Reversible Conversion between Interlaced and Progressive Scan Formats and Its Efficient Implementation

Hisakazu Kikuchi†, Shogo Muramatsu†, Takuma Ishida† and Tetsuro Kuge††

†Department of Electrical and Electronic Engineering, Niigata University, Niigata 950-2181, Japan

††Science and Technical Research Laboratories, NHK, 1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, Japan

ABSTRACT

A new class of video format conversion between interlaced and progressive scan formats is discussed. It is a kind of the deinterlacing problem and is defined as follows: format conversion of an interlaced scan signal to a progressive scan signal of half the temporal sampling rate to the interlaced signal, and vice versa, under the criterion of equal sampling density. This offers a reversible scan format conversion. The reversibility implies no loss of information contents and is advantageous in video coding. Also, an efficient implementation of the lossless conversion system is presented and deinterlacing artifacts are significantly suppressed.

1 Introduction

As is widely observed, the convergence of television and computers including mobile computing terminals seems to be in progress. In the broadcasting community the interlaced scan format is used in TV, whereas a PC uses progressive scan formats or picture frames[1, 2]. The telecommunication community seems to be a big user of video coding standards and multimedia contents, whichever would be a winner.

Once given this background, one of the challenging problems is a format conversion between interlaced and progressive scanning. Not only the problem is deeply involved with the existing television systems and video camera and displaying systems, but it is of significant relevance to video coding schemes, since a new trend such as observed in Motion JPEG 2000 can accelerate the convergence.

Deinterlacing is necessary to convert the scanning format of motion pictures from interlaced to progressive scan[1]. Historically, deinterlacing has been used to avoid the aliasing and flicker artifacts for better visual quality in broadcast TV signals, and to convert formats between different schemes, for instance, between NTSC and PAL as well as from NTSC to VGA.

On the other hand, intraframe-based motion picture coding is recognized to be preferred in terms of content editing and system complexity. Motion JPEG 2000 (MJP2) is such a candidate[3]. The intraframe-based coding of TV signals requires field merge, and this process unfortunately causes horizontal comb-tooth artifacts at edges of moving objects [2, 4, 5].

This paper deals with a new class of deinterlacing for motion picture coding that preserves the sampling density in interlaced original videos and deinterlaced videos and thus provides the reversible conversion between interlaced and progressive scan formats.

2 Statement of the Problem

Deinterlacing is a kind of sampling rate conversion of video signals[6, 7]. Usually it is defined as the sampling rate up-conversion of an interlaced scan signal to its progressive scan counterpart by interpolating the missing lines in every field[1, 2]. The deinterlaced signal will have twice the vertical sampling rate to the original.

In contrast, the problem to be addressed in this paper is slightly different from the conventional deinterlacing. Given an interlaced scanning video signal, we are trying to find its progressive scanning video signal, but both of these two signals have the same sampling density[8]. Let us formulate our problem in the context of 3-dimensional sampling lattice alteration, after some notations are summarized. Figure 1 shows a sequence of fields of a typical interlaced scan format. The circles and dots are sample points on even and odd lines, respectively. Dashed lines indicate the skipped lines. P_T, P_V, P_H are temporal, vertical, and horizontal sampling periods, respectively.

Some mathematical notations are as follows. $\mathcal{N}(\mathbf{V})$: the set of integer vectors $\mathbf{n}$ in the fundamental parallelepiped generated with a 3×3 non-singular matrix $\mathbf{V}$[6, 9]. $\mathcal{L}(\mathbf{V})$: the set of sample points given by $\mathbf{Vn}$ for all integer vectors $\mathbf{n}$. $J(\mathbf{V})$: the absolute determinant of $\mathbf{V}$, and it is identical to the number of elements in $\mathcal{N}(\mathbf{V})$. $\rho(\mathbf{V})$: the inverse of the absolute determinant of a sampling matrix $\mathbf{V}$, and it is the sampling density.

