathbf{H}^{-1}(\mathbf{z})$. The existence of $\mathbf{F}(\mathbf{z})$ follows Theorem 1.

D. Perfect Reconstruction FIR Systems

SLA systems with FIR filters are of great interest in practice as well as multirate filterbanks. Actually, this problem resembles the design of two-channel M-D maximally-decimated PR FIR filterbanks [14], [20], [22], [23], [27]–[29]. We here show a lemma with regard to the FIR invertibility.

Lemma 2: If and only if the determinant of matrix $\mathbf{H}(\mathbf{z})$ in (17) is a monomial in $\mathbf{z}$, that is, represented as

$$\det \mathbf{H}(\mathbf{z}) = \alpha \mathbf{z}^{-\mathbf{d}_H} \tag{28}$$

for $\alpha \neq 0$ and some integer vector $\mathbf{d}_H$, both of the forward and inverse SLA systems are FIR.

Proof: When $H(\mathbf{z})$ and $F(\mathbf{z})$ are FIR, the determinants of $\mathbf{H}(\mathbf{z})$ and $\mathbf{F}(\mathbf{z})$ are also FIR. From (26), the product $\det \mathbf{F}(\mathbf{z}) \det \mathbf{H}(\mathbf{z})$ should have the form $c_0\mathbf{z}^{-\mathbf{d}_0}$ for $c_0 \neq 0$ and some integer vector $\mathbf{d}_0$. Thus, if and only if $\det \mathbf{H}(\mathbf{z})$ has the form $\alpha\mathbf{z}^{-\mathbf{d}_{\mathrm{E}}}$, $\det \mathbf{F}(\mathbf{z})$ becomes FIR with similar form. See [14, ch. 5] and [20, App. B]. ■

When $\det \mathbf{H}(\mathbf{z}) = \alpha\mathbf{z}^{-\mathbf{d}_H}$, $\mathbf{F}(\mathbf{z})$ can be represented as

$$\mathbf{F}(\mathbf{z}) = \frac{c}{\alpha}\mathbf{z}^{-(\mathbf{d}_{\mathrm{G}} - \mathbf{d}_H)}\mathbf{I}_r(\mathbf{z}) \operatorname{adj} \mathbf{H}(\mathbf{z}) \tag{29}$$

where adj $\mathbf{H}(\mathbf{z})$ denotes the adjugate of $\mathbf{H}(\mathbf{z})$.

IV. Reinterlacer

In this section, we consider the deinterlacing problem introduced in Section II again. When the deinterlacer is FIR invertible, we show that a reinterlacing filter $F(\mathbf{z})$ is simply obtained as a vertically or temporally π-modulated version of given deinterlacing filter $H(\mathbf{z})$ with some shift operations.

A. Temporal Decimation Case

The following lemma holds for the temporal decimation case.

Lemma 3: Consider a deinterlacer in Fig. 2 with $\mathbf{Q}$ in (2) and $\mathbf{R}$ in (3). If the determinant of $H(\mathbf{z})$ satisfies the monomial condition described in Lemma 2, its dual reinterlacing filter $F(\mathbf{z})$ has the following form:

$$F(\mathbf{z}) = K\mathbf{z}^{-\begin{pmatrix} N_{\mathrm{T}} \\ N_{\mathrm{V}} \\ N_H \end{pmatrix}} H\left(W_2^{(0\ 1\ 0)}\mathbf{z}\right) \tag{30}$$

for $K \neq 0$, integer N_H, odd (or even) N_{T} and even (or odd) N_{V}, where $W_N = e^{-j(2\pi)/(N)}$, and then

$$W_2^{(0\ 1\ 0)}\mathbf{z} = W_2^{(0\ 1\ 0)}\begin{pmatrix} z_{\mathrm{T}} \\ z_{\mathrm{V}} \\ z_H \end{pmatrix} = \begin{pmatrix} z_{\mathrm{T}} \\ -z_{\mathrm{V}} \\ z_H \end{pmatrix}.$$

The overall delay $\mathbf{d}$ results in

$$\begin{aligned} \mathbf{d} &= \mathbf{R}'\mathbf{d}_H + \mathbf{Q}^{-1}\begin{pmatrix} N_{\mathrm{T}} + 1 \\ N_{\mathrm{V}} + 2 \\ N_H \end{pmatrix} \\ &= \mathbf{R}'\mathbf{d}_H + \begin{pmatrix} \frac{N_{\mathrm{T}} - N_{\mathrm{V}} - 1}{2} \\ \frac{N_{\mathrm{T}} + N_{\mathrm{V}} + 3}{2} \\ N_H \end{pmatrix}. \end{aligned} \tag{31}$$

Proof: From the monomial condition, (29) holds. One choice of lcrm$(\mathbf{Q}, \mathbf{R})$ is

$$\begin{aligned} \mathbf{S} &= \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{Q}} \underbrace{\begin{pmatrix} 1 & -1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{R}'} \\ &= \underbrace{\begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{R}} \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{Q}'}. \end{aligned}$$

Thus, from (17) and (25), the following relation is obtained:

$$\begin{aligned} &\begin{pmatrix} F_{\mathbf{m}_0}(\mathbf{z}) & F_{\mathbf{m}_2}(\mathbf{z}) \\ F_{\mathbf{m}_1}(\mathbf{z}) & \mathbf{z}^{-\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}} F_{\mathbf{m}_3}(\mathbf{z}) \end{pmatrix} \\ &= \frac{c}{\alpha}\mathbf{z}^{-(\mathbf{d}_{\mathrm{G}} - \mathbf{d}_H)}\mathbf{I}_r(\mathbf{z}) \begin{pmatrix} \mathbf{z}^{\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}} H_{\mathbf{m}_3}(\mathbf{z}) & -H_{\mathbf{m}_1}(\mathbf{z}) \\ -H_{\mathbf{m}_2}(\mathbf{z}) & H_{\mathbf{m}_0}(\mathbf{z}) \end{pmatrix} \end{aligned}$$

where

$$\mathbf{m}_0 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{m}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{m}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

$$\mathbf{m}_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{g}_1(1) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \mathbf{g}_1(3) = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

$$\mathbf{g}_2(2) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad \text{and} \quad \mathbf{g}_2(3) = \begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}.$$

Equations (10) and (20) and the above relation yield

$$\begin{aligned}F(\mathbf{z}) &= \frac{c}{\alpha}\mathbf{z}^{-\mathbf{S}(\mathbf{d}_{\mathrm{G}}-\mathbf{d}_H)+r\mathbf{Q}\mathbf{n}_1+\begin{pmatrix}1\\2\\0\end{pmatrix}} \\ &\quad\times\left\{\sum_{k=0,3}\mathbf{z}^{-\mathbf{m}_k}H_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}})-\sum_{k=1,2}\mathbf{z}^{-\mathbf{m}_k}H_{\mathbf{m}_k}(\mathbf{z}^{\mathbf{S}})\right\} \\ &= \frac{c}{\alpha}\mathbf{z}^{-\mathbf{Q}\{\mathbf{R}'(\mathbf{d}_{\mathrm{G}}-\mathbf{d}_H)-r\mathbf{n}_1\}+\begin{pmatrix}1\\2\\0\end{pmatrix}}H\left(W_2^{(0\ 1\ 0)}\mathbf{z}\right)\end{aligned}$$

where $\mathbf{n}_1$ is the element in $\mathcal{N}(\mathbf{R}')$ other than $\mathbf{0}$, and $\mathbf{R}'$ is the matrix such that $\mathbf{S}=\mathbf{Q}\mathbf{R}'$. The above equation is reduced to (30) when $K=c/\alpha$ and

$$\begin{pmatrix}N_{\mathrm{T}}\\N_{\mathrm{V}}\\N_H\end{pmatrix}=\mathbf{Q}\{\mathbf{R}'(\mathbf{d}_{\mathrm{G}}-\mathbf{d}_H)-r\mathbf{n}_1\}-\begin{pmatrix}1\\2\\0\end{pmatrix}.$$

