Local surface model–based deinterlacing algorithm

Sang-Jun Park
Jechang Jeong
Hanyang University
Department of Electronics and Computer Engineering
607 R&D Building
17 Haengdang-dong
Seongdong-gu, Seoul 133-791
Republic of Korea
E-mail: jjeong@ece.hanyang.ac.kr

Abstract. We present an efficient deinterlacing method via mathematical modeling of the neighbor pixels in the local region. The local surface model is designed using the quadratic equation having two-dimensional coordinate variables. Unlike conventional deinterlacing methods, the proposed method avoids using directional difference measures, resulting in reduced limitation on the number of considering edge directions. By modeling the local surface, it is easier to derive the true characteristic of the local region in a natural image than utilizing the directional difference measure. In order to decide the optimal coefficients of the surface model, the neighbor pixels around the current pixel to be interpolated are utilized. Once the coefficients are determined, the surface model estimates the pixel intensity of the current pixel to be interpolated. Simulation results show that the proposed surface model-based deinterlacing method minimizes the interpolation error. Compared to the traditional deinterlacing methods and Wiener filter-based interpolation method, the proposed method improves the subjective quality of the interpolated edges while maintaining a higher peak signal-to-noise–ratio level. *© 2011 Society of Photo-Optical Instrumentation Engineers (SPIE).* [DOI: 10.1117/1.3533027]

Subject terms: surface model; deinterlacing; intrafield.

Paper 100713R received Sep. 1, 2009; revised manuscript received Nov. 10, 2009; accepted for publication Nov. 23, 2009; published online Jan. 27, 2011.

1 Introduction

In video broadcasting, general TV systems, such as PAL, SECAM, and NTSC, currently adopt an interlaced format to halve the video transfer bandwidth. This standard frame rate is sufficient for distributing slow motion; however, it tends to introduce flickering for objects that have high horizontal frequencies, interline flicker, jaggedness, and line crawling.[1,2] In addition, the growth of modern display systems such as high-definition television (HDTV), PC monitors, liquid-crystal displays, and plasma display panels requires that the whole image be displayed at once.

For the aforementioned reasons, deinterlacing algorithms are required in order to enhance spatial resolution in the vertical direction. Research in the past couple of years has focused on deinterlacing video signals, with studies ranging from single-field spatial algorithms[3–16] to the more sophisticated interfield temporal algorithms, including motion-adaptive algorithms and motion-compensation algorithms.[17–19] Interfield temporal algorithms perform better than single-field spatial algorithms; however, they usually require sizable computational complexity for computing motion, which is not feasible for real applications. Moreover, poor performance can be seen when the motion information is unreliable.

Conversely, single-field spatial algorithms, such as the line-averaging (LA) algorithm,[3] edge-based line-average (ELA) algorithm,[4] efficient ELA (EELA) algorithm,[5] modified ELA (MELA),[7] fuzzy detection of edge direction for video line doubling (FDED),[12] new edge-dependent deinterlacing (NEDD) algorithm,[9] direction-oriented interpolation (DOI) algorithm,[8] fine directional deinterlacing (FDD) algorithm,[13] low-complexity interpolation method for deinterlacing (LCID),[10] deinterlacing using locally adaptive-thresholded binary image (LABI),[11] edge map-based deinterlacing (EMD) method,[14] deinterlaced algorithm based on sparse wide-vector correlations (DSWVC),[6] and fine edge-preserving deinterlacing algorithm for progressive display (EPD),[15] are suitable for most real-time applications due to their clarity, low cost, and easy implementation in hardware. Among them, LA,[3] a simple method that interpolates the pixel using an average value of the upper and lower pixels, has been generally used due to its clarity with small complexity while it exhibits no motion artifacts. However, the vertical resolution of the input image is halved before the image is interpolated. Therefore, the details in the progressive image are reduced. To alleviate this issue, the ELA (Ref. 4) method has been proposed. ELA provides good performance and can eliminate the blurring effect of LA. Moreover, it gives both sharp and straight edges. However, because the ELA only considers three directions (135, 90, and 45 deg), interpolation errors often become larger in the high-frequency areas. Instead of utilizing the difference between two pixels as a directional correlation, DSWVC uses the difference of vectors (neighbor pixel set). The DSWVC (Ref. 6) uses three different methods, which are the LA mode, narrow-vector correlation mode, and sparse wide-vector correlation mode, for different regions. However, the DSWVC yields higher mean-squared error (MSE) than the LA. DOI (Ref. 8) introduced a spatial direction vector to obtain finer resolution and higher accuracy of the edge direction. However, DOI requires a large amount of computations due to its large search range. FDED (Ref. 12) utilized the fuzzy set theory to detect the prevailing edge direction among the five directions (0, ± 45, and ± 60 deg). FDD (Ref. 13) introduced the modified Sobel masks to detect the edge tendency of the pixel being interpolated and performs the interpolation along the seven directions. EMD (Ref. 14) also predicts the edge direction of the current pixel to be interpolated with the original Sobel mask and performs the interpolation along the detected direction with the candidate deinterlaced pixels, which are the

0091-3286/2011/$25.00 © 2011 SPIE

average values between any two pixels in the corresponding edge direction. EPD (Ref. 15) first selects the dominant edge by calculating the gradient of pixel block subsets, and then different interpolation methods are applied to four different regions. However, FDD, EMD, and EPD sometimes yield incorrect edge direction because they only consider horizontal and vertical gradients to obtain the local edge direction. The LABI (Ref. 11) uses a binary image to extract the possible edge patterns, and the interpolation is operated according to the edge pattern. However, the LABI provides noticeable improvements on the specific regions where a horizontal edge exists. LCID (Ref. 10) is a simple deinterlacing method considering only four directional differences, but yields blurring due to excessive averaging along the incorrect edge direction. Although these conventional methods perform interpolation along the edge direction and are widely used for their simplicity and ease of implementation, they tend to yield noise due to the limitation of the candidate edge directions, as verified in Ref. 13, and incorrect edge direction detection using edge detectors based on horizontal and vertical gradients.

To alleviate this issue, Park *et al.* presented a covariance-based adaptive deinterlacing (CAD) algorithm for intrafield interpolation.[16] They first employed the geometric duality[20] concept and Wiener filtering theory[21] to estimate the model parameters. The geometric duality concept was used to estimate the high-resolution covariance from its low-resolution counterpart with a qualitative model characterizing the relationship between the covariance and the resolution. As long as the correspondence between the high- and low-resolution covariances are established, the optimal minimum MSE (MMSE) linear interpolation coefficients can be derived by the classical Wiener filter. However, the major drawback of CAD is its costly computational complexity.

In order to alleviate the computational complexity, we present a mathematical model for the local region in a natural image. Because the proposed method utilizes fewer neighbor pixels than CAD, the true characteristics of the local region can be easily derived with lower complexity than CAD. Moreover, the operation for calculating the inverse matrix is not required for the proposed method due to its coordinate-based surface model. The proposed method is specific for a sequence having high resolution because the proposed method deals with more neighbor pixels than the conventional deinterlacing methods, which use a directional pixel difference measure within the small (restricted) region. Also, it is worth noting that identifying a deinterlacing method based on spatial domain is still an active area of research. This paper is structured as follows. In Sec. 2, the existing intrafield deinterlacing methods, including the edge-based line interpolation approach and Wiener filtering approach, are described. In Sec. 3, the proposed mathematical model and its implementation are explained. In Sec. 4, experimental results and performance analyses are provided to show the feasibility of the proposed approach. Finally, conclusions are made in Sec. 5.

