

Journal of Electronic Imaging

SPIDigitalLibrary.org/jei

Spatiotemporal deinterlacing using a maximum *a posteriori* estimator based on multiple-field registration

Ho-Taek Lee
Dong-bok Lee
Byungju Lee
Byung Cheol Song



Spatiotemporal deinterlacing using a maximum *a posteriori* estimator based on multiple-field registration

Ho-Taek Lee, Dong-bok Lee, Byungju Lee, and Byung Cheol Song*

Inha University, School of Electronic Engineering, Nam-gu, Incheon 402-751, Republic of Korea

Abstract. This paper proposes an accurate deinterlacing algorithm using a maximum *a posteriori* (MAP) estimator. First, we produce accurate motion vector fields between the current field and adjacent fields by employing an advanced motion compensation scheme that is suitable for an interlaced format. Next, the progressive frame corresponding to the current field is found via the MAP estimator based on the derived motion vector fields. Here, in order to obtain a stable solution, well-known bilateral total variation–based regularization is applied. Then, at a specific mode decision step, it is decided whether the result from the aforementioned temporal deinterlacing is acceptable or not. Finally, if the temporal deinterlacing is determined to be inappropriate by the mode decision, a typical spatial deinterlacing is applied instead of the MAP estimator-based temporal deinterlacing. Experimental results show that the proposed algorithm provides at maximum 2 dB higher PSNR than a cutting-edge deinterlacing algorithm, while providing better visual quality than the latter. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.JEI.22.4.043038](https://doi.org/10.1117/1.JEI.22.4.043038)]

Keywords: deinterlacing; maximum *a posteriori* estimator; multiple-field registration.

Paper 13500 received Sep. 5, 2013; revised manuscript received Nov. 18, 2013; accepted for publication Nov. 25, 2013; published online Dec. 20, 2013.

1 Introduction

Deinterlacing is an important technique because it converts interlaced video sequences into progressive sequences for progressive digital display devices, such as LCDs, plasma display panels, and organic light-emitting diode TVs. However, visually annoying artifacts, such as edge flicker, jaggling, blurring, and feathering, often occur due to imperfect deinterlacing. Thus, a critical issue related to deinterlacing is removing such artifacts as much as possible.

During the past several decades, numerous deinterlacing algorithms have been developed.¹ We can categorize the existing algorithms into three groups as follows: spatial deinterlacing,^{2–8} motion-adaptive (MA) deinterlacing,^{9–11} and motion compensation (MC)-based deinterlacing.^{12–22} Spatial deinterlacing algorithms have the advantages of simple operation and straightforward integration into hardware. The edge-based line averaging (ELA) algorithm is one of the most popular deinterlacing algorithms in this category. However, since all the spatial deinterlacing schemes interpolate the missing pixels by using only intrafield information, the interpolated pixels in the texture area tend to be somewhat blurred.

Primitive temporal approaches such as MA deinterlacing algorithms have been devised to overcome the drawbacks of spatial deinterlacing algorithms by using temporal correlation.^{9–11} MA deinterlacing methods detect the existence of motion and then use either an intrafield method, i.e., a typical spatial deinterlacing, or an interfield method. Here, the interfield method is a simple interfield interpolation without motion compensation. However, this MA deinterlacing method still suffers from blur or jaggling artifacts because

it should apply the intrafield method when interpolating missing pixels in moving objects.

Recently, more advanced temporal deinterlacing methods that are based on some form of motion-compensated techniques have become popular.^{13–22} For example, Chang et al. presented an adaptive four-field global/local motion-compensated approach where the same parity four-field motion detection and four-field motion estimation detect static areas and fast motion by four reference fields, and global motion estimation detects camera panning and zooming motions.¹³ However, this algorithm may give rise to feathering artifacts when the assumption of strong continuity of local motion does not work. Fan and Chung proposed a temporal deinterlacing algorithm using strong spatial-temporal correlation-assisted motion estimation.¹⁷ To our knowledge, Fan's algorithm is the state-of-the-art deinterlacing scheme. However, since conventional temporal deinterlacing algorithms inherently cannot compensate for registration error between the current field and its reference fields, they have a limitation in further improving visual quality.

On the other hand, maximum *a posteriori* (MAP) estimation has recently been applied to numerous image reconstruction techniques, e.g., deblur, denoising, and super-resolution. In statistics, MAP estimation is a method of estimating the parameters of a statistical model. When applied to a data set and given a statistical model, an MAP estimator provides estimates for the model's parameters. For example, MAP estimators are often used for superresolution reconstruction.^{23,24} Superresolution restoration aims to solve the following problem: given a set of observed low-resolution images, estimate a high-resolution image. The observed low-resolution images are regarded as degraded observations of a real, high-resolution texture. These degradations typically include geometric warping, optical blur, spatial sampling, and noise. Given several such observations, the MAP estimate

*Address all correspondence to: Byung Cheol Song, E-mail: bcson@inha.ac.kr

of the superresolution image may be obtained such that, when reprojected back into the images via a generative imaging model, it minimizes the difference between the actual and predicted observations.²⁴ Note that if accurate motion information of a certain missing pixel is given, the MAP-based superresolution can reconstruct the missing pixel such that it is very close to its original.

We showed that if the MAP estimator is applied to temporal deinterlacing, subjective visual quality as well as objective visual quality can be improved by minimizing inherent registration errors in missing pixels in Ref. 20. In this paper, we propose an advanced MAP-estimator-based deinterlacing algorithm using high-performance MC method and strong mode decision. The proposed algorithm consists of four steps. First, accurate multiple-field registration is performed between the current field and its neighboring fields. Second, the progressive frame corresponding to the current field is reconstructed via an L_1 -norm-based MAP estimator using the predicted motion fields. Here, in order to obtain an optimal solution, the well-known steepest descent algorithm is employed. Also, bilateral total variation (BTV)-based regularization is applied to obtain a stable solution and preserve edges well. Third, a mode decision module determines whether the result from the aforementioned temporal deinterlacing is acceptable or not. The proposed mode decision relies on three factors: feathering artifacts, registration errors, and motion vector (MV) correlation. Fourth, if the temporal deinterlacing is determined to be inappropriate by the mode decision, a typical spatial deinterlacing based on edge-directional interpolation is applied instead of the MAP estimator-based temporal deinterlacing. Experimental results show that the proposed algorithm obtains at maximum 2 dB higher peak signal-to-noise ratio (PSNR) than the state-of-the-art spatiotemporal deinterlacing algorithm, i.e., Fan's algorithm,¹⁷ while providing better visual quality.

The remainder of this paper is organized as follows. Section 2 describes the proposed deinterlacing algorithm in detail. Section 3 provides intensive experimental results. Finally, we conclude this paper in Sec. 4.