Since $\mathbf{d}=\mathbf{R}'\mathbf{d}_{\mathrm{G}}-r\mathbf{n}_1$ from (27), we have (31). Note that $\mathbf{d}$ must be integer, and this holds if and only if either of N_{T} or N_{V} is odd and the other is even and that (30) is independent from the intermediate choice of matrix $\mathbf{S}$. ∎

Equation (30) implies that $F(\mathbf{z})$ is the vertically π-modulated version of $H(\mathbf{z})$ with some odd (or even) number of temporal delay N_{T}, even (or odd) number of vertical shift N_{V}, and some horizontal shift N_H.

Example: Let us consider the case of field interleaving. The deinterlacing filter is given by $H(\mathbf{z})=1+z_{\mathrm{T}}^{-1}$. Thus, the polyphase components are obtained as $H_{\mathbf{m}_0}(\mathbf{z})=1, H_{\mathbf{m}_1}(\mathbf{z})=0,\ H_{\mathbf{m}_2}(\mathbf{z})=0$, and $H_{\mathbf{m}_3}(\mathbf{z})=1$. This filter satisfies FIR invertibility in Lemma 2 with $\alpha=1$ and

$$\mathbf{d}_H=\begin{pmatrix}0\\-1\\0\end{pmatrix}.$$

From (30), one possible choice of reinterlacing filters is $F(\mathbf{z})=z_{\mathrm{T}}^{-1}(1+z_{\mathrm{T}}^{-1})$, where $N_{\mathrm{T}}=1, N_{\mathrm{V}}=0$, and $N_H=0$. The overall delay $\mathbf{Q}\mathbf{d}$ in the upsampled progressive domain becomes pure temporal delay z_{T}^{-2}, where

$$\mathbf{d}=\begin{pmatrix}1\\1\\0\end{pmatrix}.$$

(End of Example)

B. Vertical Decimation Case

Since the vertical decimation case of deinterlacing is only differ in a swap of vertical and temporal variables, the following lemma holds for the vertical decimation case.

Lemma 4: Consider a deinterlacer in Fig. 2 with $\mathbf{Q}$ in (2) and $\mathbf{R}$ in (5). If $H(\mathbf{z})$ satisfies the monomial condition described in Lemma 2, its dual reinterlacing filter $F(\mathbf{z})$ has the following form:

$$F(\mathbf{z})=K\mathbf{z}^{-\begin{pmatrix}N_{\mathrm{T}}\\N_{\mathrm{V}}\\N_H\end{pmatrix}}H\left(W_2^{(1\ 0\ 0)}\mathbf{z}\right) \tag{32}$$

for $K\neq 0$, integer N_H, even (or odd) N_{T} and odd (or even) N_{V}, where

$$W_2^{(1\ 0\ 0)}\mathbf{z}=\begin{pmatrix}-z_{\mathrm{T}}\\z_{\mathrm{V}}\\z_H\end{pmatrix}.$$

The overall delay $\mathbf{d}$ results in

$$\begin{aligned}\mathbf{d}&=\mathbf{R}'\mathbf{d}_H+\mathbf{Q}^{-1}\begin{pmatrix}N_{\mathrm{T}}+2\\N_{\mathrm{V}}+1\\N_H\end{pmatrix}\\&=\mathbf{R}'\mathbf{d}_H+\begin{pmatrix}\frac{N_{\mathrm{T}}-N_{\mathrm{V}}+1}{2}\\\frac{N_{\mathrm{T}}+N_{\mathrm{V}}+3}{2}\\N_H\end{pmatrix}.\end{aligned} \tag{33}$$

Proof: See the proof for Lemma 3, and swap the temporal and vertical variables. ∎

Equation (32) implies that $F(\mathbf{z})$ is the temporally π-modulated version of $H(\mathbf{z})$ with some even (or odd) number of temporal delay N_{T}, odd (or even) number of vertical shift N_{V}, and some horizontal shift N_H.

Example: For the field separation case, the dual filters are given by $H(\mathbf{z})=1+z_{\mathrm{V}}^{1}$ and $F(\mathbf{z})=1+z_{\mathrm{V}}^{-1}$, where $\alpha=1$

$$\mathbf{d}_H=\begin{pmatrix}-1\\-1\\0\end{pmatrix}$$

$N_{\mathrm{T}}=0,\ N_{\mathrm{V}}=1$, and $N_H=0$. The overall delay $\mathbf{d}$ is zero. (End of Example)

V. Design Examples and Simulation

The theorems described in the previous sections show that an invertible deinterlacer can be designed by controlling a 2×2 polyphase matrix with a monomial determinant. The design problem is, thus, similar to that for filterbanks, although only one filter is optimized instead of multiple filters.

A. Generic Design Procedure

From Lemmas 3 and 4, the following design procedure is derived.

Step 1) Determine the specification for a deinterlacing filter $H(\mathbf{z})$, such as support of region and frequency response.

Step 2) Optimize deinterlacing filter $H(\mathbf{z})$ subject to the constraint that the $\det\mathbf{H}(\mathbf{z})$ is a monomial in $\mathbf{z}$.

Step 3) Obtain a reinterlacing filter $F(\mathbf{z})$ satisfying (30) for the temporal decimation case or (32) for the vertical decimation case.

There are many techniques for constructing a multivariable polynomial matrix whose determinant is a monomial in $\mathbf{z}$. In the field of multirate-filterbank theory, the problem has been widely studied of constructing PR multidimensional FIR systems [14],

[20], [22], [23], [27]–[29], for example, a design by a cascade structure, one based on a state-space description, and one based on the Smith form.

Unlike filterbanks, multiple filters do not need to be optimized simultaneously. Only one deinterlacing filter $H(\mathbf{z})$ is required to be optimized, and a reinterlacing filter $F(\mathbf{z})$ is determined by $H(\mathbf{z})$ according to our theory. However, there are still a lot of parameters to be optimized if no constraint is imposed. The design problem would be complex and likely to yield undesirable results without any constraint. Thus, we have to take some system specifications into account from the practical point of view.

B. Design Constraints

To reduce the number of design parameters, we introduce here some constraints on deinterlacing filters.

1) Normalization: To normalize the amplitude of $Y(\mathbf{z})$, the dc amplitude $|H(\mathbf{1})|$ has to be $J(\mathbf{Q}) = 2$. This constraint is represented in terms of the polyphase components by

$$H_{\mathbf{m}_0}(\mathbf{1}) + H_{\mathbf{m}_1}(\mathbf{1}) + H_{\mathbf{m}_2}(\mathbf{1}) + H_{\mathbf{m}_3}(\mathbf{1}) = 2. \tag{34}$$

Actually, this holds for a general case.

