

Efficient Adaptive Deinterlacing Algorithm Using Bilateral Filter

Jin Wang^{1,a}, Zhensen Wu¹, Jiaji Wu²

¹ School of Physics and Optoelectronic Engineering, Xidian University, Xi'an Shaanxi, China

² School of Electronic Engineering, Xidian University, Xi'an Shaanxi, China

Abstract. This paper introduces an efficient intra-field deinterlacing algorithm using bilateral filter while considering both closeness and similarity between the interpolated pixel and the neighbor pixels. Instead of estimating the edge orientations via limited candidate directions as previous intra-field deinterlacing methods, we propose an adaptive spatial locality- and similarity-based deinterlacing method. Many deinterlacing methods described in the literature only consider the similarity between neighbor pixels and interpolated pixels; however awareness of distance correlation and spatial locality has not yet been studied. Experimental results indicate that the proposed scheme outperforms a number of existing approaches, in terms of both the objective and subjective performance.

1 Introduction

Interlaced scanning has been widely employed as the standard of TV broadcast systems, such as NTSC and SECAM, and some of the high definition television (HDTV) formats, because this method allows for not only the reduction in bandwidth and memory but also the maintenance of the existing transmission channel. However, this interlaced scanning format introduces several problematic artifacts, known as line crawling, edge flickering, interline flickering, field aliasing, and so on [1-3]. Moreover, as the screen size increases, the artifacts become more troublesome. Deinterlacing methods have been developed to minimize these issues.

Deinterlacing methods can be roughly classified into two categories. One is the intra-field spatial domain method that only uses one field [3-8] and the other is an inter-field temporal deinterlacing method that uses multiple fields. Inter-field temporal deinterlacing includes motion adaptive and motion compensation (MC) methods [9].

Intra-field deinterlacing methods are widely used in real-time application because of their simplicity. Intra-field deinterlacing methods have been proposed with different degrees of complexity and performance. The line average (LA) method [3] is a simple method that interpolates the pixel using an average value of the upper and lower pixels. LA has generally been used due to its clarity and simplicity in addition to the fact that it exhibits no motion artifacts. However, the vertical resolution of the input image is halved before the image is interpolated which causes a reduction in the amount of details in the progressive image.

In order to avoid this issue, Doyle proposed an edge-based line-average (ELA) [4] method, which uses edge directional correlation to interpolate a missing line between two adjacent lines in the interlaced images. ELA is a well-known edge-based intra-field deinterlacing method, which categorizes the edge direction as 45, 90,

and 135 degrees and then linearly interpolates in the chosen direction. ELA performs well with very little complexity and can be used to eliminate the blurring effect of LA in the regions where the edge is estimated correctly. However, this method has poor visual performance in high spatial frequency regions and can inaccurately estimate the edge detections, which leads to image degradation.

In an effort to remedy these deficiencies, numerous solutions based on the ELA algorithm have been proposed. The efficient ELA (EELA) [5] introduces two useful measurements for estimating the directional spatial correlation more efficiently. In addition, deinterlacing methods depending on content analysis [6] or the horizontal edge pattern [7] have been proposed. Kim proposed a modified ELA (MELA) [6]. In addition to the direction measures of EELA, MELA includes another vertical measure. Therefore, MELA considers three directional spatial correlations that cover vertical, anti-diagonal. The low complexity interpolation method for deinterlacing (LCID) [7] uses four directional differences that are simpler than those in MELA, which are diagonal, anti-diagonal, vertical and horizontal. Specifically, the direction-oriented interpolation (DOI) [8] method uses a window-based measure and is more robust and efficient when compared to the previous ELA methods. DOI can increase the resolution of direction and shows good performance for an image with local, single strong edges or horizontal edges. However, DOI has a high complexity for searching edge patterns within a large search range and a poor performance for regions with similar and repetitive edge patterns.

The aforementioned deinterlacing methods can be roughly separated into two classes. One is simple and highly efficient but with no consideration, for edge direction information or incorrect edge direction estimation which degrades the image detail. The other is complex with high computation overhead but has good edge direction estimation.