Assume that the input $X(\mathbf{z})$ has been sampled on the lattice $\mathcal{L}(\mathbf{V})$ generated by the sampling matrix $\mathbf{V} = \begin{pmatrix} P_{\mathrm{T}} & P_{\mathrm{T}} & 0 \\ -P_{\mathrm{V}} & P_{\mathrm{V}} & 0 \\ 0 & 0 & P_{\mathrm{H}} \end{pmatrix}$. Figure 1 shows the sampling lattice. The

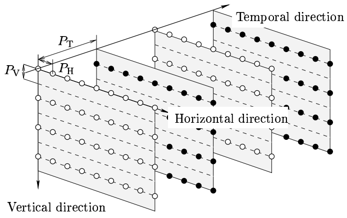


Fig. 1: Interlaced scanning with the sampling lattice $\mathcal{L}(\mathbf{V})$.

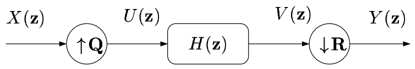


Fig. 2: The process of deinterlacing

sampling density $\rho(\mathbf{V})$ is given by $1/2P_T P_V P_H$.

In the context of this paper, deinterlacing is a kind of 3-D sampling lattice alteration[1, 6, 7]. Figure 2 shows a new class of deinterlacing, where an input interlaced video array $X(\mathbf{z})$ and the output progressive video array $Y(\mathbf{z})$ have the identical sampling density. $X(\mathbf{z})$ is upsampled by a factor $\mathbf{Q}$ and is fed to a filter H(z) followed by $\mathbf{R}$-fold downsampling to produce $Y(\mathbf{z})$.

The upsampler converts the interlaced video array $X(\mathbf{z})$ into the progressive video array. The upsampling factor $\mathbf{Q}$ is of the form $\left(\begin{smallmatrix} 1 & 1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$ for the sampling matrix $\mathbf{V}$. Downsampling is applied in such a way that the overall output array $Y(\mathbf{z})$ has the same sampling density as the original. Hence $J(\mathbf{R})$ has to be two and the lattice orthogonality has to be preserved. $H(\mathbf{z})$ is a 3-D filter that removes the imaging components caused by upsampling and tries to suppress aliasing to be caused by downsampling. The output picture array shown in Fig.2 is thus expressed as follows[8].

$$Y(\mathbf{z}) = \frac{1}{J(\mathbf{R})} \sum_{\mathbf{r} \in \mathcal{N}(\mathbf{R}^T)} H\left((W_1^{\mathbf{r}^T} \mathbf{z})^{\mathbf{R}^{-1}} \right) X\left((W_1^{\mathbf{r}^T} \mathbf{z})^{\mathbf{R}^{-1}\mathbf{Q}} \right), \tag{1}$$

where $W_N = e^{-j\frac{2\pi}{N}}$. The sampling matrix of $Y(\mathbf{z})$ is described by $\mathbf{V}' = \mathbf{V}\mathbf{Q}^{-1}\mathbf{R}$, as long as $\mathbf{Q}$ is nonsingular.

Alternately, let us consider the reverse process of our deinterlacing as shown in Fig.3, and it is referred to as reinterlacing, because a series of our deinterlacing and reinterlacing will perfectly reconstruct the original video signal up to some processing delay.

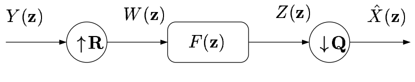


Fig. 3: The process of reinterlacing

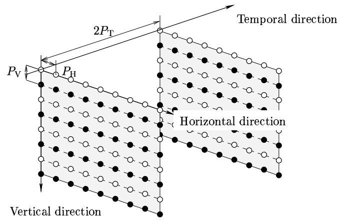


Fig. 4: Sampling lattice $\mathcal{L}(\mathbf{V}')$ of a deinterlaced array with temporal decimation.

This configuration allows us to produce an array $\hat{X}(\mathbf{z})$ on the same sampling lattice as the original $\mathbf{V}$, because $\mathbf{V}'\mathbf{R}^{-1}\mathbf{Q} = \mathbf{V}$, as long as $\mathbf{R}$ is nonsingular.