2) Regularity: Forward SLA systems or invertible deinterlacers include interpolator. An interpolator is known to result in a linear periodical shift-variant (LPSV) system [30]. Thus, unless the interpolation filter satisfies a particular smoothness, the dc gain varies periodically. In this case, we have some structural artifacts in output sequence $Y(\mathbf{z})$, even for a smooth input. SLA systems, as shown in Fig. 2, result in an LPSV systems with period $\mathbf{Q}'$ since (16) holds with (14). For the dc gain of the deinterlacer to be shift invariant, the sum of the dc gains of the top two components in (17) have to be identical to that of the bottom two. Otherwise, the dc components $Y_{\mathbf{k}_0}(\mathbf{1}) = H_{\mathbf{m}_0}(\mathbf{1})X_{\mathbf{m}_0}(\mathbf{1}) + H_{\mathbf{m}_1}(\mathbf{1})X_{\mathbf{m}_1}(\mathbf{1})$ and $Y_{\mathbf{k}_1}(\mathbf{1}) = H_{\mathbf{m}_2}(\mathbf{1})X_{\mathbf{m}_0}(\mathbf{1}) + H_{\mathbf{m}_3}(\mathbf{1})X_{\mathbf{m}_1}(\mathbf{1})$ become different from each other even if $X_{\mathbf{m}_0}(\mathbf{1})$ and $X_{\mathbf{m}_1}(\mathbf{1})$ are identical to each other. Thus, the following constraint is desirable:

$$H_{\mathbf{m}_0}(\mathbf{1}) + H_{\mathbf{m}_1}(\mathbf{1}) = H_{\mathbf{m}_2}(\mathbf{1}) + H_{\mathbf{m}_3}(\mathbf{1}). \tag{35}$$

We refer to this property as *regularity* according to the wavelet theory [26]. This also holds for a general case. For deinterlacers, this constraint implies that $H(W_2^{(1\ 1\ 0)}) = H(e^{-j(\pi\ \pi\ 0)}) = 0$.

TABLE I
OPTIMAL PARAMETERS OF $(5+3)$-TAP DEINTERLACING FILTERS DESIGNED FOR MINIMIZING THE OBJECTIVE FUNCTION ϕ IN (41) WITH THE SPECIFICATIONS SHOWN IN FIG. 7. SCALING FACTOR K FOR REINTERLACING FILTERS IS ALSO SHOWN

Type	$h(0,0)$	$h(1,0)$	K
Temporal filter	0.98287	0.98292	1.04
VT filter	0.95244	0.28059	3.85
Vertical filter	0.99329	−0.05272	−19.24

3) Vertical Symmetry: When filters are vertically symmetric, the symmetric extension method is applicable for the vertical direction [26], [31]. This constraint is given by $H(\mathbf{z}) = \mathbf{z}^{-(0\ 2C_V\ 0)^T} H(\mathbf{z}^{\mathbf{\Gamma}_V})$, where $\mathbf{\Gamma}_V = \mathrm{diag}(1,-1,1)$, and C_V is an integer multiple of $1/2$, which denotes the center of symmetry. For even $2C_V$, we have (36a), shown at the bottom of the page, and for odd $2C_V$

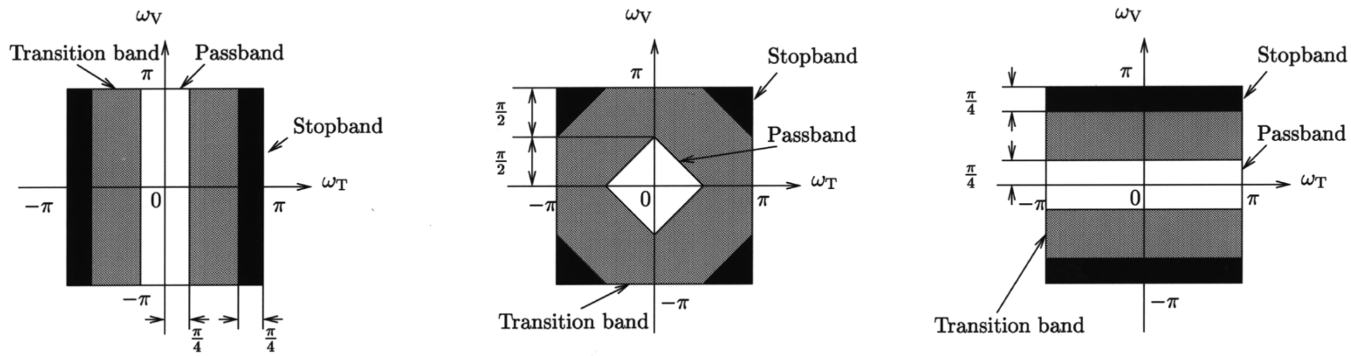
$$H_{\mathbf{m}}(\mathbf{z}) = \mathbf{z}^{-\begin{pmatrix}0\\ C_V-\frac{1}{2}\\ 0\end{pmatrix}} H_{\mathbf{m}+\begin{pmatrix}0\\1\\0\end{pmatrix}}\left(\mathbf{z}^{\mathbf{\Gamma}_V}\right)$$
$$\mathbf{m} \in \left\{ \begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix} \right\}. \tag{36b}$$

4) Horizontal Symmetry: Similarly, the constraint on the horizontal symmetry is given by $H(\mathbf{z}) = \mathbf{z}^{-(0\ 0\ 2C_H)^T} H(\mathbf{z}^{\mathbf{\Gamma}_H})$, where $\mathbf{\Gamma}_H = \mathrm{diag}(1,1,-1)$, and C_H is an integer multiple of $1/2$. Since the polyphase representation is the analog of that in the discussion on vertical symmetry, we omit it here.

C. Design Examples

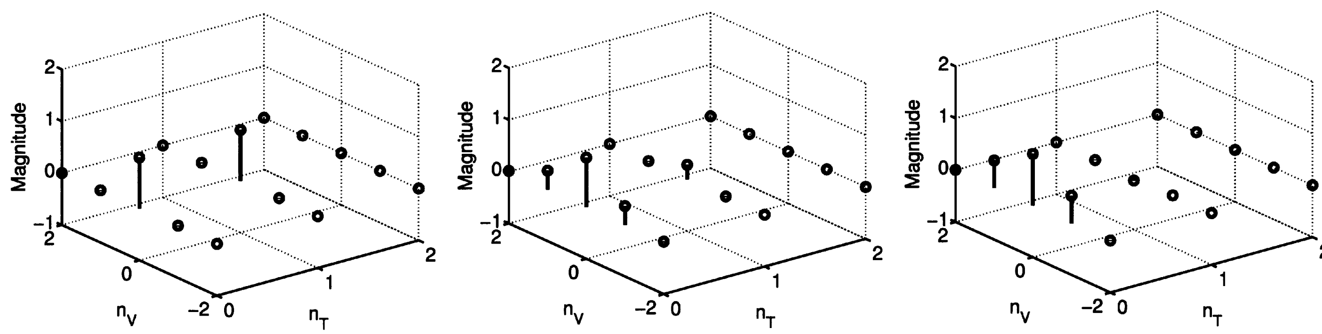
As a practical design procedure, we propose a technique for obtaining invertible deinterlacing filters for the temporal decimation case. Examples shown here are 2-D filters in the temporal and vertical domain; that is, the order of the horizontal direction is zero. All of the previous constraints, where the horizontal symmetry is trivial since it is automatically satisfied, are applied. First, a design procedure for an eight-tap filter, which is referred to as a $(5+3)$-*tap deinterlacing filter* from its region

$$H_{\mathbf{m}}(\mathbf{z}) = \begin{cases} \mathbf{z}^{-\begin{pmatrix}0\\ C_V\\ 0\end{pmatrix}} H_{\mathbf{m}}\left(\mathbf{z}^{\mathbf{\Gamma}_V}\right), & \mathbf{m} \in \left\{ \begin{pmatrix}0\\0\\0\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix} \right\} \\ \mathbf{z}^{-\begin{pmatrix}0\\ C_V-1\\ 0\end{pmatrix}} H_{\mathbf{m}}(\mathbf{z}^{\mathbf{\Gamma}_V}), & \mathbf{m} \in \left\{ \begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}1\\1\\0\end{pmatrix} \right\} \end{cases} \tag{36a}$$