^a Corresponding author: jwang0608@xidian.edu.cn

In this paper, we proposed an intra-field deinterlacing algorithm using both closeness and similarity that successfully estimates edge direction. The missing pixel is estimated using using a nonlinear spatial average without degrading the edges by means of a nonlinear combination of nearby image values. Moreover, we combine gray levels based on their geometric closeness and their similarity, and prefer near values to distant values in the context of both closeness and similarity. The deinterlacing method using closeness and similarity (DCS) is based on the assumption of closeness and similarity between the neighborhood values where two pixels can be close to one another.

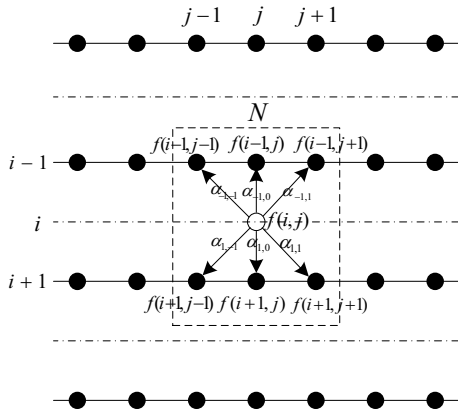


Fig. 1 Six neighbor pixels used for deinterlacing.

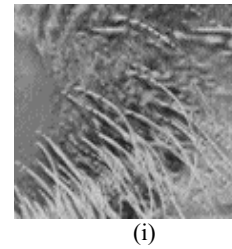
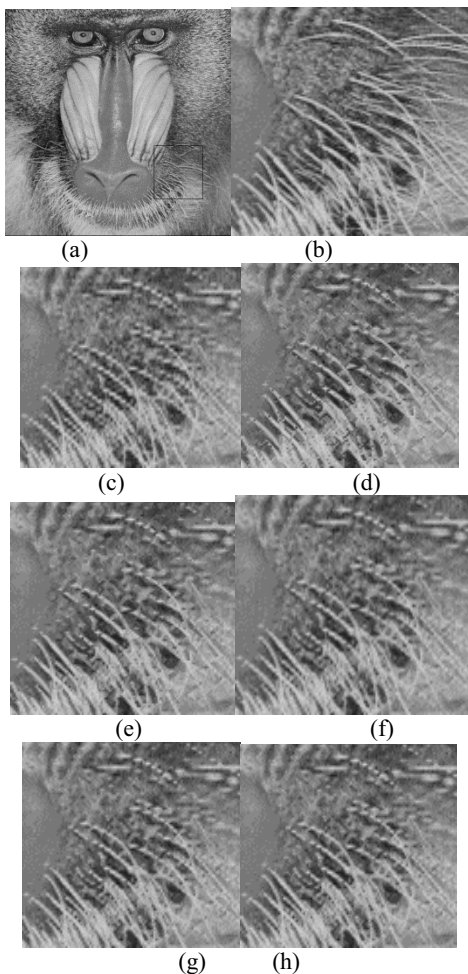


Figure 2. (a) The original image and (b) part of the original zoomed image. Perceived image quality comparison using various deinterlacing methods: (c) LA, (d) ELA, (e) EELA, (f) MELA, (g) LCID, (h) EMD, (i) proposed method.

2 The proposed algorithm

The pixel value at a given location is a function of the values in a small neighborhood at the same location. In the conventional deinterlacing method, the pixel values are averaged under the assumption that images typically vary gradually over space and neighbor pixels are likely to have similar values. The conventional deinterlacing method can be expressed as follows:

$$\hat{f}(i, j) = \frac{\sum_{(m,n) \in N} \alpha_{m,n} f(i+m, j+n)}{\sum_{(m,n) \in N} \alpha_{m,n}} \quad (1)$$

where N is the set of the neighbor pixel locations, m and n are vertical and horizontal displacement parameters, respectively, and $\alpha_{m,n}$ is the weight. Figure 1 shows an example of six neighbor pixels used in Equation (1) and LA is used as an example to explain Equation (1). For LA, we only use the pixels at the position $(i-1, j)$ and $(i+1, j)$, and the parameters $\alpha_{-1,0} = \alpha_{1,0} = 1$.