The output of reinterlacing is expressed as follows.

$$\hat{X}(\mathbf{z}) = \frac{1}{J(\mathbf{Q})} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{Q}^T)} F\left((W_1^{\mathbf{q}^T} \mathbf{z})^{\mathbf{Q}^{-1}} \right) Y\left((W_1^{\mathbf{q}^T} \mathbf{z})^{\mathbf{Q}^{-1}\mathbf{R}} \right) \tag{2}$$

In summary, the problem in this paper is stated as follows. Given an interlaced video over the sampling matrix $\mathbf{V}$, find a pair of deinterlacing and reinterlacing systems that satisfies

$$\hat{X}(\mathbf{z}) = \mathbf{z}^{-\mathbf{n}_D} X(\mathbf{z}) \tag{3}$$

where $\mathbf{n}_D$ is an integer-valued 3-D column vector. This problem setting is advantageous in motion picture coding applications as expected in Motion JPEG 2000 as well as its application to HDTV[4, 5] and potentially in transcoding applications.

3 Analysis and Solutions to The Problem[8]

The problem described in the previous section is analyzed by the multi-dimensional multirate digital signal processing. There are two simplest and most significant forms for our deinterlacing: temporal decimation illustrated in Fig. 4 The analysis of the former is outlined in this section. Detailed analysis and other solutions can be found in [8]. The down sampling factor $\mathbf{R}$ is selected in such a way that it preserves the sampling density involved with the previously shown sampling matrix V and can be actually defined by $\mathbf{R} = \left(\begin{smallmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{smallmatrix}\right)$. This factor implies that $V(\mathbf{z})$ is temporally downsampled, and the sampling matrix of $Y(\mathbf{z})$ is of the form $\mathbf{V}' = \left(\begin{smallmatrix} 2P_T & 0 & 0 \\ 0 & P_V & 0 \\ 0 & 0 & P_H \end{smallmatrix}\right)$. The sampling density $\rho(\mathbf{V}')$ results in $1/2P_T P_V P_H$, and this is identical to the original.

3.1 The Perfect Reconstruction Conditions

For $\hat{X}(\mathbf{z})$ to be aliase-free,

$$F(\mathbf{z}) = \mathbf{z}^{-\begin{pmatrix} 2S_T+1 \\ 2S_V \\ S_H \end{pmatrix}} KH\left(W_2^{(0\ 1\ 0)} \mathbf{z} \right) \tag{4}$$

has to be satisfied, where K, S_T, S_V and S_H are integers. The distortion-free condition is then expressed by

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

for any 3-D integer-valued column vector $\mathbf{d}$, where $E_{\mathbf{m}_k}(\mathbf{z})$ is the k-th type-I polyphase component of $H(\mathbf{z})$ with the factor $\mathbf{S} = \mathrm{diag}(2,2,1)$ so that

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

The resulting delay $\mathbf{n}_\mathrm{D}$ is expressed by $\mathbf{n}_\mathrm{D} = \mathbf{Q}^{-1}\mathbf{S}\left\{\mathbf{d} + \left(\begin{smallmatrix}S_\mathrm{T}\\S_\mathrm{V}\\S_\mathrm{H}\end{smallmatrix}\right) + \left(\begin{smallmatrix}1\\0\\0\end{smallmatrix}\right)\right\}$, where $\mathbf{Q}^{-1}\mathbf{S} = \left(\begin{smallmatrix}1&-1&0\\1&1&0\\0&0&1\end{smallmatrix}\right)$.

3.2 Design of Filters

There is a lot of freedom for designing a deinterlacing filter $\mathrm{H}(\mathbf{z})$. In the case of temporal decimation, if one imposes the normalization in terms of signal power, the regularity condition to prevent from checkerboard effect[10], and vertical symmetry to afford thesymmetric border extension[11], then the design problem reduces to finding just two coefficients of $h(0,0)$ and $h(1,0)$. For a simple example, an 8-tap 2-D filter

$$\begin{aligned} H(\mathbf{z}) &= h(0,0) + h(1,0)z_\mathrm{T}^{-1} + h(0,1)(z_\mathrm{V}^{1} + z_\mathrm{V}^{-1}) \\ &\quad + h(1,1)z_\mathrm{T}^{-1}(z_\mathrm{V}^{1} + z_\mathrm{V}^{-1}) + h(0,2)(z_\mathrm{V}^{2} + z_\mathrm{V}^{-2}) \end{aligned} \tag{7}$$