2 Proposed Algorithm

As illustrated in Fig. 1, the proposed deinterlacing algorithm has a structure similar to a typical spatiotemporal deinterlacing scheme based on a hard decision between spatial deinterlacing and temporal deinterlacing. The main contribution

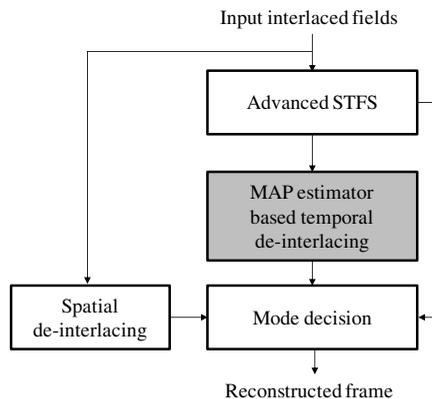


Fig. 1 Block diagram of the proposed deinterlacing algorithm.

of this paper is that we can reconstruct even registration errors in temporal deinterlacing by employing a MAP estimator. Note that previous temporal deinterlacing methods output motion-compensated pixels only. The proposed algorithm, meanwhile, can produce outstanding visual quality, especially around diagonal edges, while significantly reducing jaggging or feathering artifacts in comparison with the state-of-the-art deinterlacing schemes.

The proposed algorithm consists of four steps, as seen in Fig. 1. First, an advanced motion estimation (ME) algorithm, which is an improved version of the spatial-temporal correlation-assisted search (STFS) proposed in Ref. 17, is applied to the current field and adjacent fields. Second, based on the estimated motion information, MAP estimator-based temporal deinterlacing for the current field is performed. Third, on a block basis, a mode decision module determines whether the result from the temporal deinterlacing is acceptable or not. Fourth, if the temporal deinterlacing result is determined to be unacceptable, a typical spatial deinterlacing is applied. The following subsections describe each step of the proposed algorithm in more detail.

2.1 Registration Using the Advanced STFS

2.1.1 Conventional STFS

Prior to describing the advanced STFS, we explain the conventional STFS (Ref. 17) in detail as follows. As illustrated in Fig. 2(a), three consecutive fields, i.e., f_{n-1} , f_n , f_{n+1} , are involved in the estimation of the motion trajectory. The bidirectional ME and the two kinds of unidirectional ME, i.e., forward and backward ME, are combined in order to fully exploit the information from the three fields.

Let SAD_P and SAD_N be the sum of absolute differences (SAD) between two blocks used in the backward ME and forward ME, respectively, and let SAD_B be the SAD between two blocks used in the bidirectional ME, where these terms are defined as follows:

$$\begin{aligned} SAD_B(\mathbf{q}) &= \sum_{(x,y) \in \mathbf{b}} |f_{n+1}(x+m, y+n) - f_{n-1}(x-m, y-n)| \\ SAD_N(\mathbf{q}) &= \sum_{(x,y) \in \mathbf{b}} |f_n(x, y) - f_{n+1}(x+m, y+n)| \\ SAD_P(\mathbf{q}) &= \sum_{(x,y) \in \mathbf{b}} |f_n(x, y) - f_{n-1}(x-m, y-n)|, \end{aligned} \quad (1)$$

where (x, y) denotes a position in the current block \mathbf{b} , and $\mathbf{q} \equiv (m, n)$ indicates a candidate MV within the given search

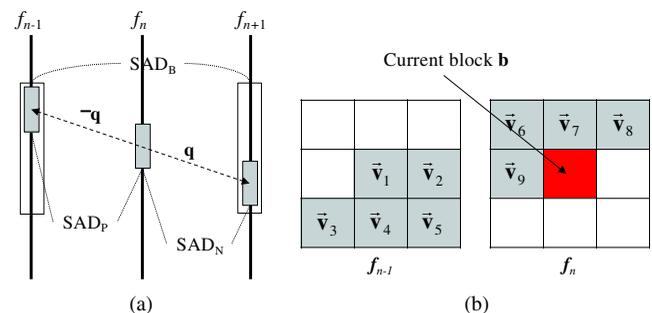


Fig. 2 Conventional spatial-temporal correlation-assisted search (STFS) scheme.

range. The best MV $\hat{\mathbf{q}} = (\hat{m}, \hat{n})$ for the block \mathbf{b} is then obtained by minimizing the following cost:

$$\hat{\mathbf{q}} = \arg \min_{\mathbf{q}} [\text{SAD}(\mathbf{q})], \quad (2)$$

where

$$\text{SAD}(\mathbf{q}) = \text{SAD}_B(\mathbf{q}) + \text{SAD}_N(\mathbf{q}) + \text{SAD}_P(\mathbf{q}). \quad (3)$$

Note that the information from the three consecutive fields is involved in the estimation of the motion trajectory. From Fig. 2(a), $\hat{\mathbf{q}}$ is chosen as the initial MV candidate $\vec{\mathbf{v}}_{\text{ini}}$. For each matching block, there are four spatial MV candidates and five temporal MV candidates, as in Fig. 2(b). The spatial/temporal bidirectional MV candidates are sorted according to the number of occurrences. For the current block, the matching error corresponding to each spatial/temporal MV candidate $\vec{\mathbf{v}}_{\text{ST}}$ is compared to the matching error of the initial MV candidate $\vec{\mathbf{v}}_{\text{ini}}$ according to Eq. (4). The comparison is performed starting from the candidate with the maximum number of occurrences.

$$|\text{SAD}(\vec{\mathbf{v}}_{\text{ST}}) - \text{SAD}(\vec{\mathbf{v}}_{\text{ini}})| < T_1, \quad (4)$$

where T_1 is a predetermined threshold that is a function of $\text{SAD}(\vec{\mathbf{v}}_{\text{ini}})$. If Eq. (4) is satisfied, $\vec{\mathbf{v}}_{\text{ST}}$ is chosen as the final MV. Otherwise, the candidate with the second largest number of occurrences is compared to the initial candidate, and so on. If all the spatial/temporal MV candidates fail to pass the matching error check, $\vec{\mathbf{v}}_{\text{ini}}$ is chosen as the final MV.

2.1.2 Advanced STFS

In order to obtain more accurate and denser MVs than the conventional STFS, we apply the overlapped motion compensation scheme to the STFS. Note that MV estimation is performed on an 8×8 block basis, but the matching block size is 16×16 . As shown in Fig. 3, the best MV for a specific 8×8 block is obtained by block matching for the extended 16×16 block with the 8×8 block as a center. The conventional STFS is used for this block matching. Finally, the estimated MV for the extended 16×16 block is allocated to the 8×8 center of the matching block. Table 1 shows the superiority of the advanced STFS to the conventional STFS in terms of PSNR. For this experiment, we employed the same progressive common intermediate format (CIF) sequences as in Sec. 3 and produced their interlaced versions in the same fashion as described in Sec. 3. For each method, we computed the

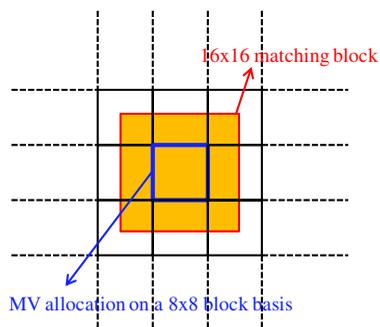


Fig. 3 The overlapped matching of the advanced STFS.

PSNR between the motion-compensated frame and the original frame. We compared the averaged PSNR for the first 300 frames of each sequence as an objective evaluation measure. For example, Table 1 shows that the advanced STFS outperforms the conventional STFS by 0.8 dB at maximum for the *container* sequence. This is because the advanced STFS can produce more accurate and denser MVs than the conventional STFS. Thus, we applied this advanced STFS for registration of the proposed temporal deinterlacing, which is described in the following subsection.

