

## Research Article

# An Interfield and Intrafield Weighted Interpolative Deinterlacing Algorithm Based on Low-Angle Detection and Multiangle Extraction

Jun Qiang,<sup>1</sup> Jun Chen,<sup>2</sup> and Jun Wang<sup>1</sup>

<sup>1</sup> School of Computer Information and Engineering, Anhui Polytechnic University, Wuhu 241000, China

<sup>2</sup> Department of Electrical Engineering, Anhui Technical College of Mechanical and Electrical Engineering, Wuhu 241000, China

Correspondence should be addressed to Jun Qiang; [liz.chiang@163.com](mailto:liz.chiang@163.com)

Received 15 April 2014; Revised 30 June 2014; Accepted 6 July 2014; Published 12 August 2014

Academic Editor: Qingsong Xu

Copyright © 2014 Jun Qiang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the process of converting interlaced scanning to progressive scanning, interline flickers, saw-tooth, and creeping could be found in motion images. This paper proposes an interfield and intrafield weighted interpolative algorithm based on low-angle detection and multiangle extraction. Using interframe difference of vertical edge area in current field and interfield difference of other areas as the input of motion detection, incorrect judgment could be avoided by four-fields motion detection so that static and motive image can be distinguished. Based on the motion image from the motion detection, multipixels are used in low-angle horizontal detection to extract higher correction angles so as to make interfield and intrafield weighted interpolation. The experiment results show that the proposed algorithm could improve peak signal-to-noise ratio (PSNR), restrain some interlace phenomenon, and reach better visual effects. Meanwhile, there is a good balance between efficiency and computation cost.

## 1. Introduction

Traditional video sources mostly adopt interlaced scanning mode. It can reduce the complexity of the device. But, in the front-end of intelligent video surveillance systems with the development of high definition digital video, when inputting interlaced scanning video into display devices with progressive scanning mode, flaws, for example, interline flickers, saw-tooth, and creeping, appear, which result in bad video quality and visual fatigue [1, 2]. The function of deinterlacing is to convert interlaced images to progressive ones and obtain fine video effects.

In the past decades, some deinterlacing algorithms, for example, linear filter, adaptive nonlinear filtering, and motion compensation (MC), have been proposed. Linear filter which has low cost of hardware and low algorithm complexity is easy to realize, but an ideal result is difficult to achieve. Compared with linear filtering, adaptive nonlinear filter

has fine results of deinterlacing but needs more costs of hardware. The algorithm of motion compensation uses the related information of spatial and temporal domain to get smooth and natural effects in motion. With its processing, it can overcome stagnation and shake which the previous two algorithms could bring on. Motion compensation has the best performance among these three algorithms, while it has the biggest amount of calculation and the most sensitivity in error. It may cause some serious problems in visual effects such as blocking effect and localized distortion [3, 4].

In order to address the problem of MC, nonmotion compensation algorithm (non-MC) is proposed [5]. Among non-MC techniques, intrafield interpolation (in spatial domain) and interfield interpolation (in temporal domain) are distinguished. Interfield algorithm interpolates the missing lines by employing pixels from different fields, using field insertion and  $t$ -line average methods. Interfield algorithm works properly in the static parts of the image. Spatial interpolation

algorithms calculate the lines by interpolating the adjacent lines from the same field, such as line averaging or directional [1, 6, 7]. Intrafield algorithms work much better in the presence of motion but cannot work well in static parts. The main idea of motion adaptive deinterlacing algorithm is proposed in [8].

The biggest problem of motion adaptive deinterlacing algorithms is the detection of the motion. So motion detection should be made before deinterlacing. The algorithms in [9, 10] use a fuzzy inference rule base to choose deinterlacing method according to the motion detection. Recently, several methods have been proposed to deinterlace by focusing on edges [1, 6].

Our research mainly focuses on two points. Firstly, an accurate motion detection method will be used. Secondly, based on the result of motion detection, we find a suitable deinterlacing algorithm to solve the interlace effect.

As for the motion detection, we adopt four-field motion detection which obtains the motion information by using adjacent four fields' luminance difference. It is easy to obtain the motion information which is used commonly. But, in previous research, this motion detection method may cause deviation. In [11], Chang et al. proposed the same parity four-field motion detection applied in MC deinterlacing which has achieved accurate effect. And Yao et al. [12] mentioned a modified four-field method for motion adaptive interpolation. They both mentioned the importance of four-field motion detection. In this paper, we adopt further detection. Interframe difference is used as the input of motion detection in vertical edge area of current field. And other areas use interfield difference as the input motion detection. It can avoid incorrect judgment.