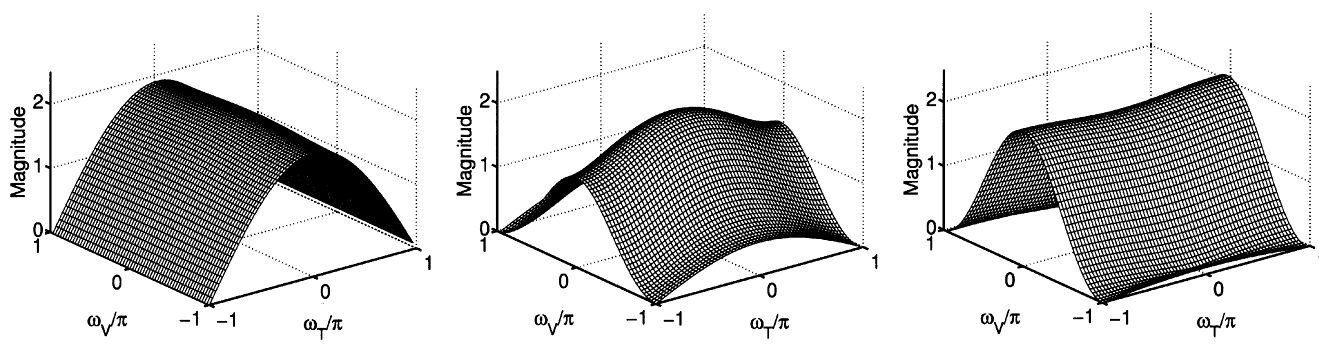
(a) (b) (c)

Fig. 7. Specifications regarding the support of frequency responses. (a) Temporal lowpass support. (b) VT lowpass support. (c) Vertical lowpass support.



(a) (b) (c)

Fig. 8. Impulse responses $h(n_T, n_V)$ of the three example $(5+3)$-tap deinterlacing filter designs. (a) Temporal filter. (b) VT filter. (c) Vertical filter.



(a) (b) (c)

Fig. 9. Amplitude responses $|H(e^{j(\omega_T\ \omega_V\ 0)})|$ of the three example $(5+3)$-tap deinterlacing filter designs. (a) Temporal filter. (b) VT filter. (c) Vertical filter.

of support, is given. Then, one for a four-tap filter, which is referred to as a $(3+1)$-*tap deinterlacing filter*, is shown as a more compact support and a sample invariant case.

1) $(5+3)$-Tap Deinterlacing Filter: The first example is assumed to have the following form:

$$H(\mathbf{z}) = h(0,0) + h(1,0)z_T^{-1} + h(0,1)\left(z_V^1 + z_V^{-1}\right) + h(1,1)z_T^{-1}\left(z_V^1 + z_V^{-1}\right) + h(0,2)\left(z_V^2 + z_V^{-2}\right) \quad (37)$$

where $h(n_T, n_V)$ denotes the (n_T, n_V)th filter coefficient. The vertical symmetry is already forced. From (28), (34), and (35), it is verified that the design parameters can be reduced to two coefficients ($h(0,0)$ and $h(1,0)$), and the other coefficients can be obtained from

$$h(0,1) = \frac{1 - h(1,0)}{2} \quad (38a)$$

$$h(0,2) = \frac{\{1 - h(0,0)\}\{1 - h(1,0)\}}{2\{1 + h(1,0)\}} \quad (38b)$$

$$h(1,1) = \frac{h(1,0)\{1 - h(0,0)\}}{1 + h(1,0)} \quad (38c)$$

where $h(1,0) \neq -1$, which is not a severe condition in a lowpass filter design (see the Appendix), is assumed. With this choice of coefficients, the scale α, i.e., $\det \mathbf{H}(\mathbf{1})$, is

$$\alpha = \frac{h(1,0)\{2h(0,0) + h(1,0) - 1\}}{1 + h(1,0)}. \quad (39)$$

To make the overall scaling factor c unity, scaling factor K for the reinterlacing filter should be $K = 1/\alpha$, that is

$$K = \frac{1 + h(1,0)}{h(1,0)\{2h(0,0) + h(1,0) - 1\}}. \quad (40)$$

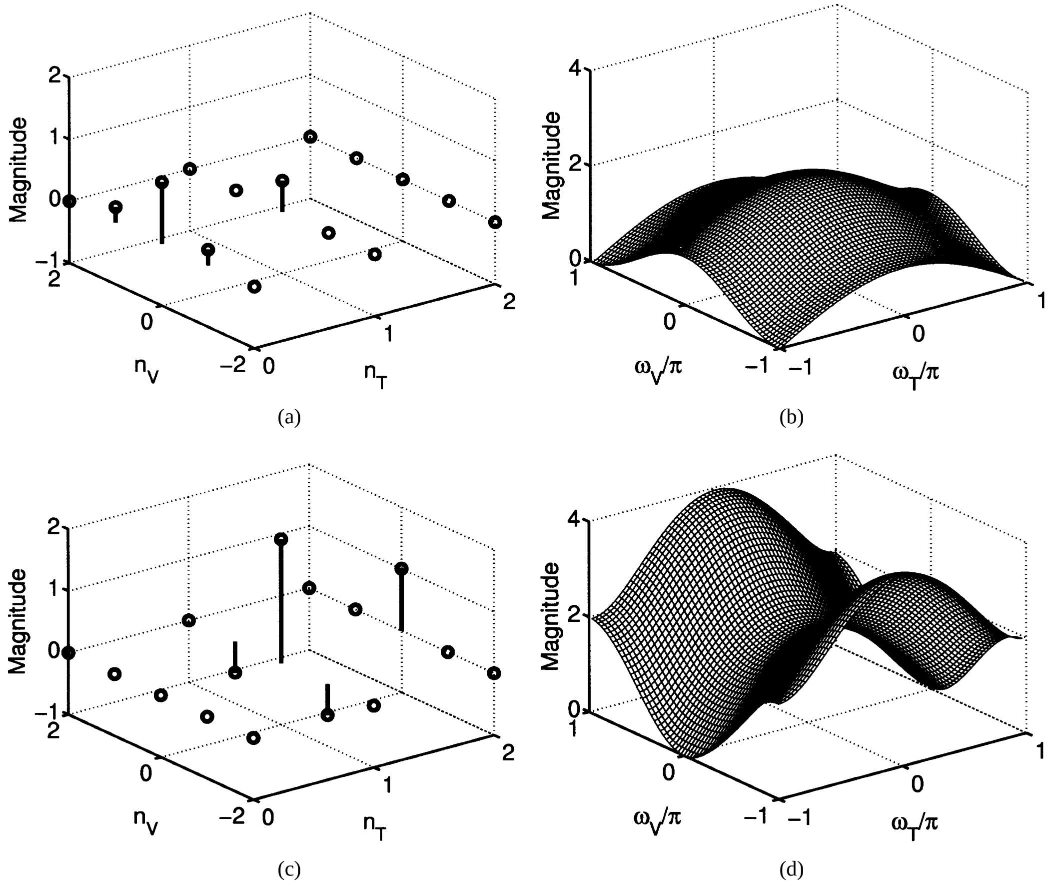


Fig. 10. Design example by a $(3+1)$-tap filter and its dual. (a) $h(n_{\mathrm{T}}, n_{\mathrm{V}})$. (b) $|H(e^{j(\omega_{\mathrm{T}}\ \omega_{\mathrm{V}}\ 0)})|$. (c) $f(n_{\mathrm{T}}, n_{\mathrm{V}})$. (d) $|F(e^{j(\omega_{\mathrm{T}}\ \omega_{\mathrm{V}}\ 0)})|$.