2.2 MAP Estimator-Based Temporal Deinterlacing

The key point of the proposed algorithm is to reconstruct even registration errors in temporal deinterlacing by employing a MAP estimator, whereas previous temporal deinterlacing methods output motion-compensated pixels only. In order to formulate the MAP estimation problem for temporal deinterlacing, we should define the acquisition model of an interlaced field. First, we assume that a progressive frame F may have a motion model M with adjacent frames. Next, a space-invariant point spread function (PSF) H is applied to F for antialiasing. Finally, an interlaced field f is generated by applying an interlace decimation operator D and adding a noise component V . Note that the decimation operator D alternatively subsamples even lines and odd lines, and the decimation operation is defined as follows:

$$f_n(x, y) = \begin{cases} F_n(x, y), & \text{if } x \bmod 2 = n \bmod 2 \\ 0, & \text{otherwise} \end{cases}. \quad (5)$$

According to Eq. (5), the n 'th field f_n is derived from the n 'th frame F_n . The current field and its neighbor fields are then defined as

$$f_n = D_n H_n M_n F + V_n \quad n = 1, \dots, N. \quad (6)$$

Let the processing block size be $b \times b$ and the scaling ratio be set to $r:1$. Then, F in Eq. (6) has a dimension of $[r^2 b^2 \times 1]$, which is arranged in lexicographic order. The $[r^2 b^2 \times r^2 b^2]$ matrix M_n is the geometric motion operator

Table 1 Peak signal-to-noise ratio (PSNR) comparison between conventional spatial-temporal correlation-assisted search (STFS) and the advanced STFS (dB).

Name	Conventional STFS	Advanced STFS
<i>Foreman</i>	37.69	38.02
<i>Mobile</i>	28.77	28.68
<i>M & D</i>	47.37	47.47
<i>Stefan</i>	31.07	31.23
<i>Coastguard</i>	38.25	38.32
<i>Table tennis</i>	35.62	36.01
<i>Container</i>	46.59	47.42
Average	37.91	38.16

between the original progressive frame F and the n 'th interlaced field f_n of size $[b^2 \times 1]$. The PSF is modeled by the $[r^2 b^2 \times r^2 b^2]$ matrix H . The $[b^2 \times r^2 b^2]$ matrix D_n represents the decimation operator and the $[b^2 \times 1]$ matrix V_n denotes the system noise. Based on this model, we propose the MAP estimator-based temporal deinterlacing algorithm.

In general, a MAP estimator for the original progressive frame F given the observed interlaced fields f_n can be formulated to find the MAP estimate \hat{F} via the following L_p minimization:

$$\hat{F} = \arg \min_F \left[\sum_{n=1}^N \|D_n H M_n F - f_n\|_p^p + \lambda \Gamma(F) \right], \quad (7)$$

where λ is a control parameter and N is the number of reference fields for MAP estimation. Note that $H_n = H$ because a common PSF is assumed for all fields without loss of generality. The second term is known as regularization. Consider regularization in deinterlacing is very useful for finding a stable solution.²³ Among many possible regularization terms, we select BTV-based regularization, which results in a full progressive frame with sharp edges and is easy to implement. The BTV-based regularization term is defined by

$$\sum_{l=-\tau}^{\tau} \sum_{\substack{m=0 \\ l+m \geq 0}}^{\tau} \alpha^{|l|+|m|} \|F - S_x^l S_y^m F\|_1, \quad (8)$$

where τ and α are parameters to control the regularization level, and S_x^l and S_y^m are operators to shift by l and m pixels in the horizontal and vertical directions, respectively. This paper employs L_1 -norm regularization owing to its simplicity and edge-preserving property. Also, L_1 -norm is applied to the data term, i.e., the first term of Eq. (7), to minimize sensitivity to noise, i.e., $p = 1$.

In addition, we compare BTV-based regularization of Eq. (8) with a typical TV-based regularization²³ to show the superiority of the BTV-based regularization quantitatively and qualitatively. Table 2 compares BTV-based regularization and TV-based regularization in terms of PSNR.

$$\hat{F}_{t+1} = \hat{F}_t - \beta \left\{ \begin{array}{l} \sum_{n=1}^N M_n^T H^T D_n^T \text{sign}(D_n H M_n \hat{F}_t - f_n) \\ + \lambda \sum_{l=-\tau}^{\tau} \sum_{\substack{m=0 \\ l+m \geq 0}}^{\tau} \alpha^{|l|+|m|} [I - S_y^m S_x^{-l}] \text{sign}(\hat{F}_t - S_x^l S_y^m \hat{F}_t) \end{array} \right\}, \quad (9)$$

where β is a scalar defining the step size in the direction of the gradient and I is an identity matrix. The matrices M , H , and D and their transposes can be exactly interpreted as direct image operators, such as shift, blur, and decimation.²³ Noting and implementing the effects of these matrices as a sequence of operators spares us from explicitly constructing them as matrices. This property allows the proposed temporal deinterlacing method to be implemented in an extremely fast and memory efficient way. After several iterations according to Eq. (9), an optimal progressive frame \hat{F} corresponding to the current field is reconstructed from N consecutive fields including the current field. Finally, MAP estimator-based temporal

Table 2 PSNR comparison between total variation (TV) regularization and bilateral total variation (BTV) regularization (dB).

Name	TV regularization	BTV regularization
<i>Foreman</i>	37.75	38.87
<i>Stefan</i>	31.89	32.03
<i>Carphone</i>	33.94	34.15
<i>Flower</i>	31.20	31.61
<i>Football</i>	33.08	33.10
<i>Highway</i>	36.39	36.73
<i>Salesman</i>	37.96	38.61
Average	34.60	35.01

The PSNR value in Table 2 is the average PSNR for the first 50 deinterlaced frames per sequence. We can see that BTV-based regularization provides at maximum 1.1 dB and on average 0.4 dB higher PSNRs than TV-based regularization. For example, Fig. 4 shows the deinterlacing results for *carphone* and *foreman* sequences. Note that the TV-based regularization causes some artifacts for flat areas due to its sensitivity to noise. On the other hand, the proposed BTV-based regularization seldom shows such a phenomenon because it is inherently robust against noise and preserves the details such as edges and textures better than TV-based regularization. As a result, we can achieve high-quality temporal deinterlacing via this robust regularized MAP estimator.

We use the steepest descent algorithm to find the solution to Eq. (7). We can thus obtain an optimal solution in the following iterative manner. That is, the $(t+1)$ 'th MAP estimate \hat{F}_{t+1} is derived from the t 'th estimate \hat{F}_t .

deinterlacing is completed by replacing the missing field pixels in the current field with the corresponding pixels in \hat{F} .

