International Journal of Software Engineering and Its Applications
Vol.8, No.2 (2014), pp.41-50
http://dx.doi.org/10.14257/ijseia.2014.8.2.05

Adaptive Interpolation Algorithm Considering Total Edge Directions for Effective Deinterlacing

Jongho Kim

Dept. of Multimedia Engineering, Sunchon National University
255 Jungangno, Suncheon, Jeonnam 540-950, Korea
jhkim@sunchon.ac.kr

Abstract

This paper presents an adaptive interpolation algorithm for effective picture format conversion, namely deinterlacing based on total edge directions estimation. The deinterlacing algorithms are generally made up of two parts, such as the estimation of edge directions and the interpolation of missing pixels along with the estimated direction. The proposed algorithm adopts and enhances the block-based search concept for estimation of edge directions, which is introduced in the direction-oriented interpolation (DOI) algorithm. In this paper the edge directions are estimated with the vertical angles as well as the horizontal angles in the DOI and then the missing pixels are interpolated by using median operation. Experimental results show that the proposed algorithm can estimate edge directions and interpolate missing pixels more accurately compared with the conventional deinterlacing algorithms in terms of the objective performance measured in PSNR and the subjective visual quality.

Keywords: *edge direction estimation; deinterlacing; image interpolation; median operation; postprocessing*

1. Introduction

The inherent nature of the interlaced scanning process of current TV systems often leads uncomfortable visual artifacts such as the edge and interline flickering or line crawling. The picture format conversion from an interlaced scan to a progressive one, namely deinterlacing, has been widely adopted in many applications, in order to reduce the visual defects due to those artifacts [1-3]. The goal of deinterlacing is to reconstruct the missing lines of pixels in interlaced images, reducing aliasing, and increase the vertical resolution of the images. As the number of digital video formats used for broadcasting, Internet video, and multimedia terminals has been increasing rapidly during the last few years, the demand for video format conversion is growing significantly. High quality deinterlacing is, therefore, becoming an important issue in the field of video processing.

Recently, a number of deinterlacing techniques have been proposed in [4-20]. They can be classified into two categories including the spatial domain methods in [4-16] which use only one field and the temporal domain methods in [17-20] which use multiple fields. In general, the performance of the temporal domain methods is better than that of the spatial domain methods. However, the spatial domain methods are less complicated than temporal domain methods. Moreover, the temporal domain methods usually include the spatial domain methods as an initial or fundamental part. Thus the enhanced image interpolation in the spatial domain improve performance is still required to improve overall performance of deinterlacing.

ISSN: 1738-9984 IJSEIA

In particular, many spatial interpolation techniques have been proposed to improve the quality of interpolated images by enhancing the edges and the overall image sharpness. For example, the Bob method is the primary intralinear filter used in the TV industry for deinterlacing [4]. Doyle proposed an edge-based line averaging (ELA) algorithm, which is a well-known intra-nonlinear filter that uses edge directional correlations to linearly interpolate an empty pixel between two adjoining lines of the even or odd field [5]. Chen improved the ELA method by estimation of edge direction more accurately and proposed the efficient ELA (EELA) in [6]. Kim and Jeong proposed the modified ELA method. Yoo proposed the direction-oriented interpolation (DOI) algorithm which uses two spatial direction vectors (SDVs) to acquire a more appropriate edge direction in [7]. Park introduced a combination of horizontal and vertical edge patterns for an edge-dependent interpolation algorithm in [8, 9]. Chen proposed a low-complexity intra-nonlinear filter in which edge directions were determined in diagonal, horizontal, and vertical directions in [10]. Lee introduced an intra-nonlinear filter that detects edge information and direction based on locally adaptive thresholded binary images, where the adaptive window size was used to efficiently detect the edge slope in [11]. Alternatives employ a nonlinear and soft computing-based approach, which includes fuzzy logic for image interpolation, specifically fuzzy detection of edge-direction based interpolation (FDED) in [12] and fuzzy direction-oriented interpolation (FDOI) in [13].