2 Previous Deinterlacing Algorithms

2.1 *Edge-Based Deinterlacing Methods*

2.1.1 *Modified edge-based line averaging*

MELA (Ref. 7) is a modified version of the ELA.[4] The ELA interpolates the pixel in the direction whose pixel difference is the smallest among the three pixel differences. The pixel differences in the three different directions are obtained as follows:

$$\begin{aligned} C_{-1} &= |u_1 - d_3|\,, \\ C_0 &= |u_2 - d_2|\,, \\ C_1 &= |u_3 - d_1|\,. \end{aligned} \tag{1}$$

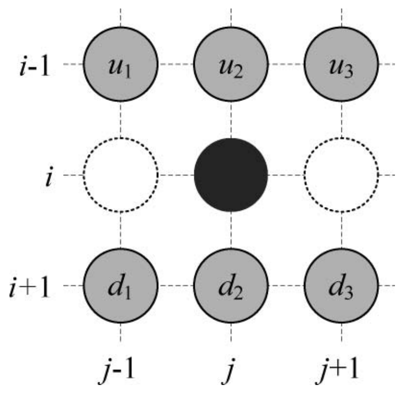


Fig. 1 Six neighbor pixels used for determining the local edge direction.

where u_1, u_2, u_3, d_1, d_2, and d_3 denote the intensities of six neighbor pixels shown in Fig. 1. In Fig. 1, i and j denote the indices for the lines and the columns, respectively. MELA additionally considers three directional spatial correlations that cover vertical (90-deg), diagonal (117-deg), and antidiagonal (63-deg) directions, and these correlations are calculated as follows:

$$\begin{cases} V = (|u_1 - d_1| + |u_2 - d_2| + |u_3 - d_3|)/3, \\ P = (|u_1 - d_2| + |u_2 - d_3|)/2, \\ Q = (|u_2 - d_1| + |u_3 - d_2|)/2. \end{cases} \tag{2}$$

Then, according to the local edge direction, an appropriate interpolation is performed as follows:

$$x(i, j) = \begin{cases} \dfrac{u_1 + u_2 + d_2 + d_3}{4}, \\ \quad \text{if } \{\min(P, Q, V) = P\} \wedge \{C_{-1} < C_0\} \\ \dfrac{u_2 + u_3 + d_1 + d_2}{4}, \\ \quad \text{if } \{\min(P, Q, V) = Q\} \wedge \{C_1 < C_0\} \\ \dfrac{u_2 + d_2}{2}, \quad \text{otherwise.} \end{cases} \tag{3}$$

The operator $\wedge$ denotes the AND operation. By interpolating the lost pixels along theses directional spatial correlations, MELA becomes more robust and efficient than other methods, including LA (Ref. 3), ELA, and EELA (Ref. 5). Additionally, MELA yields an even better peak signal-to-noise–ratio (PSNR) result with almost the same computational time as that of ELA.

2.1.2 *Low-complexity interpolation*

LCID (Ref. 10) uses four simpler directional differences (in the direction of diagonal, antidiagonal, vertical, and horizontal) than MELA, and they are denoted as D_{d1}, D_{d2}, D_{v}, and D_{h}. The four directional differences are calculated as follows:

$$\begin{cases} D_{\text{d1}} = |u_1 - d_2| + |u_2 - d_3|\,, \\ D_{\text{d2}} = |u_2 - d_1| + |u_3 - d_2|\,, \\ D_{\text{v}} = 2 \times |u_2 - d_2|\,, \\ D_{\text{h}} = |u_1 - u_2| + |d_1 - d_2|\,. \end{cases} \tag{4}$$

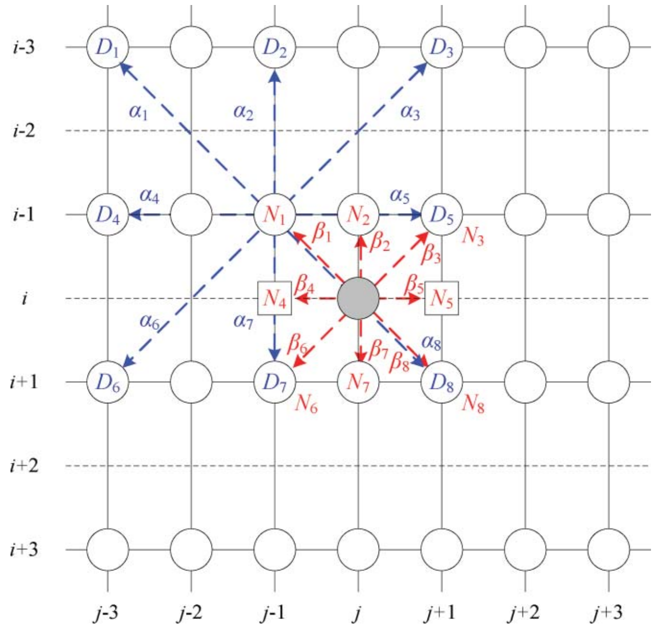


Fig. 2 Geometric duality for the CAD.

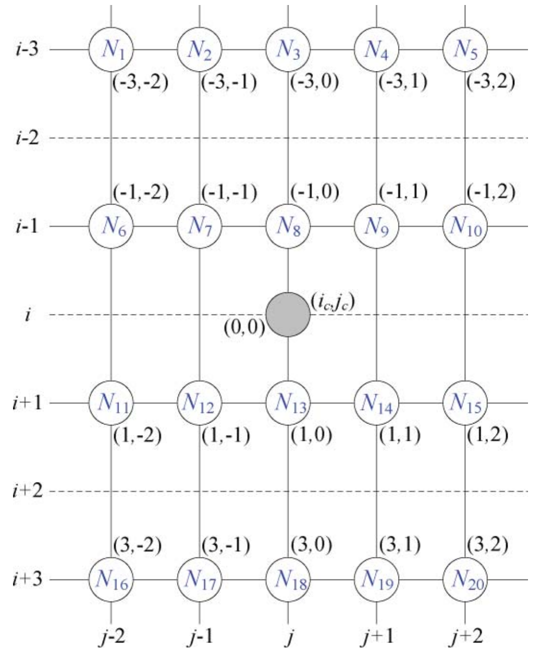


Fig. 3 Indices (i_L, j_L) of 20 neighbor pixels and the current pixel to be interpolated within the local region.

Then, similar to MELA, the pixel is interpolated in the direction having minimum directional difference as follows:

$$x(i,j)=\begin{cases} x(i,j-1), & \text{if} \quad D_h=0 \\ \dfrac{u_1+u_2+d_2+d_3}{4}, & \text{if} \quad \min(D_{d1},D_{d2},D_v)=D_{d1} \\ \dfrac{u_2+u_3+d_1+d_2}{4}, & \text{if} \quad \min(D_{d1},D_{d2},D_v)=D_{d2} \\ \dfrac{u_2+d_2}{2}, & \text{otherwise.} \end{cases} \tag{5}$$

On average, LCID provides lower image quality than MELA because LCID replaces the current pixel to be interpolated to the previously interpolated pixel $[x(i,j-1)]$ when $D_h = 0$. Because it is not certain that the pixel is interpolated correctly, we should avoid using the estimated pixel value.