2.3 Mode Decision

Conventional spatiotemporal deinterlacing methods often suffer from feathering, blur, and jaggling artifacts caused by false mode decision. For instance, feathering artifacts occur near object boundaries due to occlusion or inaccurate MVs. Mode decision, hence, significantly affects the visual quality of spatiotemporal deinterlacing algorithms. Fan and Chung employed the so-called slope detector (SD)-based

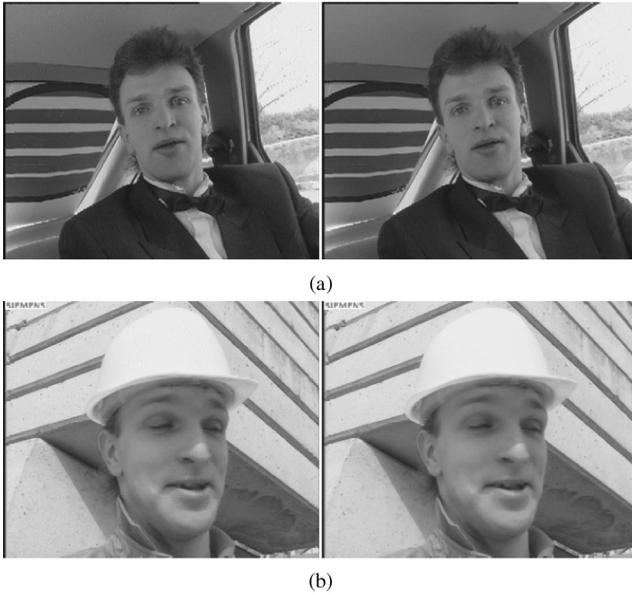


Fig. 4 The deinterlaced frames according to regularization term.

feathering artifact detector due to its low complexity and high detection ability.¹⁷

This paper presents a stronger mode decision method that additionally takes into account mean of absolute differences (MAD) values, MV correlation, and MAD correlation. Figure 5 illustrates the proposed mode decision scheme where the MV reliability is examined on a block basis in two steps. Assume that the MVs are produced by the advanced STFS. First, for each block that is motion-compensated by the advanced STFS, feathering artifacts are detected by the aforementioned SD-based detector of Ref. 17. If a feathering artifact is not detected, temporal deinterlacing is applied to the block. Otherwise, additional examination is performed with more information. The reason why we chose such a conservative approach is that the blur phenomenon caused by spatial deinterlacing is visually less annoying than feathering artifacts. The second examination procedure is explained in greater detail below.

Since the SD-based detector depends on the motion-compensated image only, it may cause false positives in detecting various feathering artifacts. We hence propose an additional step to reduce the false positive rate of the first step by taking into account MAD values, MV correlation, and MAD correlation. As shown in Fig. 5, if the MAD of the current block is sufficiently small, i.e., $MAD(\hat{v}) < \delta_1$, and the MAD and

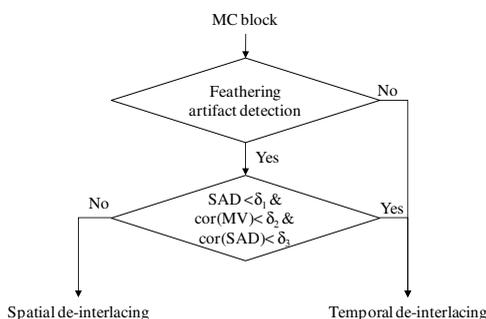


Fig. 5 The proposed mode decision scheme.

MV of the current block are highly correlated with those of its neighbors, i.e., $cor(MV) < \delta_2$ & $cor(MAD) < \delta_3$, we determine that the current block has a reliable MV. Let $cor(MV)$ be the L_1 -norm distance between the current MV and the component-wise median of MVs of its eight-connected blocks, while $cor(MAD)$ indicates the L_1 -norm distance between $MAD(\hat{v})$ and the average MAD for the eight-connected neighbor blocks. In this paper, δ_1 , δ_2 , and δ_3 are empirically set to 20, 5, and 5, respectively. Therefore, since this second decision step further examines the MV reliability independently of the motion-compensated image, it can effectively prevent the entire mode decision from being trapped in false positives.

2.4 Spatial Deinterlacing

For spatial deinterlacing of the proposed algorithm, we adopted a block-based directional interpolation algorithm proposed by Chen and Tai in Ref. 14. This subsection briefly introduces Chen's algorithm. Figure 6 shows a deinterlaced frame, where the dotted line indicates the missing scan line that needs to be interpolated and the solid lines indicate the original field data. It is assumed that the missing pixel $f_n(x, y)$ is centered in a target block BC; BU_q and BL_q are defined as the corresponding upper and lower referenced blocks, respectively, and q is assumed to be a candidate directional vector between the BC and its upper candidate block. Note that the blocks BC, BU_q , and BL_q only contain pixels in f_n . The best directional vector \hat{q} is detected using the following equation:

$$\hat{q} = \arg \min_{q \in \Omega} [SAD_U(q) + SAD_L(q)], \tag{10}$$

where $SAD_U(q)$ and $SAD_L(q)$ denote the SAD between BC and BU_q , and the SAD between BC and BL_q , respectively. The search range of q , i.e., Ω , includes 23 directional vectors so as to detect edges whose degrees are greater than 7.5. Let $SAD(\hat{q})$ be the minimal cost according to Eq. (10). When $SAD(\hat{q})$ is greater than a certain threshold θ_1 , or the difference between $SAD(\hat{q})$ and $SAD(\hat{q} + 90 \text{ deg})$ is not greater than a threshold θ_2 , the selected \hat{q} is not reliable, and the typical line average filter is used to construct the final result. In this paper, θ_1 and θ_2 are empirically set to 30 and 10, respectively.

3 Experimental Results

3.1 Experimental Condition

First, for an objective quality evaluation, we used 12 progressive CIF sequences; *foreman*, *mobile*, *mother and daughter (M&D)*, *Stefan*, *coastguard*, *table tennis*, *container*, *flower garden*, *football*, *highway*, *Paris*, and *tempete*. We generated the interlaced fields by applying a simple

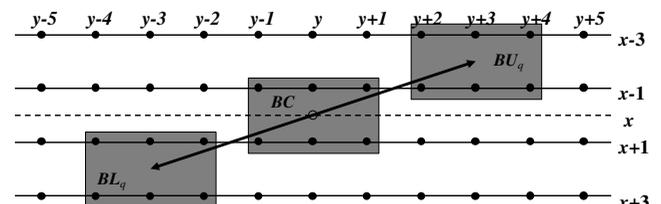


Fig. 6 Edge-based spatial deinterlacing.

low-pass filter of {1, 2, 1} to those progressive frames and alternatively subsampling the low-pass filtered frames. After deinterlacing, we computed the PSNR between the reconstructed progressive frame and the original frame. Note that only Y components are treated here. We chose the averaged PSNR for the first 300 frames of each sequence as an objective evaluation measure.

We compared the proposed algorithm with LA, ELA, Lee's algorithm,⁹ Chang's algorithm,¹³ Chen's algorithm,¹⁴ Fan's algorithm,¹⁷ Yang's algorithm,¹⁸ Mohammadi's algorithm,²¹ and Trocan's algorithm.²² The number of reference fields for the proposed temporal deinterlacing was set to 3, i.e., the current field and its temporally previous and next fields.

The search range for registration in the proposed algorithm and Chang's algorithm was ± 32 both horizontally and vertically. All the block sizes for matching in the advanced STFS were set to 16×16 and the block sizes of overlapping are set to 8×8 . In Eq. (8), α and τ are fixed to 0.5 and 2, respectively. Figure 7 shows the influence of a parameter α on the overall performance of the proposed algorithm for four test sequences, i.e., *highway*, *carphone*, *foreman*, and *football*. For this experiment, when deinterlacing the first field of each sequence, we progressively changed α from 0.1 to 1.0 and tracked the PSNR values accordingly. From Fig. 7, we can observe that as α becomes larger than 0.5, the PSNR starts to decrease. So, we set α to 0.5. Through a similar experiment, we found that another parameter τ rarely affects the overall performance of the proposed algorithm. So, we fixed τ to an acceptable value. Table 3 shows the appropriate values of the other parameters, i.e., β and λ chosen for each sequence to maximize the temporal deinterlacing performance in terms of PSNR. Note that β and λ should be determined on a shot basis in a sequence. For instance, since the *foreman* and *table tennis* sequences consist of two different shots, we derived and applied two different parameters to those sequences, as given in Table 3.