There are two drawbacks to the existing deinterlacing methods, and they are listed below:

- (1) The weight $\alpha_{m,n}$ is constant within some limited number of fixed candidate edge directions;
- (2) These methods only consider the similarity between pixels, and awareness of the closeness or spatial locality is not taken into account.

The assumption is that slow spatial variations fail at the edges, which are consequently blurred. To address the first drawback, an adaptive nonlinear function is required that averages the image values with weights, which decay with dissimilarity and also depend upon image intensity. Generally speaking, the neighbor pixels that are markedly different from the current interpolated pixel should be assigned a small weight. In contrast, the neighbor pixels most similar to the current interpolated pixel should have

Table 2. PSNR and CPU time for different images for the proposed and existing methods

Method	Image	Lena	Boat	Pepper	Finger	Baboon	Zelda	Man	Avg.
LA	PSNR(dB)	37.668	35.316	33.712	31.862	23.487	41.809	31.897	33.670
	CPU time (sec)	0.015	0.021	0.015	0.021	0.015	0.015	0.021	0.0175
ELA	PSNR(dB)	35.854	32.326	34.030	28.862	23.056	38.968	30.814	31.987
	CPU time (sec)	0.218	0.234	0.218	0.221	0.249	0.234	0.234	0.2297
EELA	PSNR(dB)	37.021	33.627	34.083	29.504	23.103	40.038	31.841	32.716
	CPU time (sec)	0.249	0.259	0.249	0.312	0.335	0.249	0.249	0.2717
MELA	PSNR(dB)	37.915	35.338	34.165	31.466	23.602	41.727	32.185	33.771
	CPU time (sec)	0.265	0.296	0.265	0.256	0.280	0.280	0.280	0.2745
LCID	PSNR(dB)	37.383	33.843	34.112	30.848	23.511	40.91	31.867	33.210
	CPU time (sec)	0.254	0.296	0.254	0.280	0.280	0.254	0.254	0.2674
EMD	PSNR(dB)	36.848	32.26	33.731	31.220	23.547	33.568	31.054	31.746
	CPU time (sec)	0.665	0.698	0.665	0.715	0.698	0.698	0.698	0.691
DCS	PSNR(dB)	37.986	35.598	34.393	31.929	24.048	42.006	32.183	34.021
	(Ψ_{GFS})								
	CPU time (sec)	0.078	0.089	0.079	0.091	0.091	0.078	0.078	0.0834

a large weight. Then, we can design an adaptive function that is a monotonic decreasing function of the absolute value of the difference in pixel value between the current interpolated pixel and the neighbor pixel, shown as Equation (2).

$$\alpha_{m,n} = g\left(\left|\hat{f}(i,j) - f(i+m,j+n)\right|\right) \quad (2)$$

where m and n are vertical and horizontal displacement parameters respectively, $m = 2k - 1, n = l, (k, l) \in \mathbf{Z}$, $\hat{f}(i, j)$ is the prediction of the current interpolated high resolution pixels obtained based on the average of the two nearest pixels or two pixels in the vertical direction. We choose $g(x)$ as

$$g_1(x) = \exp(-x^2) \quad (3)$$

Comparing the performance of several deinterlacing methods reveals that LA has a better quality peak signal to noise ration (PSNR) when compared to many of the edge-based methods with very low complexity. That is because the neighbor pixels used in LA are the shortest distance from the interpolated high resolution pixel, while, other methods just consider direction estimation, but ignore the spatial locality distance, or closeness, which

results in poor performance. In other words, closeness is also an essential concept. Thus, we designed another weight, $\beta_{m,n}$, which incorporates closeness by weighting pixel values with coefficients that fall off with distance. We can also design a monotonic decreasing function to assign a weight to the distance between the current interpolated pixel and the neighbor pixel. LA performs well when considering the nearest neighbor pixels. However, direction estimation is also an important issue due to edge regions and LA causes some artifacts because direction is not considered. In order to solve all of the above-mentioned problems, we propose a DCS method that consider closeness and similarity in both the smooth and edge regions.