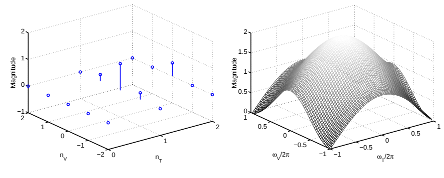
can be considered, and the above two coefficients can be found by some nonlinear optimization with respect to temporal and vertical lowpass filtering. One simple but effective example is a choice of $h(0,0) = 1$ and $h(1,0) = 1/2$. The other coefficients are derived by

$$h(0,1) = \{1 - h(1,0)\}/2, \tag{8a}$$
$$h(0,2) = \{1 - h(0,0)\}\{1 - h(1,0)\}/2\{1 + h(1,0)\}, \tag{8b}$$
$$h(1,1) = h(1,0)\{1 - h(0,0)\}/\{1 + h(1,0)\}, \tag{8c}$$
$$K = \{1 + h(1,0)\}/h(1,0)\{2h(0,0) - (1 - h(1,0))\} \tag{8d}$$

and the impulse responses and frequency responses of this deinterlacing filter and its dual are shown in Fig. 5.

4 Efficient Implementations

There seems to exist two major driving forces that justifies the problem setting discussed in this paper: prevalence of HDTV and Motion JPEG 2000. Since the reversible deinterlacing is different from the traditional deinterlacing problem, some possible application scenarios are appropriate to be exposed before the presentation of efficient implementations.



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

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

Fig. 5: Responses of the filter pair

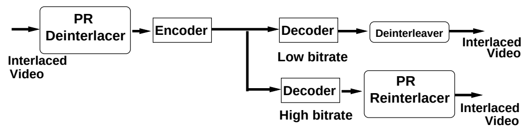


Fig. 6: An application system

4.1 Feasible System Configurations

The reversible deinterlacing can be applied to an HDTV system in combination with Motion JPEG 2000 as illustrated in Fig. 6. It assumes full and partial decoding of a single coded bit-stream. The perfect reconstruction (PR) deinterlacer converts an interlaced original video to its progressive scan format before scalable encoding. At a receiver, full/partial decoding is performed depending on individual demands and environments. For lossless decoding and high bit rate decoding, the PR reinterlacer converts the decoded progressive scan video to its counterpart in the interlaced scan format. The reinterlacer suppresses the aliasing effects that can take place in the field merge before encoding and in the field decomposition after decoding without any loss in the spatial resolution. For low bit rate decoding, a simple deinterleaver can be a substitution of the PR deinterlacer for reduction in system complexity. Also, if the front-end terminal of a user is a type of PC rather than TV, there is no use of reinterlacing. In short, flicker artifacts are suppressed in low bit rate decoding, and the spatial resolution is maintained in high bit rate decoding.

4.2 Implementations

An efficient implementation of the reversible deinterlacing/reinterlacing system of Eq. (7) is illustrated in Fig. 7, where left and right parts show deinterlacing and reinterlacing, respectively. White circles and dots indicate pixels in even and odd fields, respectively. This implementation permits the symmetric border extension to the top and bottom data and is also free from redun-

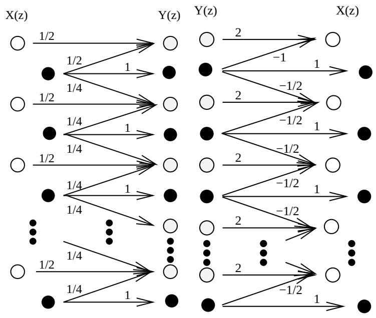


Fig. 7: Efficient implemantation

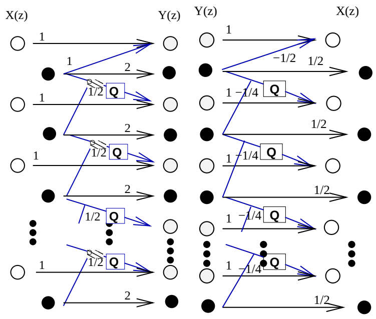


Fig. 8: Lossless implemantation

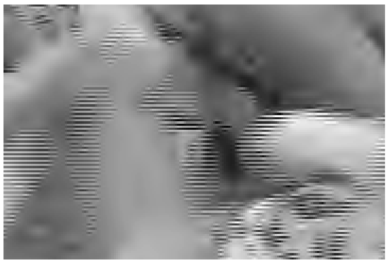
dant null interpolation. Also, an entire body of processing will proceed in the form of in-place computation, and every filter coefficient is simply the power of two. A lossless implementation is also possible as showin in Fig. 8, where Q denotes rounding to integers.