Table I lists three pairs of optimal parameters $h(0,0)$ and $h(1,0)$ with scaling factor K. These filters are designed for minimizing the objective function defined by

$$\phi = \iint_{\mathrm{Passband}} \left|2 - H\left(e^{j(\omega_{\mathrm{T}}\ \omega_{\mathrm{V}}\ 0)}\right)\right|^2 d\omega_{\mathrm{T}}\, d\omega_{\mathrm{V}} + \iint_{\mathrm{Stopband}} \left|H\left(e^{j(\omega_{\mathrm{T}}\ \omega_{\mathrm{V}}\ 0)}\right)\right|^2 d\omega_{\mathrm{T}}\, d\omega_{\mathrm{V}} \quad (41)$$

with the temporal, vertical, and vertical-temporal (VT) lowpass frequency responses specified in Fig. 7, respectively. The first and second terms on the right-hand side represent the passband error energy and the stopband energy, respectively. The design parameters $h(0,0)$ and $h(1,0)$ were controlled during optimization by using a routine of the unconstrained nonlinear optimization process "**fminunc**" (provided by the MATLAB Optimization Toolbox) with the initial values $h^{(0)}(0,0) = 1$ and $h^{(0)}(1,0) = 1$ [32]. Their impulse and amplitude responses are given in Figs. 8 and 9, respectively. From Fig. 9, it is verified that all of the amplitude responses are two at the dc point ($\omega_{\mathrm{T}} = \omega_{\mathrm{V}} = 0$) and vanish at the aliasing frequency ($\omega_{\mathrm{T}} = \omega_{\mathrm{V}} = \pi$).

One possible choice of $F(\mathbf{z})$ is given by

$$\begin{aligned} F(\mathbf{z}) = Kz_{\mathrm{T}}^{-1} \{ & h(0,0) + h(1,0) z_{\mathrm{T}}^{-1} \\ & - h(0,1)\left(z_{\mathrm{V}}^{1} + z_{\mathrm{V}}^{-1}\right) - h(1,1) z_{\mathrm{T}}^{-1}\left(z_{\mathrm{V}}^{1} + z_{\mathrm{V}}^{-1}\right) \\ & + h(0,2)\left(z_{\mathrm{V}}^{2} + z_{\mathrm{V}}^{-2}\right)\} \end{aligned} \quad (42)$$

where $N_{\mathrm{T}} = 1, N_{\mathrm{V}} = 0$, and $N_H = 0$. The overall delay $\mathbf{Qd}$ in the upsampled progressive domain becomes the pure temporal delay z_{T}^{-2}, where

$$\mathbf{d} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

The specifications shown in Fig. 7 are determined in order to remove imaging and avoid aliasing caused by resamplers. The shape, however, highly depends on the characteristics of pictures. For still pictures, temporal filters are desired to keep the fine quality. For motion pictures, vertical filters are preferable to suppress the comb-shaped artifacts, which consist of the highest vertical frequency components. From another point of view, the vertical filter has a large scaling factor K. This large gain might be a problem when combined with some codec. In this case, VT filters may be candidates. The point here is that we are allowed to control the characteristics under the PR condition.

Swapping the temporal and vertical variables, the form in (37) is also valid for the vertical decimation case in terms of PR. Temporal symmetry, then, takes the place of vertical symmetry. From the practical point of view, however, it is of interest if the vertical symmetry holds or not rather than the temporal symmetry because it determines if the symmetric extension method can be used for spatial limitation of size increasing due to vertical filtering or not. In most cases, temporal symmetry may be irrelevant.

2) $(3+1)$-Tap Deinterlacing Filter: It is desirable to remain original sample values in output $Y(\mathbf{z})$ on the same sample points, that is, $Y_{\mathbf{k}_0}(\mathbf{z}) = X_{\mathbf{n}_0}(\mathbf{z})$. From (16), this holds when $H_{\mathbf{m}_0}(\mathbf{z}) = 1$, and $H_{\mathbf{m}_1}(\mathbf{z}) = 0$. This condition is not necessarily satisfied by $(5+3)$-tap deinterlacing filters, however. In the followings, we develop a design procedure of $(3+1)$-tap deinterlacing filters as one satisfying the sample invariance and as a more compact support case.

We derive $(3+1)$-tap deinterlacing filters from the design procedure for $(5+3)$-tap ones by setting $h(0,0) = 1$. The coefficients $h(0,2)$ and $h(1,1)$ are restricted to zero, and then, the following formula is yielded:

$$h(0,1) = \frac{1 - h(1,0)}{2} \tag{43}$$

where the design parameter is reduced to only $h(1,0)$. For c to be unity, K has to be

$$K = \frac{1}{h(1,0)}. \tag{44}$$

Although the design by optimization process is still available, we here give an example of a special filter pair, which is of interest from the practical point of view.

It is verified that one possible choice of $h(1,0)$ is $h(1,0) = 1/2$, which yields the following pair of filters:

$$H(\mathbf{z}) = 1 + \frac{1}{2}z_{\mathrm{T}}^{-1} + \frac{1}{4}\left(z_{\mathrm{V}}^{1} + z_{\mathrm{V}}^{-1}\right) \tag{45a}$$

$$F(\mathbf{z}) = 2z_{\mathrm{T}}^{-1}\left\{1 + \frac{1}{2}z_{\mathrm{T}}^{-1} - \frac{1}{4}\left(z_{\mathrm{V}}^{1} + z_{\mathrm{V}}^{-1}\right)\right\} \tag{45b}$$

where the overall delay in the upsampled progressive domain results in $\mathbf{Qd} = z_{\mathrm{T}}^{-2}$, and

$$\mathbf{d} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

Note that all of the suggested constraints are satisfied and all of the coefficients are powers of two. Additionally, they are a good approximation of the VT lowpass $(5+3)$-tap deinterlacing filter shown in Fig. 8(b) with $H_{\mathbf{m}_0}(\mathbf{z}) = 1$ and $H_{\mathbf{m}_1}(\mathbf{z}) = 0$. Fig. 10 shows the impulse and amplitude responses.

Deinterlacing with the filter in (45a) and the inverse conversion with the filter in (45b) are efficiently implemented, as shown in Fig. 11. The symmetric extension method is assumed to be applied for the vertical direction.

D. Simulation

To verify the significance of our proposed deinterlacing, two simulation results are compared with pictures obtained by conventional field-interleaving.