Figure 8 shows the effects of the parameters in Eq. (7) on the overall performance of the proposed temporal deinterlacing. This experiment was performed for a field in the first shot of the *foreman* sequence. From Fig. 8(a), we can observe that if the optimal parameters are used, the PSNR performance is very stable. On the contrary, Fig. 8(b) shows that if nonoptimal parameters, i.e., the parameters of the second shot, are employed, the PSNR performance drastically decreases as the iterations go on, and the peak PSNR also becomes low in comparison with that in Fig. 8(a). From Fig. 8, we can also find that the termination

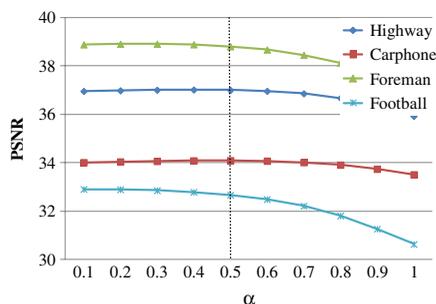


Fig. 7 Peak signal-to-noise ratio (PSNR) performance according to α for four video sequences.

Table 3 Parameter setting for temporal deinterlacing.

Name	Shot #1		Shot #2		Iteration
	β	λ	β	λ	
<i>Foreman</i>	1.3	0.2	1.5	0.1	5
<i>Mobile</i>	1.1	0.05	—	—	4
<i>M&D</i>	0.5	0.2	—	—	4
<i>Stefan</i>	1.0	0.3	—	—	2
<i>Coastguard</i>	0.2	0.2	—	—	4
<i>Table tennis</i>	0.5	0.2	0.7	0.1	5
<i>Container</i>	0.1	0.2	—	—	4
<i>Flower garden</i>	1.5	0.05	—	—	50
<i>Football</i>	1.1	0.05	—	—	10
<i>Highway</i>	0.9	0.15	—	—	10
<i>Paris</i>	1.5	0.05	—	—	30
<i>Tempete</i>	1.5	0.05	—	—	10

point of iteration is important. Thus, we selected the optimal number of iterations for each sequence, as given in Table 3.

3.2 Performance Evaluation

Table 4 shows the PSNR results of several algorithms for 12 test sequences. For this experiment, we implemented LA,

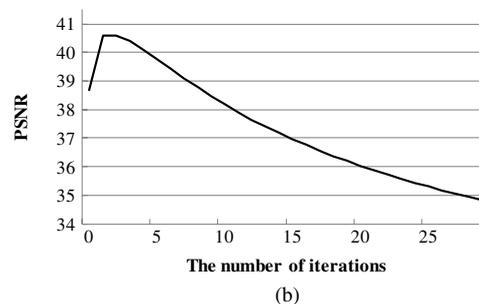
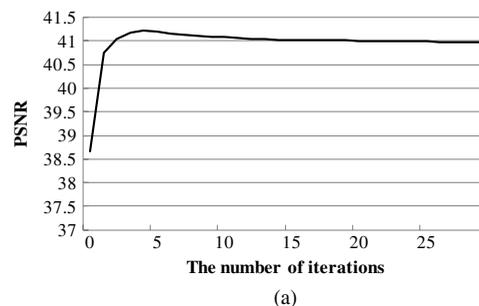


Fig. 8 PSNR values according to the number of iterations for the first shot of the *foreman* sequence. (a) Results with the optimal parameters. (b) Results with nonoptimal parameters.

Table 4 PSNR comparison for different deinterlacing algorithms (dB).

Sequences	Line averaging (LA)	Edge-based LA (ELA)	Ref. 13	Ref. 21	Proposed
<i>Foreman</i>	33.6	34.2	34.4	33.0	38.6
<i>Mobile</i>	25.4	23.5	26.8	24.8	33.9
<i>M&D</i>	39.8	38.8	44.6	41.6	46.2
<i>Stefan</i>	28.3	27.3	29.7	28.5	32.4
<i>Coastguard</i>	28.5	27.8	30.3	31.0	37.1
<i>Table tennis</i>	28.9	27.8	33.5	31.9	35.3
<i>Container</i>	28.9	27.9	34.7	38.3	46.7
<i>Flower</i>	22.6	22.1	22.8	27.9	30.5
<i>Football</i>	34.8	34.4	32.7	31.4	35.7
<i>Highway</i>	32.3	32.4	32.5	31.4	37.0
<i>Paris</i>	27.1	26.8	33.2	33.5	35.3
<i>Tempete</i>	29.2	28.3	29.9	28.9	32.3
Average	30.0	29.3	32.1	31.8	36.8

Table 5 PSNR comparison for different deinterlacing algorithms in the literature (dB).

Sequences	Ref. 9	Ref. 14	Ref. 17	Ref. 18	Ref. 22	Proposed
<i>Foreman</i>	N/A	33.8	38.1	32.9	37.3	38.6
<i>Mobile</i>	30.3	27.7	33.3	N/A	31.5	33.9
<i>M&D</i>	46.2	45.0	45.8	N/A	N/A	46.2
<i>Stefan</i>	28.0	28.8	30.5	26.5	31.3	32.4
<i>Coastguard</i>	32.9	N/A	35.9	30.9	N/A	37.1
<i>Table tennis</i>	34.0	37.0	36.4	28.2	N/A	35.3
<i>Container</i>	39.6	45.5	45.3	44.2	N/A	46.7
<i>Flower garden</i>	N/A	N/A	N/A	N/A	N/A	30.5
<i>Football</i>	N/A	N/A	N/A	N/A	N/A	35.7
<i>Highway</i>	N/A	N/A	N/A	N/A	N/A	37.0
<i>Paris</i>	N/A	N/A	N/A	N/A	N/A	35.3
<i>Tempete</i>	N/A	N/A	N/A	N/A	N/A	32.3

Note: Bold values indicate the maximum PSNR for each sequence.

ELA, Chang’s algorithm,¹³ and Mohammadi’s algorithm²¹ personally and verified them. We can see that the proposed algorithm provides 5 dB higher PSNR than Chang’s¹³ and Mohammadi’s algorithms²¹ on average. Especially, for *container* sequence, the proposed algorithm goes beyond Chang’s algorithm¹³ up to 12 dB.