In order to improve the accuracy of horizontal directive detection, the interpolative method, and the effects of deinterlacing, in this paper, we proposed an interfield and intrafield weighted interpolative algorithm based on low-angle detection and multiangle extraction. First, adjacent four fields for motion detection are used to distinguish static and motive parts. Then, based on the result of the motion detection, more correlative pixels of interpolated pixels in adjacent lines are increased to improve the accuracy of horizontal directive detection. Finally, through extracting pixel angles of higher correlation degree and setting weighted coefficient, the intrafield and interfield weight interpolation are taken. The proposed method could solve the low-angle edge detection which could not be solved by traditional algorithms. Moreover, combined with intrafield and interfield interpolation, using weighted average to calculate the value of the interpolated pixel, the effect of interpolation gets a fine result. The experimental results show that this algorithm efficiently improves PSNR and has an efficient computation time ratio.

The remainder of the paper is organized as follows: Section 2 presents four-field motion detection. Section 3 describes the proposed algorithm of deinterlacing by a low-angle extraction based on intrafield and interfield weighted interpolation. Section 4 gives the comparison of experiments and reports the results, followed by the conclusion in Section 5.

## 2. Motion Detection

**2.1. Problem Statement.** In the research of deinterlacing, we find that motion detection plays an important role. Motion detection is usually used to distinguish motive image and static image, and then different deinterlacing algorithm will be adopted accordingly; thus, better effects could be obtained. If the same deinterlacing algorithm is used in all the processes, the effect of the algorithm could not be demonstrated. Consequently, phenomenon such as no smooth edge, virtual image, breakpoint, and hazy detail will appear. In [13–15], Lee et al. mentioned the importance of motion detection. However, the incorrect motion detection methods cause two kinds of result, motion misjudgment and motion missing. Motion misjudgment mistakes the motive image as a static one. Motion missing is a situation that the motive images cannot be detected. These two problems cause serious effects on deinterlacing. So, motion detection needs to minimize the phenomenon of incorrect judgments [16]. With the accurate motion detection, follow-up deinterlacing algorithm could adaptively choose suitable method to deal with motive or static images.

**2.2. Algorithm Description.** Four-field motion detection, which is shown in Figure 1, is used in the proposed algorithm. It denotes the pixels needed in four fields.

Motion information could be obtained by using adjacent four fields' luminance difference. Prior two fields, prior one field, current field, and rear one field are denoted by  $f - 2$ ,  $f - 1$ ,  $f$ , and  $f + 1$ , separately. The interpolated pixel in motion detection is denoted by  $p$ .

Two kinds of absolute image errors are defined:

$$\begin{aligned} D1 &= \{|Z_{Ln-1} - Z_{Ln+1}|, |Z_{n-1} - Z_{n+1}|, |Z_{Rn-1} - Z_{Rn+1}|\}, \\ D2 &= \{|X_{Ln-2} - X_{Ln}|, |X_{n-2} - X_n|, |X_{Rn-2} - X_{Rn}|, \\ &\quad |Y_{Ln-2} - Y_{Ln}|, |Y_{n-2} - Y_n|, |Y_{Rn-2} - Y_{Rn}|\}. \end{aligned} \quad (1)$$

Detection input is

$$d = D1 \cup D2. \quad (2)$$

Comparing threshold  $T$  which was set in advance with  $d$  from (2), the result of detection is outputted. Motive images are defined as "1" and static images as "0."

In previous research, this motion detection method may cause deviation. So, further detection is needed. A suitable method is adopted as follows: interframe interpolation is used as the input of motion detection in vertical edge area of current field, while interfield interpolation is used as the input motion detection in other areas. It can avoid incorrect judgment.