Fig. 12 shows simulation results by using motion picture "Football." The results by the proposed method is computed via the three $(5+3)$-tap filters shown in Fig. 8 and the $(3+1)$-tap filters given in Fig. 10(a), respectively. The result by the conventional method is obtained via the two-tap deinterlacing filter $H(\mathbf{z}) = 1 + z_{\mathrm{T}}^{-1}$.

The original picture processed in this simulation is a noticeable region of a frame in "Football." The results show that the proposed deinterlacing suppresses comb-shaped artifacts, whereas the conventional field-interleaving causes the artifacts at the edges of moving footballers. Note that the original fields

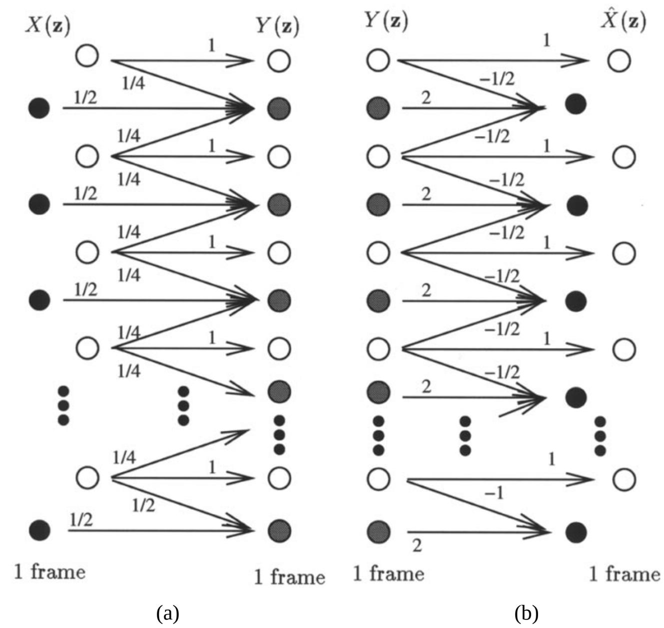


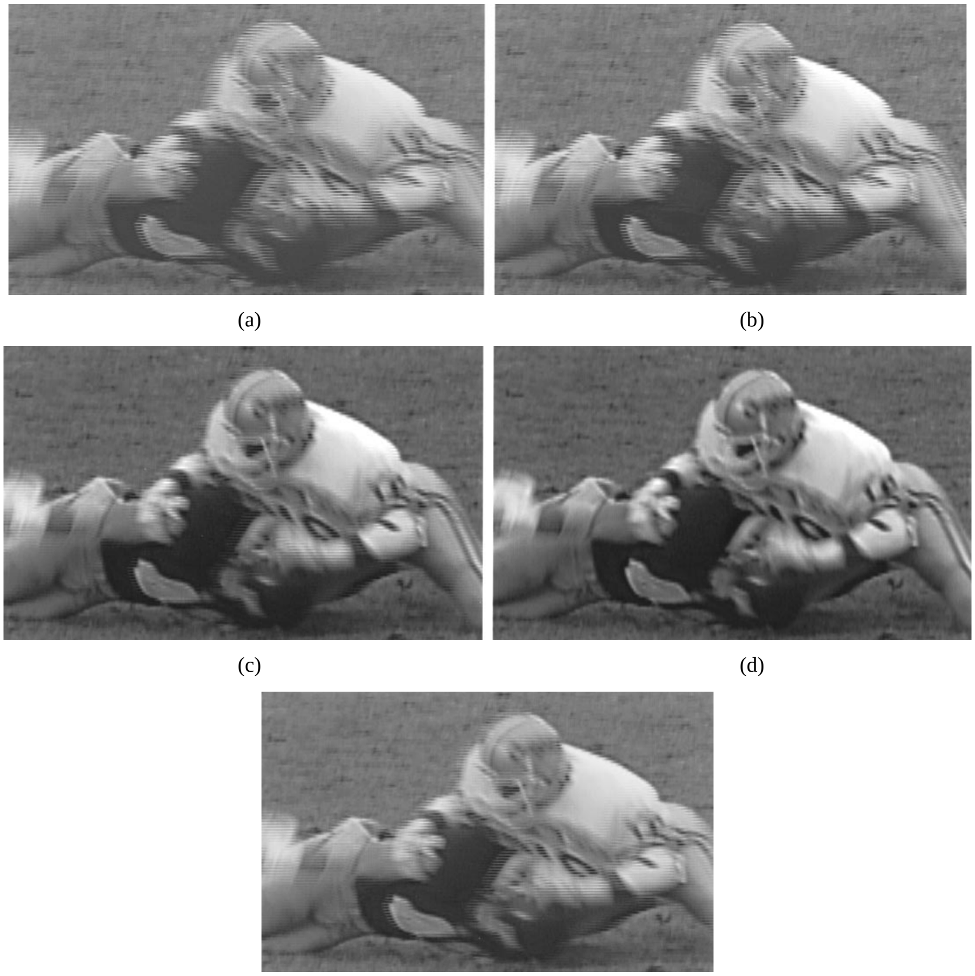
Fig. 11. Efficient implementation of deinterlacing and reinterlacing for two succeeding fields in order to generate a frame picture with the filters given by (45a) and (45b). (a) Deinterlacing. (b) Reinterlacing.

constructing the frame picture can be perfectly reconstructed from the deinterlaced pictures by using the corresponding reinterlacer. The $(5+3)$-tap vertical filter suppresses the comb-shaped artifacts well, although the scale factor for the reinterlacing filter is large, as shown in Table I.

To give objective evaluation, we also experiment with deinterlacing by a moving zone-plate (MZP) sequence [7]. Fig. 13(a) is an original 8-b/pixel frame picture of size 256 × 256, which is referred to as a circular zone-plate picture (CZP). Fig. 13(b) is a frame picture generated via field-interleaving, where the processed MZP sequence is generated by moving the original CZP frame to right and downsampling into an interlaced-scanning sequence. The right motion is set to eight pixels per field. MSE and PSNR of the picture given in Fig. 13(b) result in 8.13×10^3 and 9.03 dB. Fig. 14 shows frame pictures generated via deinterlacing with the four filters shown in Figs. 8 and 10(a). MSE and PSNR are also shown. The significance of our proposed technique is verified objectively. The vertical filter is better than the others in terms of MSE and PSNR, whereas the VT filters give moderate results with moderate K. Especially, the $(3+1)$-tap VT filter is shown to be a good approximation of the $(5+3)$-tap VT filter.