In addition, Table 5 provides PSNR comparison results for several recent methods in the literature. In Table 5, the numerical values of Refs. 9, 14, 17, 18, and 22 were directly extracted from those papers. Hence, a few values are not available in the table. Note that the proposed algorithm provides on average 0.8 dB higher PSNR than the cutting-edge algorithm, i.e., Fan’s algorithm.¹⁷ For the *Stefan* sequence, in

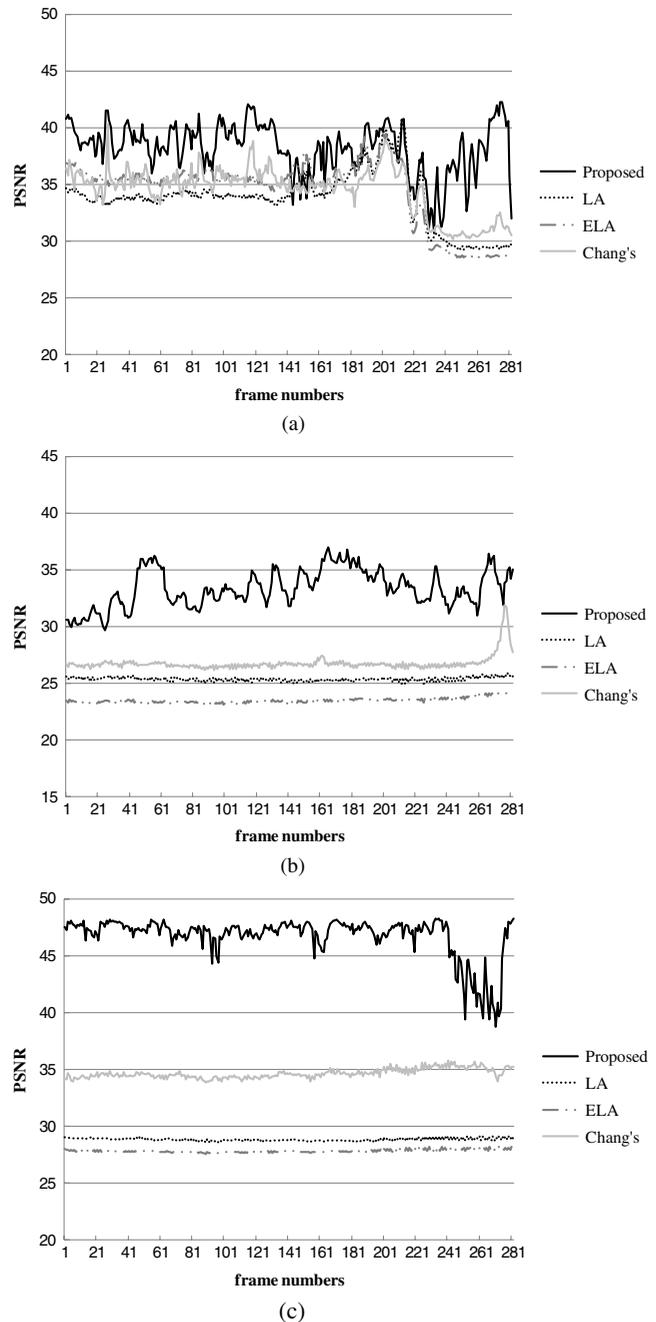


Fig. 9 PSNR performance according to the frame number. (a) *Foreman*. (b) *Mobile*. (c) *Container*.

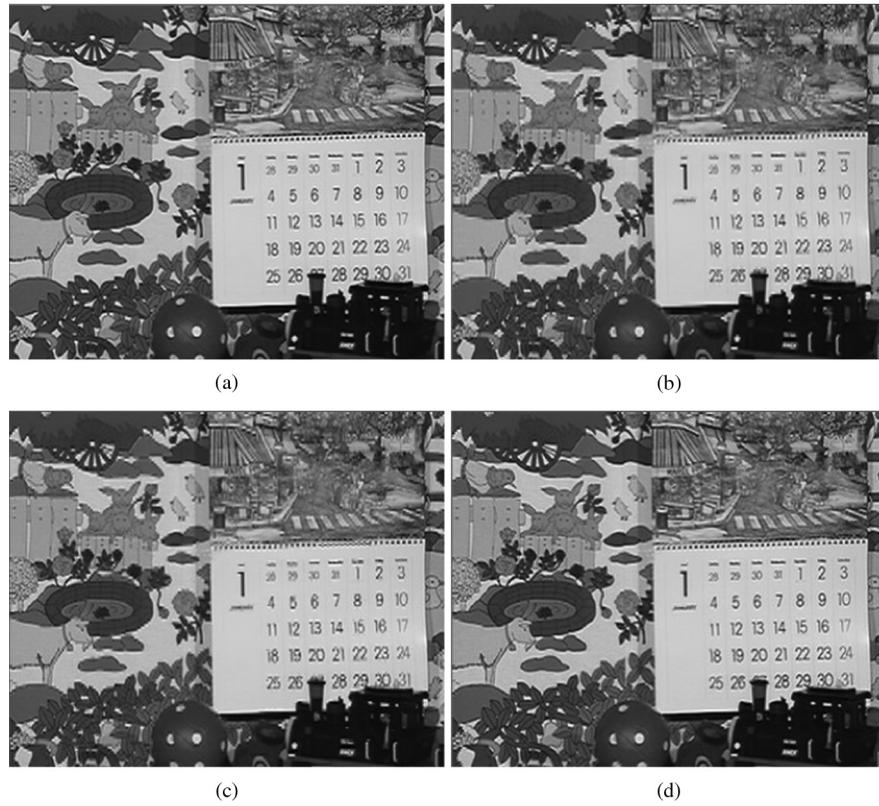


Fig. 10 The deinterlaced frames for the *mobile* sequence. (a) Original. (b) Line averaging (LA). (c) Chang's method.¹³ (d) Proposed method.

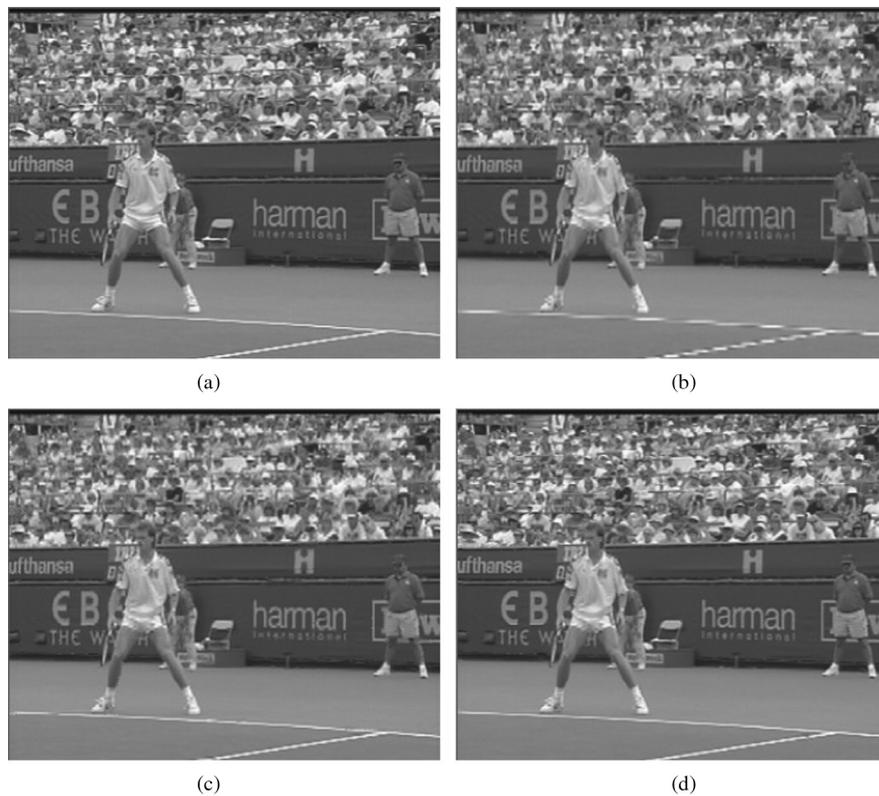


Fig. 11 The deinterlaced frames for the *Stefan* sequence. (a) Original. (b) LA. (c) Chang's method.¹³ (d) Proposed method.