2.2 *Covariance-Based Adaptive Deinterlacing Method*

Geometric duality refers to the correspondence between the high- and low-resolution covariances, which couples the pair of pixels at different resolutions but along the same orientation.[20] Hence, geometric duality facilitates the estimation of local covariance for images without the necessity of explicitly estimating the edge direction.

A main goal of CAD (Ref. 16) is using low-resolution image samples to estimate the parameter α in Eq. (6). Referring to the spatial relation between the samples in Fig. 2, we can obtain a linear least-squares estimator of the model parameter vector α as follows:

$$\hat{\alpha}=\arg\min_{\alpha}\sum_{m\in S}\left(N_m-\sum_{1\le t\le 8}\alpha_t N^{(8)}_{m\diamond t}\right)^2, \tag{6}$$

where $N^{(8)}_{m\diamond t}$ are the eight connected neighbors of the location m within the local window S in the low-resolution image. Note that the estimates of α in Eq. (6) are made using the low-resolution pixels N_m. Hence, the resulting estimates $\hat{\alpha}$ are the optimal in the least-squares sense under the assumption that the sample covariances are not altered in the local window, which is generally true for natural images. As shown in Fig. 2,

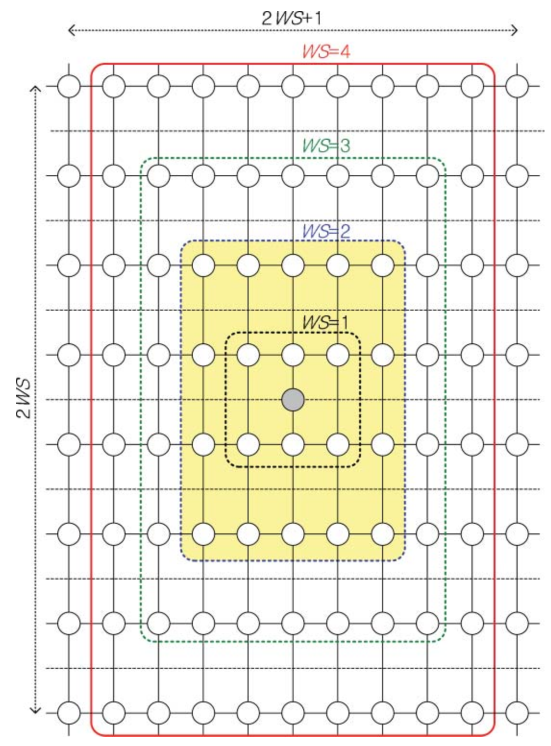
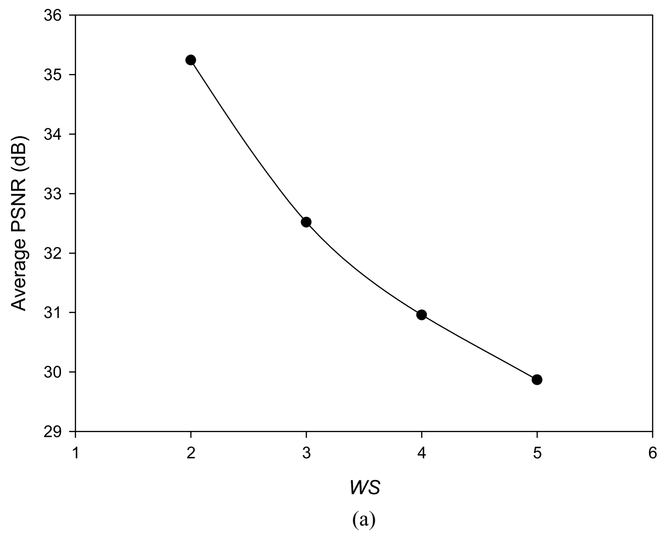


Fig. 4 Various window sizes.



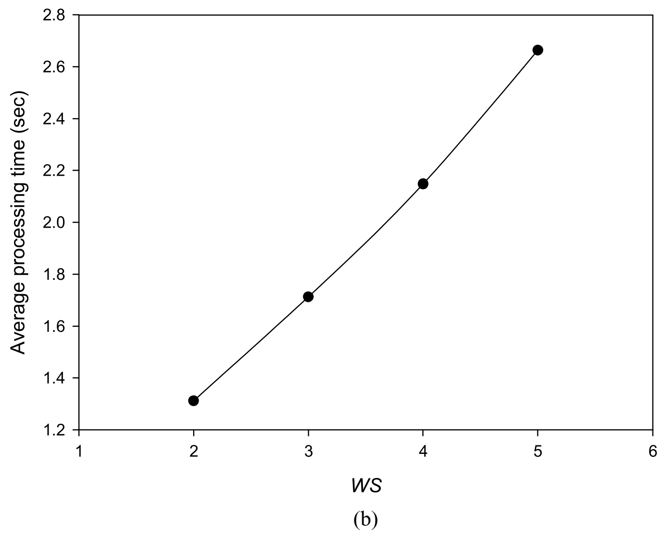


Fig. 5 Simulation results with varying WS. (a) Average PSNR. (b) Average processing time.

CAD assumes that the pixel intensity of N_1 is a weighted sum of the eight neighbor pixels (from D_1 to D_8). The similar assumption for N_1 is extended to the other pixels (data vector) within S. The various sizes of S are shown in Ref. 16 and determined by the parameter of window size, WS. In Ref. 16, the optimal value for the WS is determined as 7. Hence, the optimal weight vector ($\alpha = [\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_8]^T$) for the data vector ($\mathbf{y} = [y_0, y_1, \ldots, y_{2\mathrm{WS}\times(2\mathrm{WS}+1)}]^T$) can be obtained by solving the following:

$$\mathbf{y} = C\alpha, \tag{7}$$

where C is a $[2\mathrm{WS} \times (2\mathrm{WS}+1)] \times 8$ data matrix whose k'th row vector is the eight neighbors of y_k. Because the data matrix C is not square, the weight vector α is derived using the least-squares approximation as follows:

$$\alpha = \left(C^T C\right)^{-1}\left(C^T \mathbf{y}\right). \tag{8}$$

Note that the terms C^TC and $C^T\mathbf{y}$ are the autocorrelation and cross-correlation of the low-resolution samples, respectively. Finally, the pixel can be interpolated using the eighth-order linear interpolation as follows:

$$x(i, j) = \sum_{k=1}^{8} \alpha_k N_k. \tag{9}$$

Note that α is used as an interpolation weight instead of β because we assume that there is geometric duality between the high- and low-resolution covariances so that β is almost the same as α. N_4 and N_5 are the missing pixels in the interlaced (low-resolution) image. It is possible to use previously reconstructed value for N_4, and N_5 can be predicted using conventional deinterlacing methods, such as MELA or the average value of the upper (N_3) and lower (N_8) pixels.