In this paper, we propose an adaptive interpolation algorithm for effective deinterlacing which considers the edge directions with steep angles to utilize the advanced SDVs and the median operation to improve the DOI. The advanced SDVs describe the edge direction in higher resolution and accuracy and the median operation removes the estimation errors so that the interpolated image shows good visual quality and objective performance. The rest of the paper is organized as follows. Section 2 briefly reviews the related algorithms used to devise the proposed algorithm. Section 3 describes the proposed algorithm in detail. Finally, the experimental results and conclusions are presented in Sections 4 and 5, respectively.

2. Conventional Deinterlacing Algorithms

We examine two related deinterlacing algorithms, including the ELA and DOI algorithms, in this section that form the basis of the proposed algorithm.

2.1. The ELA and EELA Algorithms

The ELA algorithm, which is a well-known interpolation in the spatial domain, utilizes directional correlation among pixels to linearly interpolate a missing line between two adjacent lines [5]. If the edge direction is estimated correctly by using (1), the method gives a good result. Otherwise, it produces an undesirable result.

$$\begin{cases} a = \left| x(i-1,\ j-1) - x(i+1,\ j+1) \right| \\ b = \left| x(i-1,\ j) - x(i+1,\ j) \right| \\ c = \left| x(i-1,\ j+1) - x(i+1,\ j-1) \right| \end{cases} \tag{1}$$

where a, c, and b indicate the edge direction from left to right (diagonal), the edge direction from right to left (anti-diagonal), and the vertical direction, respectively. The current pixel $x(i, j)$ is then interpolated by

$$x(i,j)=\begin{cases}\left(x(i-1,j-1)+x(i+1,j+1)\right)/2, & \text{if } \min(a,b,c)=a\\ \left(x(i-1,j)+x(i+1,j)\right)/2, & \text{if } \min(a,b,c)=b\\ \left(x(i-1,j+1)+x(i+1,j-1)\right)/2, & \text{if } \min(a,b,c)=c.\end{cases} \tag{2}$$

The key to the success of the ELA method is an accurate estimation of edge direction. However, the pixel-based approach such as ELA produces unpleasant results, specifically around edges, due to noise, variation of intensity, and a weak edge. In order to increase the probability of detecting a reliable edge direction, the EELA (efficient ELA) algorithm was developed in [6] by introducing two useful measurements within the operation window to alleviate misleading decision in determining the direction where the interpolation is to be made. We define P and Q as follows to improve the directional estimation between two adjacent lines.

$$\begin{cases}P=\left|x(i-1,j-1)-x(i+1,j)\right|+\left|x(i-1,j)-x(i+1,j+1)\right|\\ Q=\left|x(i-1,j+1)-x(i+1,j)\right|+\left|x(i-1,j)-x(i+1,j-1)\right|\end{cases} \tag{3}$$

where P and Q represent the edge direction from left to right (diagonal) and the edge direction from right to left (anti-diagonal), respectively. Then, the EELA algorithm can be defined as

$$x(i,j)=\begin{cases}\left(x(i-1,j-1)+x(i+1,j+1)\right)/2, & \text{if } \begin{cases}P<Q \text{ and } \min(a,b)=a\\ P=Q \text{ and } \min(a,b,c)=a\end{cases}\\ \left(x(i-1,j)+x(i+1,j)\right)/2, & \text{if } \begin{cases}P<Q \text{ and } \min(a,b)=b\\ P=Q \text{ and } \min(a,b,c)=b\\ P>Q \text{ and } \min(b,c)=b\end{cases}\\ \left(x(i-1,j+1)+x(i+1,j-1)\right)/2, & \text{if } \begin{cases}P>Q \text{ and } \min(b,c)=c\\ P=Q \text{ and } \min(a,b,c)=c.\end{cases}\end{cases} \tag{4}$$

The EELA algorithm defined in (4) are very similar to the ELA method, but P and Q can find edge directions more accurately than the ELA method.