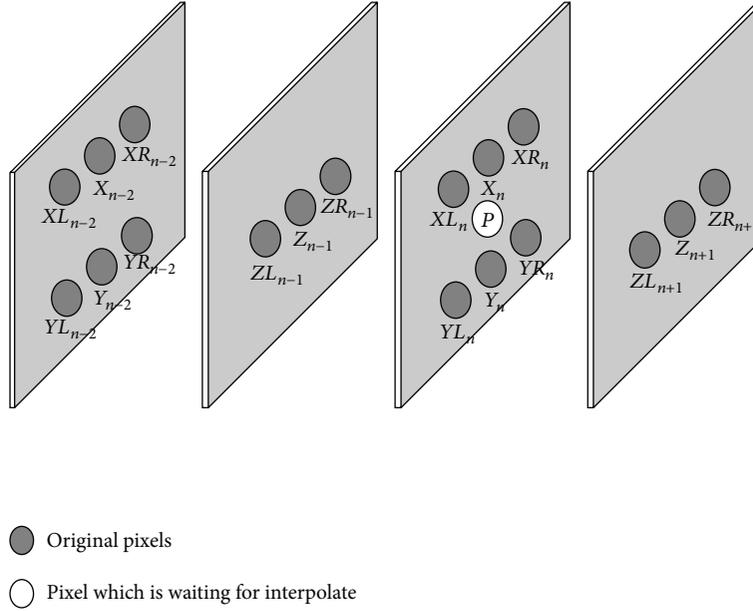


FIGURE 1: Schematic diagram of four-field motion detection.

Fields differences are defined as follows:

$$\begin{aligned}\Delta f_1 &= |Z_{n-1} - Z_{n+1}|, \\ \Delta f_2 &= \left| Z_{n-1} - \frac{(X_n + Y_n)}{2} \right|, \\ \Delta f_3 &= \left| Z_{n+1} - \frac{(X_n + Y_n)}{2} \right|,\end{aligned}\quad (3)$$

where  $\Delta f_1$  represents interframe difference and  $\Delta f_2$  and  $\Delta f_3$  represent interfield difference. Motion estimate of  $p$  is

$$\Delta p = \max(\Delta f_1, |X_n - Y_n| \cdot \max(\Delta f_2, \Delta f_3)). \quad (4)$$

Extended in spatial domain by  $\Delta p$ , calculated by nonlinear two-valued function, the result of motion detection will be obtained [16, 17].

### 3. Deinterlacing Algorithm by a Low-Angle Extraction

**3.1. Problem Statement.** According to the previous analysis, in the adaptive motion deinterlacing algorithm, it is important to distinguish motion and static status. But, correct detection to the edge information in motive image and proper application of the edge information to the spatial interpolate filter are also very important to improve the quality of image.

Edge-based line average (ELA) algorithm is widely used in deinterlacing. But, its defect is that it is hard to detect the horizontal edge. ELA only detects the edges in  $45^\circ$ ,  $90^\circ$ , and  $135^\circ$  and is unable to detect low-angle edge [18]. Kuo et al. proposed the ELA based on  $3 + 3$  taps which improved image horizontal edge detection by comparing the differences of

adjacent pixels [19]. Lee et al. expand the  $3 + 3$  taps ELA to  $5 + 5$  taps [20] and de Haan and Lodder to  $7 + 7$  taps [21]. All these improved algorithms help to improve the low-edge detect function, but they are unable to reach the horizontal edge. The  $7 + 7$  taps can only detect the smallest angle in  $18^\circ$ . Chang et al. proposed EIELA using adaptive method, which has large amount of calculation and complexity [22]. It is hard to realize in hardware and not a suitable method in video surveillance system. Yong et al. proposed a spatial-temporal weight and edge adaptive deinterlacing algorithm (STW-EA) [23]. It is a good method to use low-angle edge detection and spatial-temporal weight calculation. It detects low-angle edge by using an adaptive searching radius in which the  $6^\circ$  edge can be detected. This algorithm achieves high image quality and low hardware complexity, but it still has problems in dealing with details, especially in high speed motive image.

**3.2. Algorithm Design and Description.** In order to improve the effect of deinterlacing and get high quality of details, we propose an algorithm combined with low-angle detection and intrafield and interfield bidirectional interpolation based on angle extraction. Considering the advantages and disadvantages of the existing edge detection algorithm, an improved algorithm is designed from two aspects: (1) expand detection region and search more pixels in one line in order to find the edge direction accurately; (2) use angle analysis result and extract the highest correlated angle to interpolate, both in intrafield and interfield.

Let  $p(i, j)$  be the interpolated pixel and related pixels in adjacent lines are  $a_0, a_1, \dots, a_{16}$  of line  $j - 1$  and  $b_0, b_1, \dots, b_{16}$  of line  $j + 1$ , shown in Figure 2.