3 Local Surface Model-Based Deinterlacing Method

A mathematical model of the local region in the natural image is proposed. We can estimate the intensity of the current pixel to be interpolated by evaluating the value of the model function at the position of the current pixel. The size of local region to be modeled is fixed as shown in Fig. 3. Let the origin (i_C,j_C) denotes the position of the current pixel to be interpolated. It is well known that there exist spatial

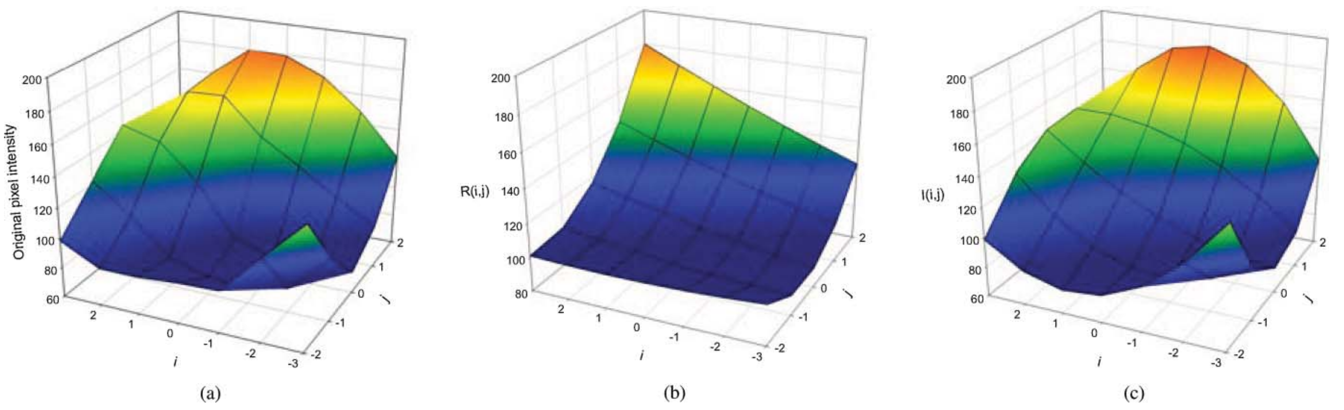


Fig. 6 Comparison of local surfaces. (a) original local surface at (*i,j*) = (220,100) of Finger test image. (b) local surface modeled by *R*(*i,j*). (c) local surface modeled by *I*(*i,j*).

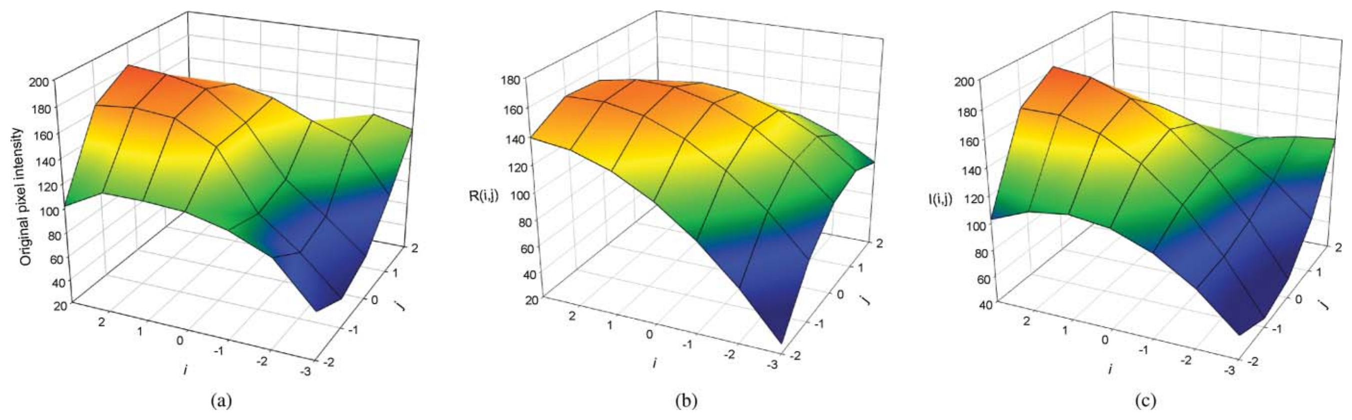


Fig. 7 Comparison of local surfaces. (a) original local surface at $(i,j) = (330,300)$ of Finger test image. (b) local surface modeled by $R(i,j)$. (c) local surface modeled by $I(i,j)$.

correlations within a local region of the natural images.[16,22] Therefore, we can assume that the intensity of the current pixel is dependent on the existing 20 neighbor pixels ($N_1, N_2, N_3, \ldots, N_{20}$). Hence, we model the local region as follows:

$$
\begin{aligned}
I(i,j) &= c_1 i_L^2 j_L^2 + c_2 i_L^2 j_L + c_3 i_L j_L^2 + c_4 i_L j_L + c_5 i_L^2 \\
&\quad + c_6 i_L + c_7 j_L^2 + c_8 j_L + c_9 \\
&\quad \text{where} \quad i_L = i - i_C, \quad j_L = j - j_C.
\end{aligned} \tag{10}
$$

Here, $I(i,j)$ denotes the pixel intensity at i-th line and j-th column in the image. i_L and j_L are position indices within the given local region. The possible coordinate values for the i_L are –3, –1, 1, and 3 and the possible coordinate values for the j_L are –2, –1, 0, 1, and 2. The nine coefficients can be determined using the intensities of the neighbor pixels at 4×5 positions around (i_C, j_C). Hence, the intensities of 20 neighbor pixels determine nine coefficients in Eq. (10). We used 20 neighbor pixels after observing the relationship between the number of neighboring pixels and the interpolation results. Figure 4 shows the various window sizes. WS denotes the parameter of window size. We simulated our method with varying window sizes from WS = 2 to WS = 5 (i.e., from 20 to 110 neighbor pixels). Note that we cannot consider the case of WS = 1, because the number of neighbor pixels (which is 6) is less than the number of coefficients (which is 9) of surface model. Simulation results in terms of average PSNR and average processing time are shown in Fig. 5; 38 different images were used for the simulation. In Fig. 5, it is obvious that the highest average PSNR with the shortest average processing time is obtained when WS = 2. Note that just using many neighbor pixels does not guarantee performance. We used the nearest 20 neighbor pixels that are optimal for modeling a local surface.

Let us define 20 neighbor pixels and 9 coefficients as a column vector **N** and **c**, respectively. Then, substituting the intensities of 20 neighbor pixels sequentially from the upper left position [i.e., (–3,–2)] to the lower right position [i.e., (3,2)], the following equation is obtained:

$$
\mathbf{N} = \mathbf{Mc} \tag{11}
$$

where the neighbor pixel vector **N**, coordinate polynomial term matrix **M**, and coefficient vector **c** are given as follows:

$$
\begin{bmatrix} N_1 \\ N_2 \\ N_3 \\ N_4 \\ N_5 \\ N_6 \\ N_7 \\ N_8 \\ N_9 \\ N_{10} \\ N_{11} \\ N_{12} \\ N_{13} \\ N_{14} \\ N_{15} \\ N_{16} \\ N_{17} \\ N_{18} \\ N_{19} \\ N_{20} \end{bmatrix}
=
\begin{bmatrix}
36 & -18 & -12 & 6 & 9 & -3 & 4 & -2 & 1 \\
9 & -9 & -3 & 3 & 9 & -3 & 1 & -1 & 1 \\
0 & 0 & 0 & 0 & 9 & -3 & 0 & 0 & 1 \\
9 & 9 & -3 & -3 & 9 & -3 & 1 & 1 & 1 \\
36 & 18 & -12 & -6 & 9 & -3 & 4 & 2 & 1 \\
4 & -2 & -4 & 2 & 1 & -1 & 4 & -2 & 1 \\
1 & -1 & -1 & 1 & 1 & -1 & 1 & -1 & 1 \\
0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1 \\
1 & 1 & -1 & -1 & 1 & -1 & 1 & 1 & 1 \\
4 & 2 & -4 & -2 & 1 & -1 & 4 & 2 & 1 \\
4 & -2 & 4 & -2 & 1 & 1 & 4 & -2 & 1 \\
1 & -1 & 1 & -1 & 1 & 1 & 1 & -1 & 1 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
4 & 2 & 4 & 2 & 1 & 1 & 4 & 2 & 1 \\
36 & -18 & 12 & -6 & 9 & 3 & 4 & -2 & 1 \\
9 & -9 & 3 & -3 & 9 & 3 & 1 & -1 & 1 \\
0 & 0 & 0 & 0 & 9 & 3 & 0 & 0 & 1 \\
9 & 9 & 3 & 3 & 9 & 3 & 1 & 1 & 1 \\
36 & 18 & 12 & 6 & 9 & 3 & 4 & 2 & 1
\end{bmatrix}
\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \\ c_8 \\ c_9 \end{bmatrix}. \tag{12}
$$

Obviously, the matrix **M** is always fixed and common for every pixel to be interpolated. Every component within the matrix **M** is always fixed because it is the polynomial term of coordinate within the fixed window, shown in Fig. 3. For example, in the first row of **M**, (–3,–2) is inserted to variables i_L and j_L of Eq. (10). In the last row of **M**, (3,2) is inserted to variables i_L and j_L of Eq. (10). That is why **M** has 20 rows and 9 columns. Because the matrix **M** is tall and thin, there

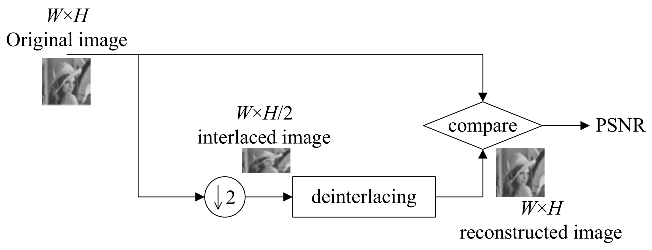


Fig. 8 Objective performance measurement method.

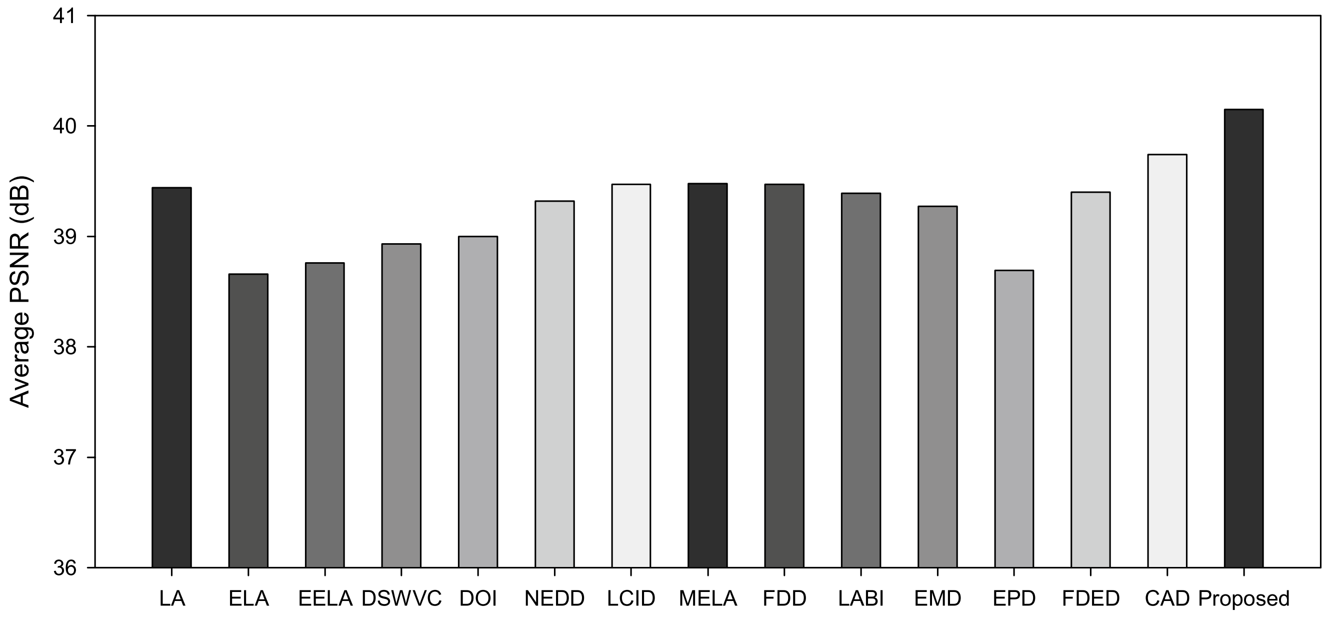


Fig. 9 Comparison of average PSNR results for different intrafield deinterlacing methods.

should be a unique solution if the matrix **M** has full column rank. This overdetermined equation is a typical least-squares problem and can be solved via following equation:

$$\mathbf{c} = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{N}. \tag{13}$$

Figure 6 depicts the examples of the local surface at pixel position (220, 100) in Finger image. Figures 6(a)–6(c) show original local surface, local surface modeled by reduced form [$R(i,j)$] of Eq. (10), and local surface modeled by $I(i,j)$ in Eq. (10), respectively. [Note that $R(i,j) = c_1 i_L^2 + c_2 i_L j_L + c_3 j_L^2 + c_4 i_L + c_5 j_L + c_6$. $R(i,j)$ is a simplified model obtained by eliminating higher-order terms of $I(i,j)$, such as $i_L^2 j_L^2$, $i_L^2 j_L$, and $i_L j_L^2$.] It can be confirmed that a local surface modeled by $I(i,j)$ is more similar to the original local surface than that of $R(i,j)$. Note that the pixel to be interpolated is located at (0, 0). Hence, we can obtain the pixel intensity by substituting the 0 and 0 for i_L and j_L in Eq. (10). In addition, another local surface comparison at different pixel position (330, 300) is shown in Fig. 7.