4.3 Experiments

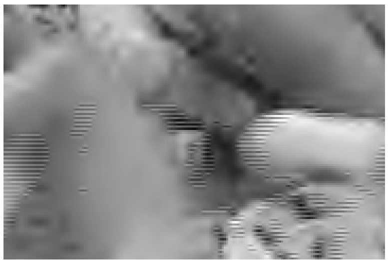
A few experiments have been developed to demonstrate that PR deinterlacing offers better visual quality than just a simple field merge/split system, when partial decoding is applied at a receiver. The test video sequence is *Football* of 720x480-pixel per frame of which temporal rate is 30 frames/sec and 60 fields/sec. The JPEG 2000 encoder/decoder is JJ2000 in [12]. Fig. 9 shows the results encoded at 2.0bpp and then decoded at 0.1bpp. Note that pictures were magnified for ease of visual inspection. The come-tooth interlacing artifacts have been significantly suppressed in part (b) of the figure.

5 Conclusions

A reversible scan format conversion system between interlaced and progressive scan videos are described in the context of 3D sampling rate alteration and its efficient implementations have been presented. The reversible deinterlacing/reinterlacing system has the perfect reconstruction property, and suppresses the interlacing artifacts in low bit rate decoding and also keeps the temporal resolution in high bit rate decoding, when a JPEG



(a) Field interleaving



(b) PR deinterlacing

Fig. 9: Decoded frames(0.1bpp)

2000 encoder is applied as Motion JPEG.

References

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

[2] G. de Haan and E. B. Bellers: "Deinterlacing-An Overview", *Proc. IEEE*, vol. 86, No. 9, pp. 1837–1857, Sep. 1998.

[3] T. Fukuoka, K. Katoh, S. Kimura, K. Hosaka and A. Leung: "Motion-JPEG 2000 Standardization and Target Market", *Proc. ICIP*, No. TA0208, 2000.

[4] T. Kuge and H. Hoshino: "A Study on HDTV Moving Picture Coding Using the JPEG 2000", *Proc. IEICE Spring Conv.*, D-11-18, p. 18, Mar. 2001.

[5] T. Kuge: "Several Considerations on the Visual Distortion by Wavelet Picture Coding -Several Problems of the Application to the Interlace HDTV and the Ultra-high Definition Images-", *ITE Tech. Report*, Vol. 25, No. 79, pp. 33–38, Nov. 2001.

[6] P. P. Vaidyanathan: *Multirate Systems and Filter Banks*, Prentice Hall, 1993.

[7] S. Muramatsu and H. Kiya,"Parallel Processing Techniques for Multidimensional Sampling Lattice Alteration Based on Overlap-add and Overlap-save Methods," *IEICE Trans. Fundamentals*, vol. E78-A, no. 8, pp. 934–943, Aug. 1995.

[8] S. Muramatsu, S. Sasaki, Z. Jie and H. Kikuchi: "Deinterlacing with Perfect Reconstruction Property", *16th DSP Symp.*, Okinawa, pp. 427–432, Nov. 2001.

[9] 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*, **47**, pp. 690–700, Mar. 1999.

[10] Y. Harada, S. Muramatsu and H. Kiya: "Multidimensional Multirate Filter and Filter Bank without Checkerboard Effect", *IEICE Trans. Fundamentals*, **E81-A**, 8, pp. 1607–1615, 1998.

[11] H. Kiya, K. Nishikawa and M. Iwahashi: "A Development of Symmetric Extension Method for Subband Image Coding", *IEEE Trans. Image Processing*, **3**, 1, pp. 78–81, 1994.

[12] EPFL, Ericsson and Canon, http://jj2000.epfl.ch, Nov. 2000.