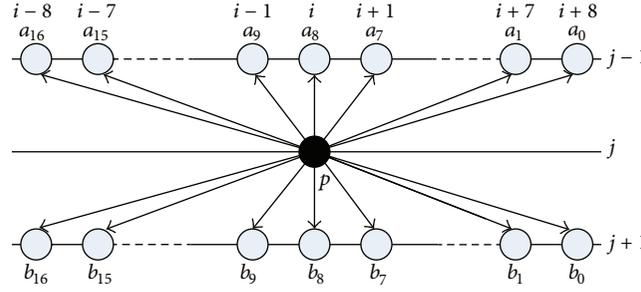


FIGURE 2: Low-angle edge detection.

Pixels in diagonal direction on two rows represent the probable edge direction. By comparing the absolute difference of luminance values among the pixels in diagonal direction, the correlation of pixels in diagonal direction and interpolated pixel could be determined. The smaller the value of the absolute difference is, the bigger correlation it has. The direction of minimum difference is the edge direction. We select 17 pixels in one row; it means that 18 angles with horizontal direction will be detected, such as  $90^\circ$  ( $270^\circ$ ),  $45^\circ$  ( $135^\circ$ ),  $26.57^\circ$  ( $123.43^\circ$ ),  $18.43^\circ$  ( $161.57^\circ$ ),  $14.03^\circ$  ( $163.97^\circ$ ),  $11.3^\circ$  ( $168.7^\circ$ ),  $9.46^\circ$  ( $170.54^\circ$ ),  $8.13^\circ$  ( $171.87^\circ$ ), and  $7.12^\circ$  ( $172.88^\circ$ ).

Proposed algorithm is described as follows.

*Step 1.* The angles mentioned above are used to find the pixels which correspond to interpolated pixel. In (5), (6), and (7), all 18 angles are shown:

$$\varphi_{p_{m\theta}} = a_8(i, j-1), b_8(i, j+1), \theta = 90^\circ, \quad (5)$$

$$\varphi_{P_{i\theta}} = \begin{cases} a_0(i+8, j-1), b_{16}(i-8, j+1), \theta = 7.12^\circ, \\ a_1(i+7, j-1), b_{15}(i-7, j+1), \theta = 8.13^\circ, \\ \vdots \\ a_7(i+1, j-1), b_9(i-1, j+1), \theta = 45^\circ, \end{cases} \quad (6)$$

$$\varphi_{P_{r\theta}} = \begin{cases} a_{16}(i-8, j-1), b_0(i+8, j+1), \theta = 172.88^\circ, \\ a_{15}(i-7, j-1), b_1(i+7, j+1), \theta = 171.87^\circ, \\ \vdots \\ a_9(i-1, j-1), b_7(i+1, j+1), \theta = 135^\circ. \end{cases} \quad (7)$$

*Step 2.* In order to state clearly, the diagonal pixels are divided into three parts: (1) middle part,  $90^\circ$  angles,  $a_8$  and  $b_8$ ; (2) right part, pixels on the right of  $a_8$  and right of  $b_8$ ; (3) left part, pixels on the left of  $a_8$  and right of  $b_8$ . Just as defined in (5), (6), and (7), the angles are divided into three parts. Equation (5) denotes the middle part, (6) denotes the right part, and (7) denotes the left part.

Then, the threshold functions are defined as follows, which are the absolute difference of diagonal pixels:

$$\text{middle part: } S_M = |Y(i, j-1, n), Y(i, j+1, n)|, \quad (8)$$

$$\text{right part: } S_{Rk} = |Y(i+k, j-1, n), Y(i-k, j+1, n)|, \\ k = 1, 2, \dots, 8, \quad (9)$$

$$\text{left part: } S_{Lk} = |Y(i-k, j-1, n), Y(i+k, j+1, n)|, \\ k = 1, 2, \dots, 8. \quad (10)$$

$Y(i, j)$  denotes the luminance component of pixel  $(i, j)$  and  $n$  denotes the  $n$ th field.

*Step 3.* Intrafield: we choose the optimal direction angle by comparing these differences to select the minimum value, and then the diagonal direction which is corresponding to minimum value is generally the edge direction:

$$S = \text{MIN}(S_M, S_{Rk}, S_{Lk}), \quad k = 1, 2, \dots, 8. \quad (11)$$

If all the absolute differences of two pixels in diagonal direction are over the  $S$ , we consider that no edge exists. Line averaging is managed.

*Step 4.* Interfield: considering the accuracy of edge detection, other two adjacent fields are used for calculating the temporal correlation. The direction of the  $S$  (get from Step 3) is the most correlation direction. Two corresponding pixels in fields  $n-1$  and  $n+1$  are also the most correlative pixels. They are considered as the two positions before and after the motion. Adopting bidirectional interpolation, this algorithm uses the average of these two pixels to interpolate.