To further reduce the computational time, the partial term $(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$ of Eq. (13) can be precalculated because **M** is a fixed matrix. Finally, the estimation of the current pixel can be obtained by substituting the 0 and 0 for i_L and j_L. In other words, the ninth coefficient, c_9, can be the optimal intensity for the current pixel. Hence, the c_9 can be directly obtained as follows:

$$c_9 = \text{row9}\left[\left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\right]\mathbf{N} \tag{14}$$

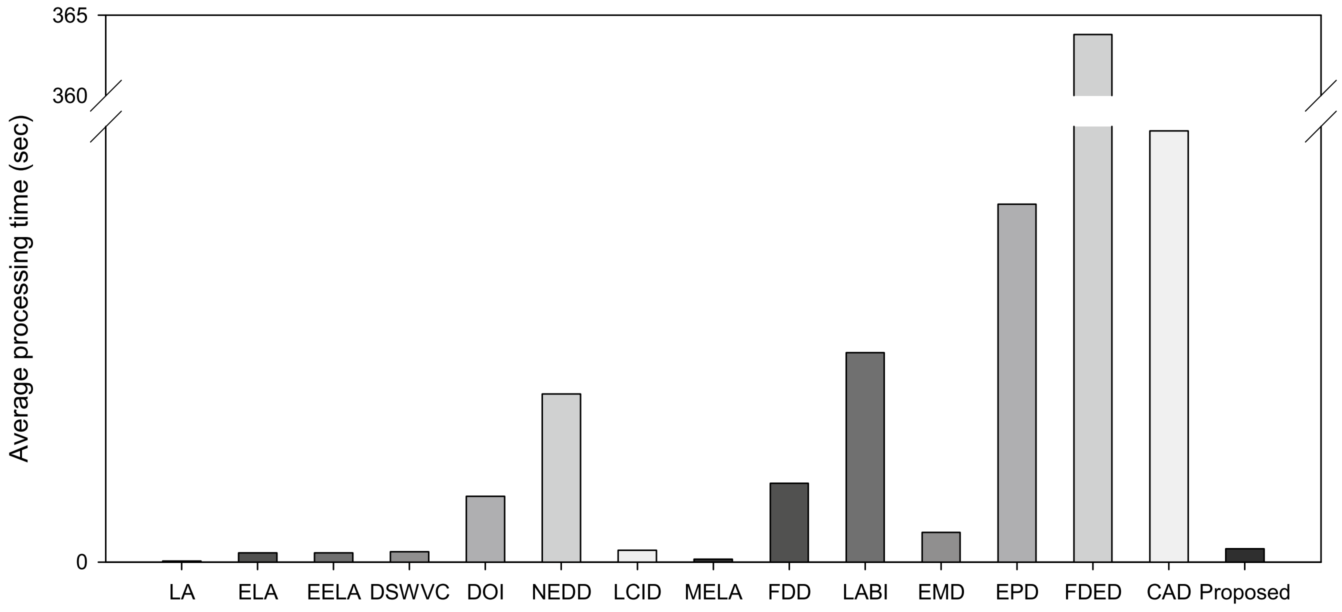


Fig. 10 Comparison of average processing time results for different intra-field deinterlacing methods.

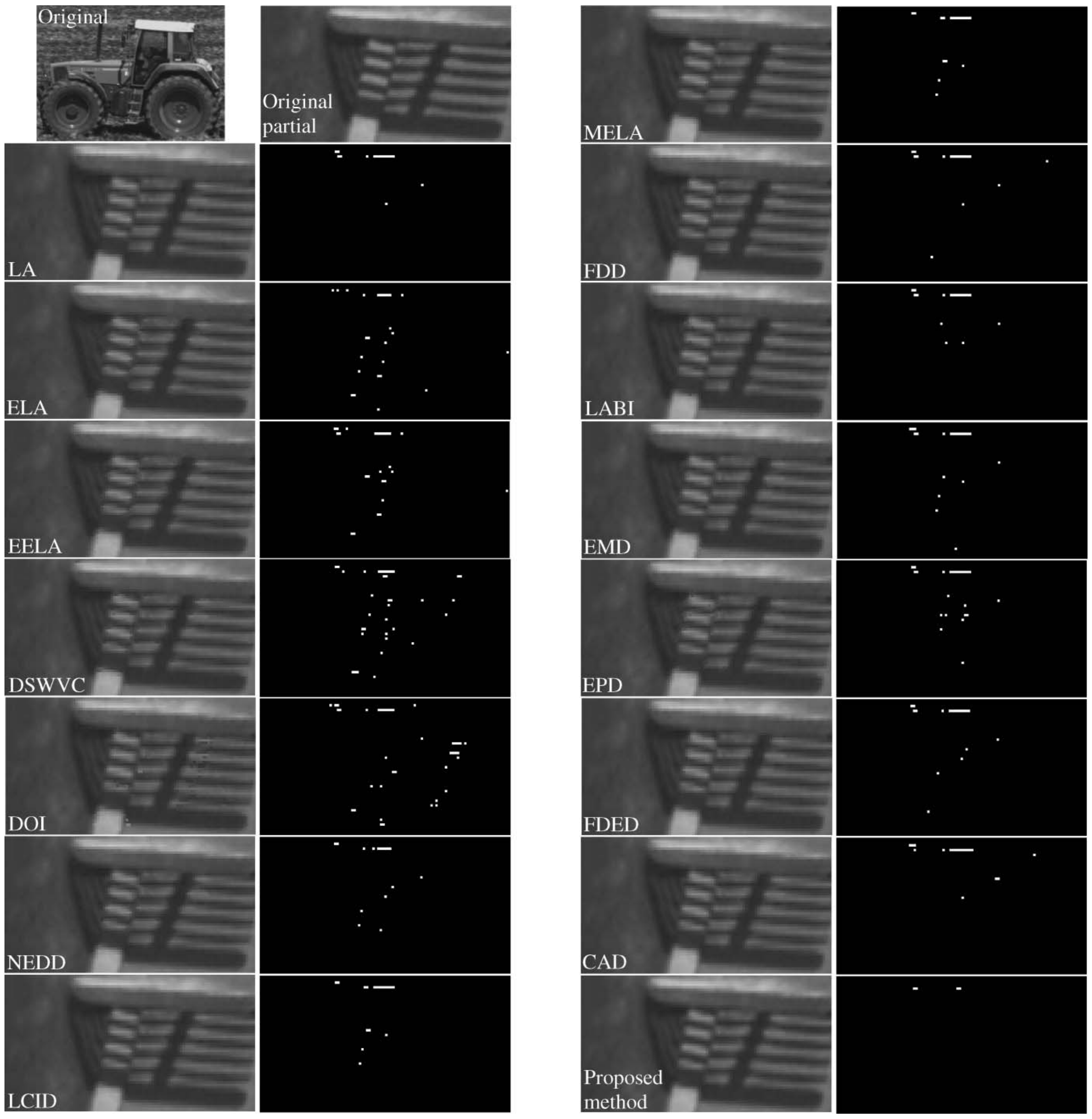


Fig. 11 Enlarged views of the reconstructed images for subjective image quality comparison. First row in left column: Original Tractor image (zoomed region is marked by rectangle) and original partial image. Images with method label are reconstructed images using various algorithms. Images right next to the reconstructed images are absolute difference images between the regions of original and corresponding reconstructed images using various algorithms. The values (absolute pixel difference) less than 20 are displayed as black, and the values greater than 20 are displayed as white. Since the white pixels represent the large interpolation error, a good deinterlacing method should have less number of white pixels within a certain region. It is noticeable that the proposed method yields the least number of white pixels.

where row9$[(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T]$ denotes the ninth row vector of the partial term $(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T$.

4 Performance Evaluation

In this section, a comparison is made of objective and subjective qualities, and computational processing time for the different intrafield deinterlacing methods including the proposed methods. We compare the subjective and objective qualities of LA,[3] ELA,[4] EELA,[5] DSWVC,[6] DOI,[8] NEDD,[9] LCID,[10] MELA,[7] FDD,[13] LABI,[11] EMD,[14] EPD,[15] FDED,[12] CAD,[16] and the proposed method. In addition, two 512 × 512 still images (Finger and Peppers), two 1280 × 720 sequences (Jets and Raven), and five 1920 × 1080 sequences (Bluesky,

Table 1 Comparison of PSNR (measured in decibels) and average processing time (measured in seconds) for eight test images with different intrafield deinterlacing methods.