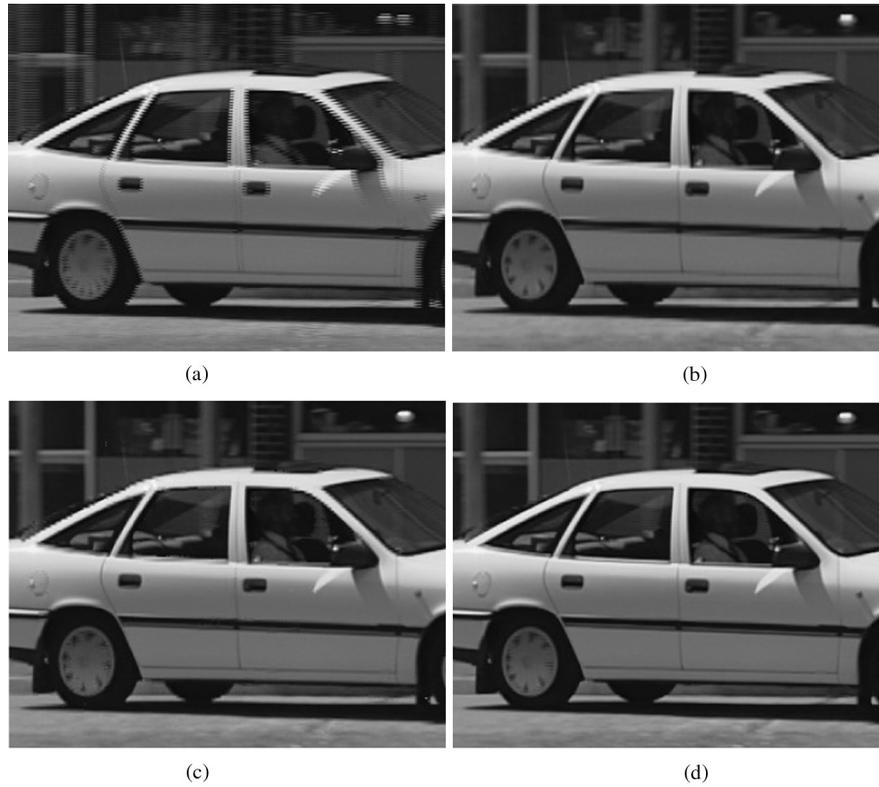


Fig. 12 The results for a cropped region of the *car* sequence. (a) Input interlaced frame. (b) LA. (c) Chang's method.¹³ (d) Proposed method.

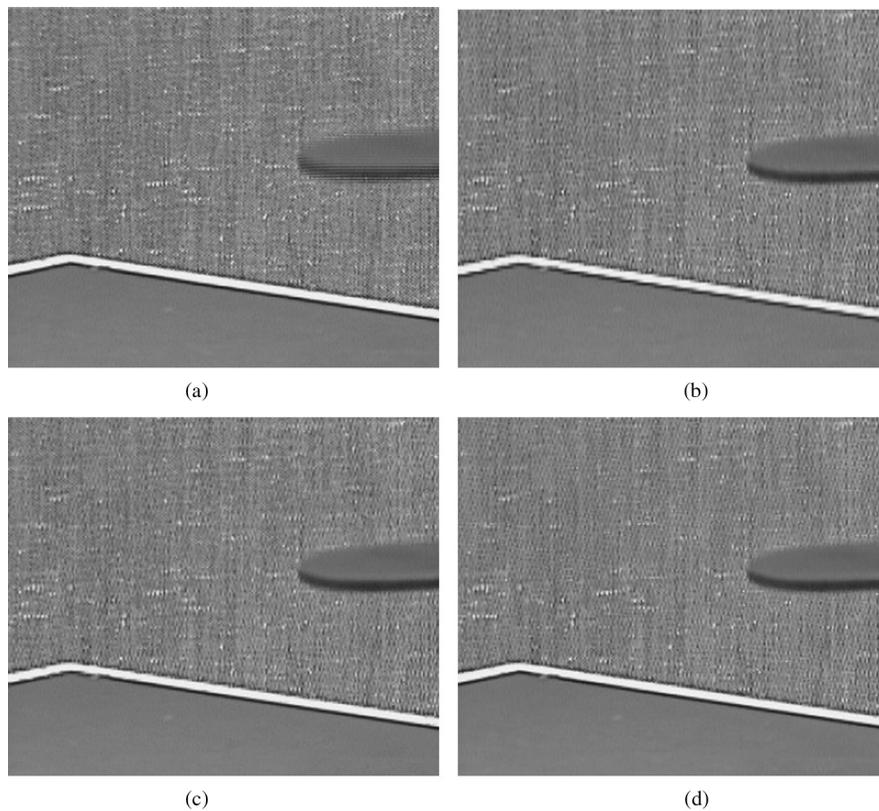


Fig. 13 The results for a cropped region of the *table tennis* sequence. (a) Input interlaced frame. (b) LA. (c) Chang's method.¹³ (d) Proposed method.

Table 6 CPU running times for various algorithms (s).

LA	ELA	Ref. 13	Proposed (fast full search)	Proposed (enhanced predictive zonal search)
0.005	0.09	9.6	34.5	17.4

particular, we achieve a significant PSNR improvement of 2 dB over Fan's algorithm. We also find from the available PSNR values in Table 5 that the proposed algorithm is superior to Trocan's algorithm.²² As a result, the proposed algorithm outperforms the previous works in terms of objective visual quality of PSNR. In addition, Fig. 9 shows the PSNR values according to the frame number. We can state that the proposed algorithm consistently provides higher PSNR values than the previous works.

In Figs. 10 and 11, the deinterlaced results for *mobile* and *Stefan* sequences are compared. Note that the proposed algorithm produces a deinterlaced result close to its original frame, while Chang's algorithm and LA suffer from visually annoying artifacts, such as jaggling and blur. For the *mobile* sequence (see Fig. 10), we find that the proposed algorithm outperforms the others. Especially observing the numbers in the calendar, the proposed algorithm shows almost the same visual quality as the original frame. From Fig. 11, we can find a perfect court line generated by the proposed temporal deinterlacing. Note that the competitors show severe jaggling artifacts and line-crawling in the near horizontal edges.

In addition, Figs. 12 and 13 show the deinterlaced results for real interlaced video sequences. We adopted two well-known interlaced sequences ($720 \times 480i$), *car* and *table tennis*. Even for real interlaced video sequences, the proposed algorithm still provides outstanding visual quality without any artifacts in comparison with the existing methods. For example, LA and Chang's algorithm show jaggling and feathering artifacts [see Figs. 12(b) and 12(c)], but the proposed algorithm does not cause such artifacts, as shown in Fig. 12(d).