At the basis of Step 3, adjacent fields' information is used. All directions' absolute differences of the same line in adjacent fields and the up-down lines in adjacent fields are defined as follows (Figure 3).

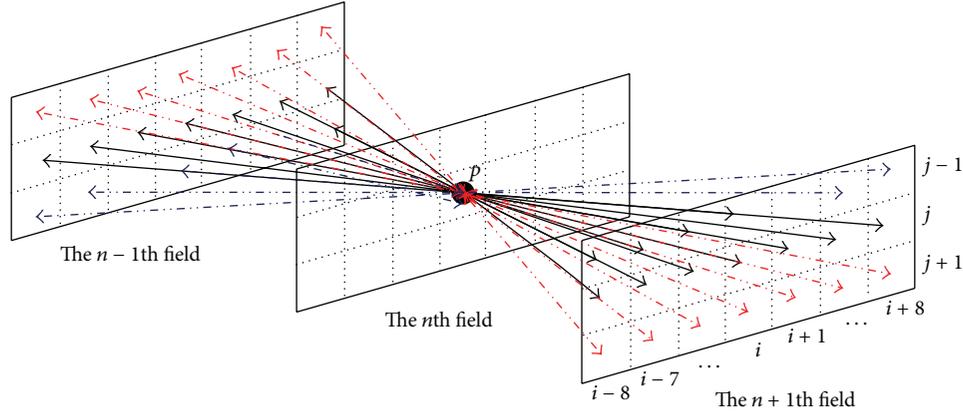


FIGURE 3: The directions of interfield interpolation (the black arrows denote the two pixels related to interpolated pixel  $p$  in adjacent two fields which are in the same line, and red and blue arrows denote the adjacent two fields but in the up-down lines in diagonal directions).

Same line:

$$T_{A_k} = |Y(i-k, j, n-1) - Y(i+k, j, n+1)|, \quad (12)$$

$$k = 8, 7, \dots, 1, 0, -1, \dots, -7, -8.$$

Adjacent line:

$$T_{B_k} = |Y(i-k, j-1, n-1) - Y(i+k, j+1, n+1)|, \quad (13)$$

$$k = 8, 7, \dots, 1, 0, -1, \dots, -7, -8,$$

$$T = \text{MIN}(T_{A_k}, T_{B_k}), \quad k = 8, 7, \dots, 1, 0, -1, \dots, -7, -8. \quad (14)$$

The minimum of all directions from native and adjacent fields  $T$ , as shown in (14), is considered the most correlated direction in all directions, which will be interpolated.

*Step 5.* According to Steps 3 and 4, intrafield and interfield interpolation are both managed. Spatial difference “ $S$ ” (get from (11)) is the absolute difference of correlated pixels in forward and back fields. Temporal difference “ $T$ ” (get from (14)) is the absolute difference of diagonal pixels in up-down lines which is in the edge direction of intrafield. We define the spatial weighted coefficient  $r_s$  and temporal weighted coefficient  $r_t$  as follows:

$$r_t = \frac{S}{(S+T)}, \quad r_s = 1 - r_t. \quad (15)$$

Then, using  $V_s$  which is the value of the intrafield interpolation and  $V_t$  which is the value of the interfield interpolation to calculate the value of weighted interpolated pixel, formula is as follows:

$$V = V_s * r_s + V_t * r_t. \quad (16)$$

*Step 6.* The value of interpolated pixel is used for filter. If the adjacent direction consistent with interpolated pixel, no operation will be taken. Otherwise, median filtering is adopted to revise the adjacent pixel which has been interpolated.

## 4. Experimental Results

*4.1. Experiment Setup and Result.* The whole deinterlacing system is simulated by using ISE 14.1 design tools, 2v1000fg256-4 simulation device by Xinlinx, and ModelSim SE 7.0 simulation tools.

After simulation, PSNR and MSE are contrasted between original image and the processed image by proposed algorithm. The definitions are

$$\text{MSE} = \frac{1}{M \times N} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} [P_{\text{src}}(i, j) - P_{\text{de}}(i, j)]^2, \quad (17)$$

$$\text{PSNR} = 10 \cdot \log_{10} \left( \frac{255 \times 255}{\text{MSE}} \right).$$

$M \times N$  denotes resolution of original image.  $P_{\text{src}}(i, j)$  is the luminance value of original image in point  $(i, j)$ .  $P_{\text{de}}(i, j)$  is the luminance value of deinterlaced image in point  $(i, j)$ .