		Bluesky	Finger	Jets	Kimono	Peppers	Raven	Riverbed	Sunflower	Tractor	Avg.
LA	dB	37.88	31.97	39.11	44.09	33.80	42.13	39.83	47.57	38.60	39.44
	s	0.19	0.03	0.09	0.20	0.03	0.08	0.19	0.20	0.20	0.14
ELA	dB	37.23	28.91	38.93	43.60	34.11	39.93	39.53	47.37	38.28	38.66
	s	2.13	0.28	0.97	2.19	0.28	0.97	2.16	2.17	2.17	1.48
EELA	dB	37.00	29.67	39.00	43.75	33.64	40.57	39.62	47.36	38.21	38.76
	s	2.13	0.27	0.97	2.28	0.27	0.98	2.17	2.13	2.08	1.47
DSWVC	dB	37.11	29.52	39.06	43.98	33.67	41.34	39.77	47.41	38.49	38.93
	s	4.72	2.20	0.73	0.72	0.34	0.55	2.36	0.69	2.78	1.68
DOI	dB	37.32	29.26	39.40	43.93	33.83	41.57	39.68	47.48	38.49	39.00
	s	16.48	4.73	5.30	11.94	2.13	5.52	19.34	10.30	20.69	10.71
NEDD	dB	38.03	30.75	39.18	43.93	34.26	41.84	39.81	47.51	38.57	39.32
	s	40.64	4.88	17.84	40.13	4.98	17.70	40.11	39.89	39.97	27.35
LCID	dB	37.98	31.27	39.34	44.24	34.22	41.85	39.94	47.79	38.65	39.47
	s	2.72	0.36	1.27	2.83	0.38	1.23	2.78	2.80	2.73	1.90
MELA	dB	37.98	31.38	39.31	44.19	34.14	41.99	39.93	47.73	38.66	39.48
	s	0.69	0.08	0.31	0.69	0.09	0.31	0.69	0.70	0.69	0.47
FDD	dB	37.89	31.73	39.30	44.11	33.93	42.24	39.93	47.56	38.58	39.47
	s	20.95	10.61	6.61	7.17	2.58	6.83	22.86	5.27	32.81	12.85
LABI	dB	37.71	31.93	39.10	44.04	33.77	42.09	39.80	47.53	38.55	39.39
	s	49.55	6.86	21.81	50.17	6.28	22.00	49.55	49.14	51.77	34.13
EMD	dB	37.73	31.12	39.09	44.14	33.99	41.31	39.82	47.68	38.52	39.27
	s	6.78	0.95	3.05	7.11	0.92	3.11	7.34	6.95	7.03	4.81
EPD	dB	36.71	29.29	38.87	43.83	33.76	40.48	39.58	47.42	38.25	38.69
	s	84.70	11.11	38.16	86.75	11.06	38.81	84.88	85.78	83.77	58.34
FDED	dB	37.92	31.37	39.18	44.11	33.93	42.09	39.84	47.57	38.61	39.40
	s	532.20	68.17	236.09	532.50	67.64	236.78	534.13	531.44	535.25	363.80
CAD	dB	38.43	31.85	39.40	44.47	34.54	42.32	40.03	47.99	38.67	39.74
	s	103.25	13.06	45.66	102.67	12.97	46.00	103.39	102.56	103.00	70.28
Proposed	dB	39.43	32.63	39.55	44.57	34.29	42.57	40.74	48.09	39.48	40.15
	s	3.17	0.41	1.41	3.16	0.41	1.41	3.17	3.14	3.14	2.16

Kimono, Riverbed, Sunflower, and Tractor) were used for the extensive simulation. We chose to use the PSNR as an objective performance measurement, and it can be obtained as follows:

$$\mathrm{MSE}(x_{\mathrm{org}}, x_{\mathrm{rec}}) = \frac{\sum_{i=1}^{\mathrm{width}} \sum_{j=1}^{\mathrm{height}} [x_{\mathrm{org}}(i,j) - x_{\mathrm{rec}}(i,j)]^2}{\mathrm{width} \times \mathrm{height}}, \quad (15)$$

$$\mathrm{PSNR}(x_{\mathrm{org}}, x_{\mathrm{rec}}) = 10 \log_{10} \frac{255^2}{\mathrm{MSE}(x_{\mathrm{org}}, x_{\mathrm{rec}})}, \quad (16)$$

where x_{org} and x_{rec} represent the original and reconstructed images of the width × height, respectively. All the test images were converted from the original size into the vertically interlaced size according to the system[3] shown in Fig. 8, and then the interpolated image by various deinterlacing methods was compared to the original image.

Table 1 depicts the average PSNR and processing time results for the different intrafield deinterlacing methods including our proposed methods. The proposed method provided the best average PSNR performance (average PSNR improvement up to 0.83, 0.76, 1.46, and 0.75 dB when compared to the NEDD, LABI, EPD, and FDED, respectively), while reducing the average processing time up to 92.12, 93.68, 96.30, and 99.41% when compared to the NEDD, LABI, EPD, and FDED, respectively. Moreover, the proposed method obtained 0.41 dB higher PSNR than the CAD method with 96.93% less processing time. It is also noticeable that the DOI gave a relatively lower average PSNR than did MELA due to its large search range. We observed that our proposed method outperformed the conventional intrafield deinterlacing methods in terms of PSNR. Figures 9 and 10 show the average PSNR results and average processing time results, respectively.

For the subjective performance evaluation of the result images processed by different methods, the Tractor image was used as shown in Fig. 11. Magnified images are presented in Fig. 11 to enable the subjective consideration of the edge details. The full-size original Tractor image and its partial image within the red rectangle are shown in the first row of Fig. 11. The images, in Fig. 11, with the white label are the enlarged reconstructed images using LA, ELA, EELA, DSWVC, DOI, NEDD, LCID, MELA, FDD, LABI, EMD, EPD, FDED, CAD, and the proposed method. The corresponding absolute difference images between the original partial image and the reconstructed images are located right next to the reconstructed images. Here, the absolute difference image simply means the absolute pixel difference between the original image and the reconstructed images by various methods. Although the LA method yielded a higher average PSNR value than did the other methods except for LCID, MELA, FDD, CAD, and the proposed method, it provided poor result images from the subjective point of view as shown in Fig. 11. Because the LA method always interpolates the missing pixel by averaging the upper and lower pixels, direction of texture is hard to be preserved. Because the ELA and EELA methods consider only three directions, they are very sensitive to noise; thus inaccurate edge detections lead to image degradation. The MELA, LCID, FDD, and EMD methods showed similar results. Some noiselike components can be seen in the result images by DOI, DSWVC, EPD, and LABI methods. Although the CAD method preserves the fundamental edge of the original image, blurring effects are observed in the background region. In contrast, the proposed method provides the most similar image to the original image. In addition to the visual comparison of the result images to the original image, a comparison of the absolute difference images between the partial original image and the partial reconstructed images are presented. We can confirm that the resulting image is well reconstructed with the small error by comparing the absolute difference images. The absolute difference image is a binary image. The values (absolute pixel difference between the original and the reconstructed images) of <20 are displayed as black, and the values of >20 are displayed as white. Because the white pixels represent the large interpolation error, a good deinterlacing method should have less white pixels within a certain region. Note that the absolute difference images are obtained with the same magnified regions of corresponding reconstructed images and original image. It is obvious that the proposed method reconstructed the image with the smallest interpolation error because the proposed method yields the least number of white pixels. These extensive experimental results show that the proposed methods are superior to other intrafield deinterlacing methods in terms of subjective image quality.