In addition, we measured the execution times of several algorithms. This experiment was executed on a quad-core CPU at 2.66 GHz with 3 GB DDR2 DRAM. Since motion estimation occupies most of the complexity of MC-based deinterlacing algorithm, we adopted a fast full search²⁵ and a famous fast search algorithm, i.e., enhanced predictive zonal search (EPZS)²⁶ for fast motion estimation in this experiment. Table 6 compares the proposed algorithm with LA, ELA, and Ref. 13 in terms of the CPU running time. Note that each numerical value indicates the average for the first 10 fields of *foreman* sequence, and all the algorithms were implemented in MATLAB®. We could find that in comparison with fast full search, the EPZS cuts the CPU running time of the proposed algorithm in half with an acceptable PSNR drop of 1 dB on average. The proposed algorithm using EZPS still provides 1.8 times longer execution time than Ref. 13. However, if we employ several optimization skills additionally, we can significantly reduce the computational cost of the proposed algorithm.

4 Concluding Remarks

This paper presents a robust temporal deinterlacing algorithm based on an MAP estimator. First, registration using

an advanced STFS algorithm is performed between the current field and its neighboring fields. Second, the progressive frame corresponding to the current field is found via an L_1 -norm-based MAP estimator based on the predicted interfield MV information. Third, a mode decision module determines whether the result from the temporal deinterlacing is acceptable or not. Finally, edge-directional interpolation is applied to the pixels whose MVs are not reliable, instead of the aforementioned temporal deinterlacing. Experimental results show that the proposed algorithm yields at maximum 2 dB higher PSNR than the cutting-edge deinterlacing algorithm,¹⁷ while providing better visual quality.

Acknowledgements

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012000446), and was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under IT/SW Creative research program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013-H0502-13-1010), and supported by INHA UNIVERSITY Research Grant.

References

- G. de Haan and E. B. Bellers, "Deinterlacing: an overview," *Proc. IEEE* **86**(9), 1839–1857 (1998).
- M. K. Park et al., "New edge dependent de-interlacing algorithm based on horizontal edge pattern," *IEEE Trans. Consum. Electron.* **49**(4), 1508–1512 (2003).
- M. K. Park et al., "Edge-dependent interpolation-based de-interlacing using edge patterns," *J. Electron. Imaging* **15**(4), 043003 (2006).
- J. Wang, G. Jeon, and J. Jeong, "Deinterlacing algorithm with an advanced non-local means filter," *Opt. Eng.* **51**(4), 047009 (2012).
- H. Y. Lee et al., "Adaptive scan rate up-conversion system based on human visual characteristics," *IEEE Trans. Consum. Electron.* **46**(4), 999–1006 (2000).
- J. Lee and C. Lee, "Deinterlacing with reduced flickering," *Opt. Eng.* **51**(6), 067401 (2012).
- Y. T. Kim, "De-interlacing algorithm based on sparse wide vector correlations," *Proc. SPIE* **2727**, 89–99 (1996).
- W. Kim, S. Jin, and J. Jeong, "Novel intra deinterlacing algorithm using content adaptive interpolation," *IEEE Trans. Consum. Electron.* **53**(3), 1036–1043 (2007).
- K. Lee, J. Lee, and C. Lee, "Deinterlacing with motion adaptive vertical temporal filtering," *IEEE Trans. Consum. Electron.* **55**(2), 636–643 (2009).
- S. F. Lin, Y. L. Chang, and L. G. Chen, "Motion adaptive interpolation with horizontal motion detection for deinterlacing," *IEEE Trans. Consum. Electron.* **49**(4), 1256–1264 (2003).
- S. F. Lin, Y. L. Chang, and L. G. Chen, "Motion adaptive interpolation with morphological operation and 3:2 pull-downed recovery for deinterlacing," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, pp. 661–664, IEEE (2002).
- H. Choi and C. Lee, "Motion adaptive deinterlacing with modular neural networks," *IEEE Trans. Circ. Syst. Video Technol.* **21**(6), 844–849 (2011).
- Y. L. Chang et al., "Video de-interlacing by adaptive 4-field global/local motion compensated approach," *IEEE Trans. Circ. Syst. Video Technol.* **15**(12), 1569–1582 (2005).
- Y. R. Chen and S. C. Tai, "True motion-compensated de-interlacing algorithm," *IEEE Trans. Circ. Syst. Video Technol.* **19**(10), 1489–1498 (2009).
- J. Chang et al., "Adaptive arbitration of intra-field and motion compensation methods for de-interlacing," *IEEE Trans. Circ. Syst. Video Technol.* **19**(8), 1214–1220 (2009).
- Q. Huang et al., "Deinterlacing using hierarchical motion analysis," *IEEE Trans. Circ. Syst. Video Technol.* **20**(5), 673–686 (2010).
- Y. C. Fan and C. H. Chung, "De-interlacing algorithm using spatial-temporal correlation-assisted motion estimation," *IEEE Trans. Circ. Syst. Video Technol.* **19**(7), 932–944 (2009).
- S. Yang et al., "Motion compensation assisted motion adaptive interlaced-to-progressive conversion," *IEEE Trans. Circ. Syst. Video Technol.* **14**(9), 1138–1148 (2004).

19. J. Dong and K. N. Ngan, "Real-time de-interlacing for H.264-coded HD videos," *IEEE Trans. Circ. Syst. Video Technol.* **20**(8), 1144–1149 (2010).
20. H. Lee, T. H. Lee, and B. C. Song, "Spatio-temporal de-interlacing based on maximum likelihood estimation," in *IEEE Visual Communications and Image Processing*, pp. 1–4, IEEE (2011).
21. H. M. Mohammadi, Y. Savaria, and J. M. P. Langlois, "Enhanced motion compensated deinterlacing algorithm," *IET Image Process.* **6**(8), 1041–1048 (2012).
22. M. Trocan, B. Mikovicova, and D. Zhanguzin, "An adaptive motion-compensated approach for video deinterlacing," *Multimed. Tools Appl.* **61**(3), 819–837 (2012).
23. S. Farsiu et al., "Fast and robust multiframe super-resolution," *IEEE Trans. Image Process.* **13**(10), 1327–1344 (2004).
24. D. Capel and A. Zisserman, "Super-resolution from multiple views using learnt image models," in *Proc. of 2001 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. II-627–II-634, IEEE (2001).
25. M. Brunig and W. Niehsen, "Fast full-search block matching," *IEEE Trans. Circ. Syst. Video Technol.* **11**(2), 241–247 (2001).
26. A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Proc. Visual Communications and Image Processing*, pp. 1069–1079, IEEE (2002).

Ho-Taek Lee received his BS and MS degrees in electronic engineering from Inha University, Incheon, Korea, in 2010 and 2012,

respectively. Currently, he is working for LG Electronics in Seoul, Republic of Korea. His research interests include image processing and video coding.

Dong-bok Lee received his BS and MS degrees in electronic engineering from Inha University, Incheon, Korea, in 2011 and 2013, respectively. Currently, he is pursuing his PhD degree in Inha University. His research interests include image processing and video coding.

Byungju Lee received his BS degree in electronic engineering from Inha University, Incheon, Korea, in 2012. Currently, he is pursuing his MS degree in Inha University. His research interests include image processing and video coding.

Byung Cheol Song received his BS, MS, and PhD degrees in electrical engineering from the KAIST, Daejeon, Korea, in 1994, 1996, and 2001, respectively. From 2001 to 2008, he was a senior engineer at Digital Media R&D Center, Samsung Electronics Co., Ltd., Suwon, Korea. In March 2008, he joined the School of Electronic Engineering, Inha University, Incheon, Korea, and currently is an associate professor. His research interests are in the general areas of image/video processing.