The experiment is designed as follows. Two odd-even overlapped fields test sequences of 25 frames length are used for algorithm testing. The 50th frame progressive image of sequence plays fluently and has fine test result by deinterlacing the proposed algorithm. Experimental results are shown in Figure 4 and Table 1.

Original images in Figure 4 are the 35th frame in test sequence. The proposed algorithm has also been tested by using several standard video sequences and the results have been compared with other deinterlacing algorithm, such as ELA, ELA using 3 + 3 and 5 + 5 windows, STW-EA, and ECA. By contrast to Table 1, PSNR of test sequences are both improved by the proposed algorithm. We got the fluent visual effect and removed saw-tooth phenomenon.

*4.2. Analysis.* From Figure 4, we could find that traditional ELA has the drawback of low vertical definition. And using intrafield interpolation directly leads to fuzzy-edge phenomenon. ELA using 3 + 3 and 5 + 5 windows are also making errors when processing nonclear edges or ambiguous

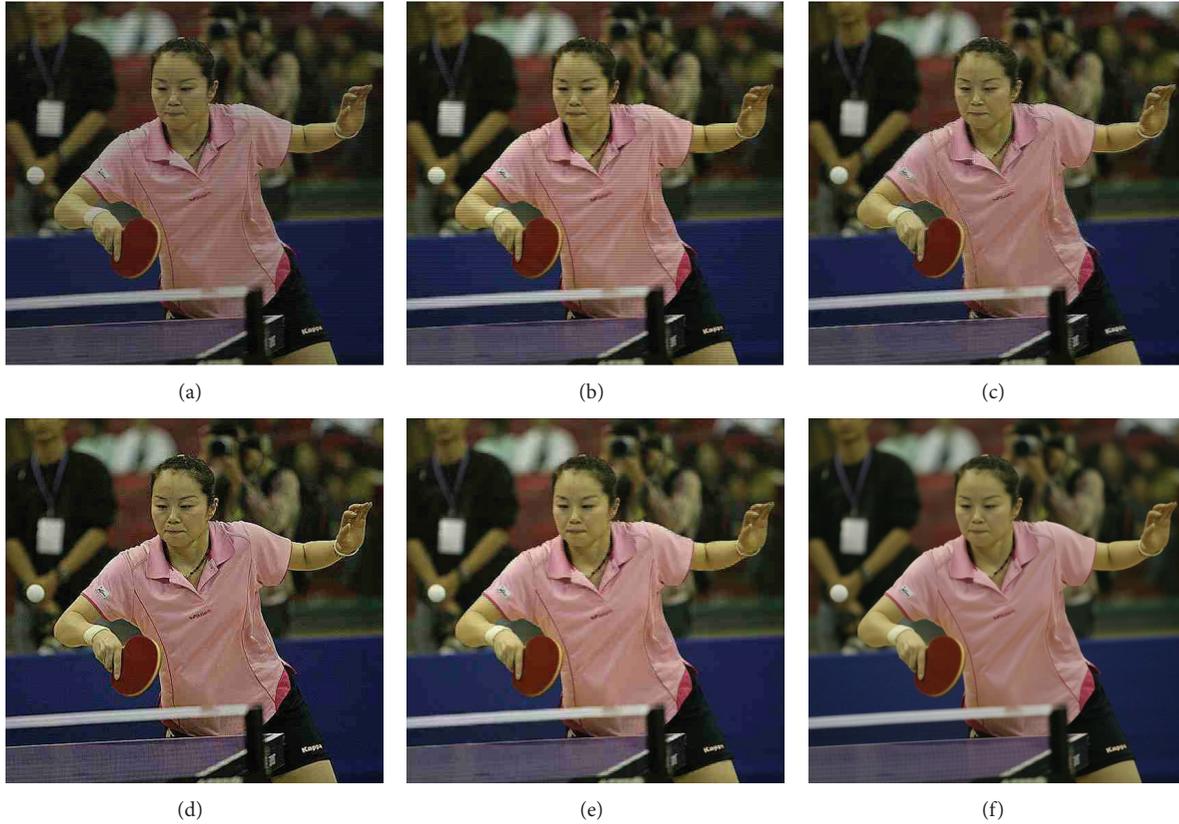


FIGURE 4: Comparison of testing result ((a) interlaced original image, (b) by ELA, (c) ELA with 3 + 3 window, (d) ELA with 5 + 5 window, (e) STW-EA, and (f) proposed algorithm).