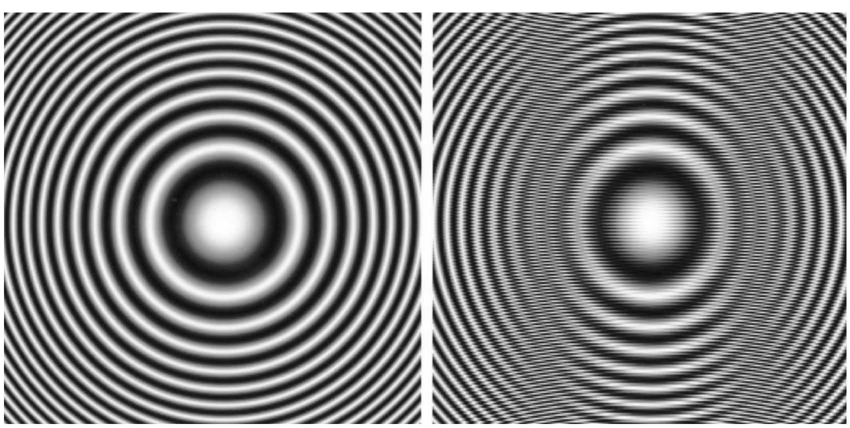
VI. Conclusion

A new class of deinterlacing, which preserves the sampling density (an indispensable property for coding applications), was developed. It is also invertible; thus, the original interlaced pictures can be perfectly reconstructed on demand. The proposed deinterlacing is applicable as preprocessing to an intra-frame/field-based motion-picture coding with scalable functionalities such as MJP2, as well as an inter-frame-based one, without the support of interlaced scanning video such as MPEG-1. The proposed deinterlacing is a generalization of conventional field interleaving and field separation and



(a) (b) (c) (d) (e)

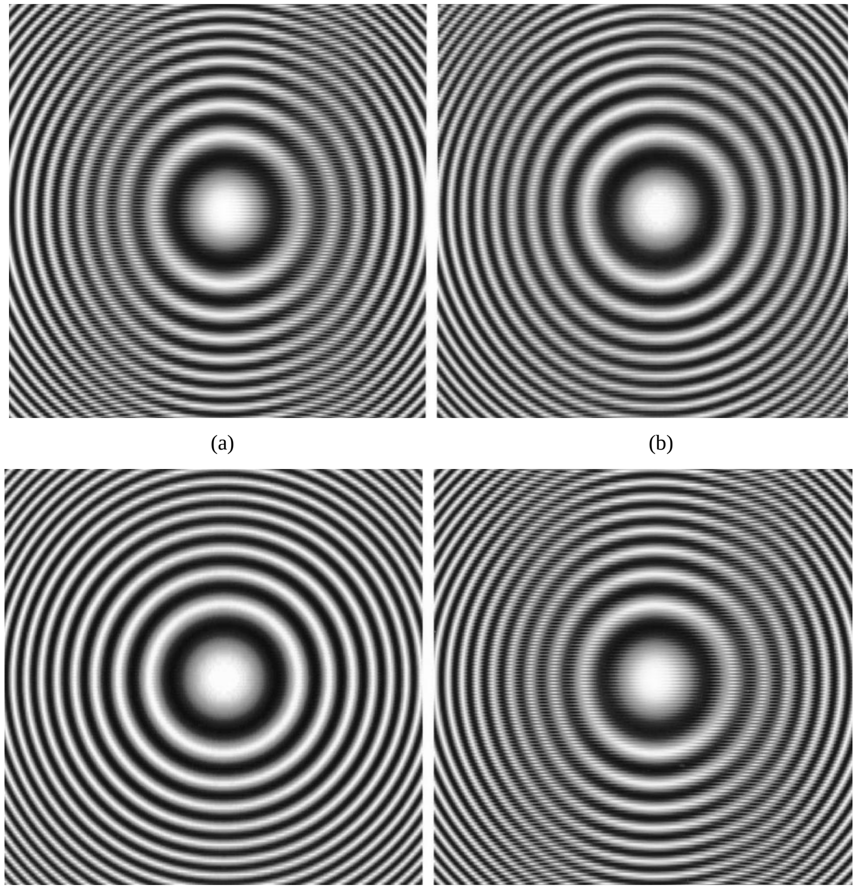
Fig. 12. Deinterlaced array $Y(\mathbf{z})$. (a) Deinterlacing with the field-interleaving. (b) Deinterlacing with the $(5+3)$-tap temporal filter shown in Fig. 8(a). (c) Deinterlacing with the $(5+3)$-tap VT filter shown in Fig. 8(b). (d) Deinterlacing with the $(5+3)$-tap vertical filter shown in Fig. 8(c). (e) Deinterlacing with the $(3+1)$-tap VT filter shown in Fig. 10(a).



(a) (b)

Fig. 13. Original and field-interleaved frame pictures of a moving zone-plate (MZP) sequence of size 256×256 (8 bits/pixel). (a) Original frame picture. (b) Field-interleaved frame picture, where MSE $= 8.13 \times 10^3$, and PSNR $= 9.03$ dB.

can be designed by finding a 2×2 multivariable polynomial matrix with a monomial determinant. By taking constraints on normalization, regularity, and vertical symmetry into account, a practical design procedure was suggested. Consequently, with a nonlinear unconstraint optimization process, three $(5+3)$-tap deinterlacing filters were designed. Temporal, vertical, and VT lowpass deinterlacing filters were produced. Additionally, a special $(3+1)$-tap deinterlacing filter was developed. They were applied to deinterlacing simulation in order to verify the significance of our proposed technique. It was found that the developed technique produces fewer comb-shaped artifacts than conventional field-interleaving technique.



(a) (b)

(c) (d)

Fig. 14. Experimental results of deinterlacing with a moving zone-plate (MZP) sequence. (a) $(5+3)$-tap temporal filter shown in Fig. 8(a), where MSE $= 7.87 \times 10^3$, and PSNR $= 9.17$ dB. (b) $(5+3)$-tap VT filter shown in Fig. 8(b), where MSE $= 8.58 \times 10^2$, and PSNR $= 18.80$ dB. (c) $(5+3)$-tap vertical filter shown in Fig. 8(c), where MSE $= 7.47 \times 10^1$, and PSNR $= 29.39$ dB. (d) $(3+1)$-tap VT filter shown in Fig. 10(a), where MSE $= 2.26 \times 10^3$, and PSNR $= 14.59$ dB.

Since this work is the first step concerning this novel class of deinterlacing, we assumed that filter coefficients were fixed. In practice, however, some motion adaptation technique is desired. This problem, that is, construction with variable coefficients, is left for future work.

APPENDIX
DERIVATION OF (38a)–(38c)

This Appendix presents the derivation of the design formulae in (38a)–(38c).

From (37), it is verified that the polyphase components are represented by

$$H_{m_0}(\mathbf{z}) = h(0,0) + h(0,2)\left(z_{\mathrm{V}}^{1} + z_{\mathrm{V}}^{-1}\right) \tag{46a}$$
$$H_{m_1}(\mathbf{z}) = h(1,1)\left(z_{\mathrm{V}}^{1} + 1\right) \tag{46b}$$
$$H_{m_2}(\mathbf{z}) = h(0,1)\left(z_{\mathrm{V}}^{1} + 1\right) \tag{46c}$$
$$H_{m_3}(\mathbf{z}) = h(1,0). \tag{46d}$$

The PR FIR condition in (28) is, thus, reduced to

$$\begin{aligned}\{h(0,0)h(1,0) - 2h(0,1)h(1,1)\}z_{\mathrm{V}}^{1} \\ + \{h(0,2)h(1,0) - h(0,1)h(1,1)\}\left(z_{\mathrm{V}}^{2}+1\right) \\ = \alpha \mathbf{z}^{-\mathbf{d}_H}.\end{aligned} \tag{47}$$

Note that $\mathbf{d}_H$ has to be

$$\begin{pmatrix} 0 \\ -1 \\ 0 \end{pmatrix}$$

since the second term on the left-hand side has to vanish, and only the first term must remain in order to hold the monomial condition.