5 Conclusion

In this paper, an efficient intrafield deinterlacing method based on the local region modeling is proposed. Instead of using a directional difference measure, the interlaced image is restored using the local surface model designed by the quadratic equation. The optimal coefficients of the surface model are determined with the given neighbor pixels around the current pixel to be interpolated. Using this model, the last coefficient can be treated as the estimated current pixel. To further reduce the computational time, the precalculated ninth row of the inverse matrix, fixed and common for every pixel to be interpolated, is utilized. Hence, it is possible to lower the computational complexity of the proposed method. We have shown from the experiments that the proposed method has lowered the overall processing time while improving its performance compared to the conventional methods. The extensive experimental results demonstrated that the proposed method outperformed conventional intrafield deinterlacing methods in terms of both the objective and subjective image qualities. The proposed method reduced the average CPU time up to 99.4 and 96.9% when compared to FDED and CAD, respectively, while providing higher PSNR values. Therefore, the proposed method is able to reduce the complexity of the deinterlacing process, which is regarded as an essential part in postprocessing at an HDTV.

Acknowledgments

This research was supported by Seoul Future Contents Convergence (SFCC) Cluster established by Seoul R&BD Program (Grant No. 10570).

References

1. K. Jack, *Video Demystified—A Handbook for the Digital Engineer*, 4th ed., Elsevier, Jordan Hill, Oxford (2005).
2. G. De Haan, "Television display processing: past & future," in *Proc. IEEE ICCE'07, Las Vegas*, pp. 1–2 (2007).
3. E. B. Bellars and G. De Haan, "*De-interlacing: A key technology for scan rate conversion*," Elsevier, Amsterdam (2000).
4. T. Doyle, "Interlaced to sequential conversion for EDTV applications," in *Proc. 2nd Int. Workshop Signal Processing of HDTV*, pp. 412–430 (1998).
5. T. Chen, H. R. Wu, and Z. H. Yu, "Efficient deinterlacing algorithm using edge-based line average interpolation," *SPIE Opt. Eng.* **39**(8), 2101–2105 (2000).
6. Y. Kim, "Deinterlacing algorithm based on sparse wide vector correlations," *SPIE Opt. Eng.* 2727, 89–99 (1996).
7. W. Kim, S. Jin, and J. Jeong, "Novel intra deinterlacing algorithm using content adaptive interpolation," *IEEE Trans. Cons. Elect* **53**(3), 1036–1043 (2007).
8. H. Yoo and J. Jeong, "Direction-oriented interpolation and its application to de-interlacing," *IEEE Trans. Cons. Elect.* **48**(4), 954–962 (2002).
9. M. K. Park, M. G. Kang, K. Nam, and S. G. Oh, "New edge dependent deinterlacing algorithm based on horizontal edge pattern," *IEEE Trans. Cons. Elect.* **49**(4), 1508–1512 (2003).
10. P.-Y. Chen and Y.-H. Lai, "A low-complexity interpolation method for deinterlacing," *IEICE Trans. Inf. Syst.* **E90-D**(2), 606–608 (2007).
11. D.-H. Lee, "A new edge-based intra-field interpolation method for deinterlacing using locally adaptive-thresholded binary image," *IEEE Trans. Cons. Elect.* **54**(1), 110–115 (2008).

12. F. Michaud, C. T. Le Dinh, and G. Lachiver, "Fuzzy detection of edge-direction for video line doubling," *IEEE Trans. Circuits Syst. Video Technol.* **7**(3), 539–542 (1997).
13. S. Jin, W. Kim, and J. Jeong, "Fine directional de-interlacing algorithm using modified Sobel operation," *IEEE Trans. Cons. Elect.* **54**(2), 857–862 (2008).
14. K. Kang, G. Jeon, and J. Jeong, "A single field interlaced to progressive format conversion using edge map in the image block," in *Proc. IASTED SIP 2009*, pp. 80–85, Hawaii, USA, 606–608 (2009).
15. S. Yang, D. Kim, and J. Jeong, "Fine edge-preserving deinterlacing algorithm for progressive display," *IEEE Trans. Cons. Electron.* **55**(3) (2009).
16. S.-J. Park, G. Jeon, and J. Jeong, "Computation-aware algorithm selection approach for interlaced-to-progressive conversion," *SPIE Opt. Eng.* **49**(5), 057005 (2010).
17. R. Li, B. Zheng, and M. L. Liou, "Reliable motion detection/compensation for interlaced sequences and its applications to deinterlacing," *IEEE Trans. Circuits Syst. Video Technol.* **10**(1), 23–29 (2000).
18. D. Wang, A. Vincent, and P. Blanchfield, "Hybrid de-interlacing algorithm based on motion vector," *IEEE Trans. Circuits Syst. Video Technol.* **15**(8), 1019–1025 (2005).
19. O. Kwon, K. Sohn, and C. Lee, "Deinterlacing using directional interpolation and motion compensation," *IEEE Trans. Cons. Elect.* **49**(1), 198–203 (2003).
20. X. Li and M. T. Orchard, "New edge-directed interpolation," *IEEE Trans. Image Process.* **10**(10), 1521–1527 (2001).
21. N. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*. Prentice-Hall, Englewood Cliffs, NJ (1984).
22. J. W. Suh and J. Jeong, "Fast sub-pixel motion estimation techniques having lower computational complexity," *IEEE Trans. Cons. Elect.* **50**(3), 968–973 (2004).

Sang-Jun Park received his BS and MS in electronic engineering from Hanyang University, Seoul, Korea, in 2006 and 2008, respectively. He is currently a PhD candidate in the Department of Electronics and Computer Engineering, Hanyang University. His research interests fall under the umbrella of image processing, video coding standards such as H.264/AVC, SVC, motion estimation, demosaicing, and image-enhancement as well as image-resizing algorithms.

Jechang Jeong received his BS in electronic engineering from Seoul National University, Korea, in 1980, MS in electrical engineering from the Korea Advanced Institute of Science and Technology in 1982, and P.D in electrical engineering from the University of Michigan, Ann Arbor, in 1990. From 1982 to 1986, he was with the Korean Broadcasting System, where he helped develop teletext systems. From 1990 to 1991, he worked at the University of Michigan, Ann Arbor, as a Postdoctoral Research Associate, where he helped to develop various signal processing algorithms. From 1991 through 1995, he was with the Samsung Electronics Company, Korea, where he was involved in the development of HDTV, digital broadcasting receivers, and other multimedia systems. Since 1995, he has conducted research at Hanyang University, Seoul, Korea. His research interests include digital signal processing, digital communication, and image/audio compression for HDTV and multimedia applications. He has had over 30 technical papers published. Dr. Jeong received the "Scientist of the Month" award in 1998, from the Ministry of Science and Technology of Korea and was also honored with a government commendation in 1998, from the Ministry of Information and Communication of Korea. He was with the recipient of the IEEE Chester Sall Award in 2007 and 2008 ETRI Journal Paper Award.