TABLE 1: (Quality) PSNR values (in dBs) for different deinterlacing algorithms.

Algorithms	Sequence					Average
	Table tennis	Basketball	Salesman	News		
Original	30.23	23.31	27.79	24.76		<b>26.52</b>
ELA	32.34	24.67	30.12	26.03		<b>28.29</b>
ELA 3 + 3	32.97	25.12	32.11	26.63		<b>29.21</b>
ELA 5 + 5	33.16	26.98	32.18	26.89		<b>29.80</b>
STW-EA	34.72	27.64	34.22	27.49		<b>31.02</b>
Proposed	35.99	29.35	36.03	31.52		<b>33.22</b>

situations, as we analyzed in Section 2. The proposed algorithm combined with interfield interpolation and intrafield low-angle edge detection solves these problems well.

Computational time ratio, which means quality/cost ratio, is calculated to evaluate the proposed algorithm if it is competitive versus the other deinterlacing algorithms. Table 1 (PSNR) shows the PSNR of every test sequence, which indicates the “quality.” Table 2 (CPU time) shows the consumed CPU times of every algorithm which means “cost.” In Tables 1 and 2, the last columns both mean the average value of every algorithm, which are used in calculating the computational time ratio in Table 3.

In Table 3, the proposed algorithm gets the best quality/cost ratio. But traditional algorithm ELA’s computational

time ratio is better than ELA 3 + 3, ELA 5 + 5, and STW-EA. Analyses show that ELA 3 + 3, ELA 5 + 5, and STW-EA are more complex than ELA and have better PSNR. ELA has less cost of CPU times but has bad deinterlacing quality. It is not recommended in intelligent video surveillance systems. As all things are considered, our approach offers a better quality/cost in comparison with other traditional algorithms.

## 5. Conclusion

Through the research and analysis of existed deinterlacing algorithms, this paper summarizes their merits and drawbacks and proposes a novel deinterlacing algorithm.

TABLE 2: (Cost) CPU time(s).

Algorithms	Sequence				
	Table tennis	Basketball	Salesman	News	Average
ELA	29.97	22.86	27.91	24.12	<b>26.21</b>
ELA 3 + 3	31.07	23.68	30.26	25.10	<b>27.53</b>
ELA 5 + 5	30.99	25.21	30.07	25.13	<b>27.85</b>
STW-EA	32.27	25.69	31.80	25.55	<b>28.83</b>
Proposed	33.17	27.05	33.21	29.05	<b>30.62</b>

TABLE 3: Computational time ratio.

	ELA	ELA 3 + 3	ELA 5 + 5	STW-EA	Proposed
Quality/cost	1.079	1.061	1.070	1.076	1.085

Effective four-field motion detection is adopted to avoid the incorrect judgment, which could improve the accuracy of motion detection. Low-angle detection and multiangle extraction can obtain higher corrected pixels angle. Interpolation with weighted interfield and intrafield increases accuracy of interpolation. Due to the proposed algorithm with higher ability of multiangle extraction and interpolation, we get good effect on deinterlacing. By theoretical and experimental analysis, this algorithm overcomes the drawbacks of traditional algorithms, gets subjective evaluation and PSNR and computational ratio contrasted in objective evaluation, and obtains better visual effects.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgment

This research was supported by the Natural Science Foundation of the Anhui Higher Education Institutions of China (Grant no. KJ2013B031).