The normalization constraint in (34) is satisfied when

$$h(0,0) + 2h(0,1) + 2h(0,2) + h(1,0) + 2h(1,1) = 2. \tag{48}$$

Additionally, the regularity in (35) holds if

$$h(0,0) - 2h(0,1) + 2h(0,2) - h(1,0) + 2h(1,1) = 0. \tag{49}$$

Therefore, we get the following four simultaneous equations:

$$h(0,0)h(1,0) - 2h(0,1)h(1,1) = \alpha \tag{50a}$$
$$h(0,2)h(1,0) - h(0,1)h(1,1) = 0 \tag{50b}$$
$$h(0,0) + 2h(0,2) + 2h(1,1) = 1 \tag{50c}$$
$$2h(0,1) + h(1,0) = 1. \tag{50d}$$

Since six unknown variables, including α, exist, solving the above four equations leaves two unknown variables. Under the assumption that $h(1,0) \neq -1$, letting $h(0,0)$ and $h(1,0)$ be those unknown parameters yields (38a)–(38c) and (39). Note that when $h(1,0) = -1,\ h(0,0)$ has to be one in order to meet the constraints. Practically, such a filter is hardly ever employed as a lowpass filter because it removes low-frequency components in the temporal direction. Thus, the variable range is not taken into account during the optimization process.

ACKNOWLEDGMENT

The authors would like to thank T. Kuge, Dr. Y. Abe, and Prof. H. Kiya for their helpful comments. They would also like to thank the anonymous reviewers, whose critique helped the authors to improve the quality of the presentation.

REFERENCES

[1] A. M. Tekalp, *Digital Video Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1995.

[2] Y. Wang, J. Ostermann, and Y. Q. Zhang, *Video Processing and Communications*, ser. Signal Processing. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[3] R. A. Beuker and I. A. Shah, "Analysis of interlaced video signals and its applications," *IEEE Trans. Image Processing*, vol. 3, pp. 501–512, May 1994.

[4] K. Self, "Prolog to deinterlacing—An overview," *Proc. IEEE*, vol. 86, pp. 1837–1838, Sept. 1998.

[5] G. de Haan and E. B. Bellers, "Deinterlacing—An overview," *Proc. IEEE*, vol. 86, pp. 1839–1857, Sept. 1998.

[6] A. J. Patti, M. I. Sezan, and A. M. Tekalp, "Robust methods for high-quality still from interlaced video in the presence of dominant motion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, pp. 328–342, Feb. 1997.

[7] R. Li, B. Zeng, and M. L. Liou, "Reliable motion detection/compensation for interlaced sequences and its application to deinterlacing," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 23–29, Jan. 2000.

[8] V. Bhaskaran and K. Konstantinides, *Image and Video Compression Standards, Algorithms, and Architectures*, 2nd ed. Boston, MA: Kluwer, 1997.

[9] *Motion JPEG2000 Final Committee Draft 1.0*, Mar. 2001.

[10] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards, and Practice*. Boston, MA: Kluwer, 2002.

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

[12] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, vol. 18, pp. 36–58, Sept. 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*, vol. WA-P2.1, Sept. 2002.

[14] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice-Hall, 1993.

[15] T. Chen and P. P. Vaidyanathan, "The role of integer matrices in multidimensional multirate systems," *IEEE Trans. Signal Processing*, vol. 41, pp. 1035–1047, Mar. 1993.

[16] ——, "Recent developments in multidimensional multirate systems," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 116–137, Feb. 1993.

[17] S. Muramatsu and H. Kiya, "Multidimensional parallel processing methods for rational sampling lattice alteration," in *Proc. IEEE ISCAS*, vol. 1, Apr. 1995, pp. 756–759.

[18] ——, "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.

[19] E. Viscito and J. P. Allebach, "The analysis and design of multidimensional FIR perfect reconstruction filter banks for arbitrary sampling lattices," *IEEE Trans. Circuits Syst.*, vol. 38, pp. 29–41, Jan. 1991.

[20] J. Kovačević and M. Vetterli, "Nonseparable multidimensional perfect reconstruction filter banks and wavelet bases for R^n," *IEEE Trans. Inform. Theory*, vol. 38, pp. 533–555, Feb. 1992.

[21] T. Chen and P. P. Vaidyanathan, "Multidimensional multirate filters and filter banks," *IEEE Trans. Signal Processing*, vol. 41, pp. 1749–1765, May 1993.

[22] J. Kovačević and M. Vetterli, "Nonseparable two- and three-dimensional wavelets," *IEEE Trans. Signal Processing*, vol. 43, pp. 1269–1273, May 1995.

[23] S. Muramatsu, A. Yamada, and H. Kiya, "A design method of multidimensional linear-phase paraunitary filter banks with a lattice structure," *IEEE Trans. Signal Processing*, vol. 47, pp. 690–700, Mar. 1999.

[24] B. L. Evans, R. H. Bamberger, and J. H. McClellan, "Rules for multidimensional multirate structures," *IEEE Trans. Signal Processing*, vol. 42, pp. 762–771, Apr. 1994.

[25] R. A. Gopinath and C. S. Burrus, "On upsampling, downsampling, and rational sampling rate filter banks," *IEEE Trans. Signal Processing*, vol. 42, pp. 812–824, Apr. 1994.

[26] G. Strang and T. Q. Nguyen, *Wavelets and Filter Banks*, 2nd ed. Boston, MA: Wellesley-Cambridge, 1998.

[27] G. Karlsson and M. Vetterli, "Theory of two-dimensional multirate filter banks," *IEEE Trans. Accoust., Speech, Signal Processing*, vol. 38, pp. 925–937, June 1990.

[28] L. Tolhuizen, H. Hollmann, and T. A. C. M. Kalker, "On the realizability of biorthogonal, m-dimensional two-band filter banks," *IEEE Trans. Signal Processing*, vol. 43, pp. 640–648, Mar. 1995.

[29] C. Charoenlarpnopparut and N. K. Bose, "Multidimensional FIR filter bank design using Gröbner bases," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1475–1486, Dec. 1999.

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

[31] H. Kiya, K. Nishikawa, and M. Iwahashi, "A development of symmetric extension method for subband image coding," *IEEE Trans. Image Processing*, vol. 3, pp. 78–81, Jan. 1994.

[32] *Optimization Toolbox (Version 2) Userś Guide*, Sept. 2000.

Shogo Muramatsu (M'99) received the B.E., M.E., and Dr.Eng. 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, Niigata, Japan, 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 the Institute of Electronics, Information, and Communication Engineers, the Information Processing Society of Japan, and the Institute of Image Information and Television Engineers.

Takuma Ishida received the B.E. and M.E. degrees from Niigata University, Niigata, Japan, in 2001 and 2003, respectively. He is currently pursuing the Dr.Eng. degree at Niigata University.

His research interests are in image processing and VLSI architecture.

Hisakazu Kikuchi (M'84) received the B.E. and M.E. degrees from Niigata University, Niigata, Japan, in 1974 and 1976, respectively, and the Dr.Eng. degree in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1988.

From 1976 to 1979, he was with the Information Processing Systems Laboratory, Fujitsu, Ltd., Tokyo. Since 1979, he has been with Niigata University, where he is a Professor with the Department of Electrical Engineering. From 1992 to 1993, he was a visiting scientist at the Electrical Engineering Department, University of 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 the Institute of Electronics, Information, and Communication Engineers, the Institute of Image Information and Television Engineers of Japan, the Japan Society for Industrial and Applied Mathematics, the Research Institute of Signal Processing, and SPIE.