$$\hat{f}(i, j) = \frac{\sum_{(m,n) \in N} \alpha_{m,n} \beta_{m,n} f(i+m, j+n)}{\sum_{(m,n) \in N} \alpha_{m,n} \beta_{m,n}} \quad (4)$$

$$\alpha_{m,n} = \exp\left(\frac{-\left(\hat{f}(i, j) - f(i+m, j+n)\right)^2}{2\sigma_s^2}\right) \quad (5)$$

$$\beta_{m,n} = \exp\left(\frac{-(m^2 + n^2)}{2\sigma_c^2}\right) \quad (6)$$

Here, σ_s and σ_c are regularization factors for $\alpha_{m,n}$ and $\beta_{m,n}$, respectively, used to control the performance of the algorithm and they are generally selected empirically.

3 Experiments

A simulation was performed using standard test 512×512 images Lena, Boat, Pepper, Finger, Baboon, Zelda, and Man in order to verify the superiority of the proposed method compared to the existing algorithms (LA, ELA, EELA, MELA, LCID, and EMD). We assessed the objective performance in terms of PSNR and CPU time in addition to the subjective performance, which is a widely adopted criterion in the literature. Our experiment was carried out on an Intel Core 2 Duo CPU E8500 @ 3.16 GHz with s set to 1.

our proposed algorithm can get good PSNR with fixed factor set. In terms of CPU time, our algorithm required only 30% of the complexity of other MELAs. Except for the simplest method (LA), our method required the least CPU time. For the subjective performance evaluation in terms of the visual effect of the resulting images, we also show part of the perceived image quality of Baboon in Fig. 2. It is apparent from Fig. 2 that the proposed method results in an improved visual quality in terms of the details of the fur compared to the other methods. These results indicate that the proposed method is superior to other intra-field deinterlacing methods in terms of objective and subjective qualities and has low complexity.

Acknowledgement

This work was partly supported by the National Natural Science Foundation of China (No. 61501359); this work was partly supported by the Fundamental Research Funds for the Central Universities (No. JB160504).

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, *Deinterlacing: 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," *Opt. Eng.* 39(8), 2101–2105 (2000).
6. W. Kim, S. Jin, and J. Jeong, "Novel intra deinterlacing algorithm using content adaptive interpolation," *IEEE Trans. Cons. Elect* 53(3), 1036–1043 (2007).
7. P.-Y. Chen and Y.-H. Lai, "A low-complexity interpolation method for deinterlacing," *IEICE Trans. Inf. Syst.* E90-D(2), 606–608 (2007).
8. H. Yoo and J. Jeong, "Direction-oriented interpolation and its application to deinterlacing," *IEEE Trans. Consum. Electron* 48(4), 954–962 (2002).
9. 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).
10. D. Gellies, M. Plantholt, and D. Westerkamp, "Motion adaptive field rate upconversion algorithms for 900 lines/100 Hz 2:1 displays," *IEEE Trans. Consum. Electron.*, 36(5), 149–160 (1990).
11. Y. Wang and S. K. Mitra, "Motion/pattern adaptive interpolation of interlaced video sequences," *IEEE Trans. Acoust., Speech, Signal Process.*, 2829–2832 (1991).
12. D. Hargreaves and J. Vaisey, "Bayesian motion estimation and interpolation in the interlaced video sequences," *IEEE Trans. Image Process.*, 6(5), 764–769 (1997).
13. O. Kwon, K. Sohn, and C. Lee, "Deinterlacing using directional interpolation and motion compensation," in Proc. Image and Video Communications and Processing 2003, 604–611 (2003).
14. T. Jeong, Y. Kim, K. Sohn, and C. Lee, "Deinterlacing with selective motion compensation," *SPIE Optical Engineering*, 45(07), 077001 (2006).
15. Y. D. Kim, J. Chang, G. S. Shin, and M. G. Kang, "Interlaced-to-progressive conversion using adaptive projection-based global and representative local motion estimation," *Journal of Electronic Imaging*, 17(2), 023008 (2008).