## References

- [1] G. Jeon, M. Anisetti, V. Bellandi, E. Damiani, and J. Jeong, "Designing of a type-2 fuzzy logic filter for improving edge-preserving restoration of interlaced-to-progressive conversion," *Information Sciences*, vol. 179, no. 13, pp. 2194–2207, 2009.
- [2] P. Brox, I. Baturone, S. Sánchez-Solano, J. Gutiérrez-Ríos, and F. Fernández-Hernández, "A fuzzy edge-dependent motion adaptive algorithm for de-interlacing," *Fuzzy Sets and Systems*, vol. 158, no. 3, pp. 337–347, 2007.
- [3] G. de Haan and E. B. Bellers, "De-interlacing an overview," *Proceedings of the IEEE*, vol. 8, pp. 819–825, 1998.
- [4] H. Hwang, M. H. Lee, and D. I. Song, "Interlaced to progressive scan conversion with double smoothing," *IEEE Transactions on Consumer Electronics*, vol. 39, no. 3, pp. 241–246, 1993.
- [5] G. de Haan, "De-interlacing," in *Digital Video Post Processing*, vol. 9, pp. 185–201, University Press Eindhoven, 2006.
- [6] G. Jeon, M. Anisetti, J. Lee, V. Bellandi, E. Damiani, and J. Jeong, "Concept of linguistic variable-based fuzzy ensemble approach: application to interlaced HDTV sequences," *IEEE Transactions on Fuzzy Systems*, vol. 17, no. 6, pp. 1245–1258, 2009.
- [7] S. Park, G. Jeon, and J. Jeong, "Deinterlacing algorithm using edge direction from analysis of the DCT coefficient distribution," *IEEE Transactions on Consumer Electronics*, vol. 55, no. 3, pp. 1674–1684, 2009.
- [8] A. M. Bock, "Motion-adaptive standards conversion between formats of similar field rates," *Signal Processing: Image Communication*, vol. 6, no. 3, pp. 275–280, 1994.
- [9] D. van de, B. Ville, W. Philips, and I. Lamahieu, "Fuzzy-based motion detection and its application to de-interlacing," in *Fuzzy Techniques in Image Processing*, vol. 52 of *Studies in Fuzziness and Soft Computing*, pp. 337–369, Springer, 2000.
- [10] D. van de Ville, B. Rogge, W. Philips, and I. Lamahieu, "Evaluation of several operators for fuzzy-based motion adaptive de-interlacing," in *Proceedings of the PRORISC IEEE Benelux Workshop on Circuits, Systems and Signal Processing*, vol. 11, pp. 535–544, 1999.
- [11] Y. Chang, P. Wu, S. Lin, and L. Chen, "Four field local motion compensated de-interlacing," in *Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. V-253–V-256, May 2004.
- [12] Y. Yao, H. Chen, and Y. Cheng, "A low power IC for efficient de-interlacing based on refined motion adaptive method," in *Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE '12)*, pp. 1094–1099, Hangzhou, China, May 2012.
- [13] S. Lee and D. Lee, "A motion-adaptive de-interlacing method using an efficient spatial and temporal interpolation," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1266–1271, 2003.
- [14] S.-F. Lin, Y.-L. Chang, and L.-G. Chen, "Motion adaptive de-interlacing by horizontal motion detection and enhanced ELA processing," in *Proceedings of the International Symposium on Circuits and Systems (ISCAS '03)*, vol. 2, pp. II696–II699, May 2003.
- [15] B. Yoo, B. Kim, and K. Lee, "An efficient motion adaptive de-interlacing algorithm using spatial and temporal filter," in *Proceedings of the IEEE Region 10 Conference: Trends and Development in Converging Technology Towards (TENCON '11)*, pp. 288–292, Indonesia, November 2011.
- [16] S. F. Lin, Y. L. Chang, and L. G. Chen, "Motion adaptive interpolation with horizontal motion detection for de-interlacing," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1256–1265, 2003.

- [17] J. Qiang, M. Zhou, and J. Wang, "Algorithm research on eliminating saw-tooth phenomenon in motion image," *Video Engineering*, vol. 36, no. 15, pp. 32–35, 2012.
- [18] S. G. Lee and D. H. Lee, "A motion-adaptive de-interlacing method using an efficient spatial and temporal interpolation," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 4, pp. 1266–1271, 2003.
- [19] C. J. Kuo, C. Liao, and C. C. Lin, "Adaptive interpolation technique for scanning rate conversion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 317–321, 1996.
- [20] H. Y. Lee, J. W. Park, T. M. Bae, S. U. Choi, and Y. H. Ha, "Adaptive scan rate up-conversion system based on human visual characteristics," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 999–1006, 2000.
- [21] G. de Haan and R. Lodder, "De-interlacing of video data using motion vectors and edge information," in *Proceedings of the International Conference on Consumer Electronics (2002 Digest of Technical Papers)*, pp. 70–71, Los Angeles, Calif, USA, 2002.
- [22] Y. Chang, S. Lin, and L. Chen, "Extended intelligent EDGE-based line average with its implementation and test method," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '04)*, vol. 2, pp. 341–344, Vancouver, Canada, May 2004.
- [23] D. Yong, L. Shengli, and S. Longxin, "Spatio-temporal weight and edge adaptive de-interlace," *Chinese Journal of Computers*, vol. 30, no. 4, pp. 655–660, 2007.



# Hindawi

Submit your manuscripts at  
<http://www.hindawi.com>