2.2. The DOI Algorithm

The DOI algorithm reported in [7] interpolates the missing pixels using two upper reference lines and two lower reference lines. By using the SDVs, it reduces possibility of wrong decision for the highest-correlated spatial direction. The line-based directional interpolation is carried out after the SDVs are found. The SDVs that estimate the edge directions for the upper line (SDV_u) and the lower line (SDV_l) to obtain more accurate direction of the highest spatial correlation are defined by

$$\begin{cases}SDV_u=\arg\min\left(D_u(k)\right), & -R\le k\le R\\ SDV_l=\arg\min\left(D_l(k)\right), & -R\le k\le R\end{cases} \tag{5}$$

where

$$\begin{cases} D_u(k) = \sum_{n=-1}^{1} \left(\left| x(i-1, j+n) - x(i-3, j+n+k) \right|^2 + \left| x(i+1, j+n) - x(i-1, j+n+k) \right|^2 \right) \\ D_l(k) = \sum_{n=-1}^{1} \left(\left| x(i+1, j+n) - x(i+3, j+n+k) \right|^2 + \left| x(i-1, j+n) - x(i+1, j+n+k) \right|^2 \right) \end{cases} \quad (6)$$

and the search range, R, is set to 16 in this paper. The correct edge estimation can be found that the magnitudes of SDV_u and SDV_l are nearly the same, while the directions (i.e., signs) are opposite. The sum of SDV_u and SDV_l is close to zero, and thus, it is a good indicator as to whether the current pixel to be interpolated is on the edge region or not. The current pixel, $x(i, j)$ is interpolated by $x(i, j) = (x(i-1, j + SDV_u/2) + x(i+1, j + SDV_l/2))/2$ in the edge region, and $x(i, j) = (x(i-1, j) + x(i+1, j))/2$ in the non-edge region.

Table 1. SDVs and the Corresponding Direction Angles

SDV	0	±1	±2	±3	…	±SDV
Angle (deg.)	90	±45	±27	±18	…	$\pm\tan^{-1}(1/SDV)$

Although the DOI algorithm can estimate the edge directions better than the ELA and EELA algorithms, it mainly considers perpendicular (90 deg.) and horizontal angle (HA) edge directions, but does not consider vertical angle (VA) edge directions. The relationship between the edge direction angles and the SDVs are shown in Table 1.

3. Proposed Algorithm

The deinterlacing algorithms generally consist of two parts: the edge directions must be estimated first, and then the missing pixels are interpolated along with the estimated edge directions. A number of methods to find the edge directions or to interpolate missing pixels between two adjacent lines have been reported. Among them, the DOI algorithm does not consider the edge directions with vertical high angles even though it estimates the edge directions better than the earlier methods such as ELA and EELA, and thus it often leads interpolation artifacts around edges specifically the edges with high angles. In order to overcome these problems, we focus on estimating the edge directions of vertical angles more accurately and reducing artifacts based on the median operation to improve the interpolated image quality.

It is possible to find the edge direction of horizontal angles by using horizontal direction vectors (HDVs) in the spatial domain which are introduced in the DOI method. It is also possible to find the edge direction of vertical angles by using vertical direction vectors (VDVs) in the spatial domain proposed in this paper. The HDVs and VDVs that estimate the edge directions are defined as follows.

$$\begin{cases} HDV_u = \arg\min\left(DH_u(p)\right), & -R \le p \le R \\ HDV_l = \arg\min\left(DH_l(p)\right), & -R \le p \le R \\ VDV_u = \arg\min\left(DV_u(p)\right), & -R \le p \le R,\ p: odd\ number \\ VDV_l = \arg\min\left(DV_l(p)\right), & -R \le p \le R,\ p: odd\ number \end{cases} \quad (7)$$

where

$$\begin{cases} DH_u(p) = \sum_{n=-1}^{1}\left(\left|x(i-1, j+n) - x(i-3, j+n+p)\right|^2 + \left|x(i+1, j+n) - x(i-1, j+n+p)\right|^2\right) \\ DH_l(p) = \sum_{n=-1}^{1}\left(\left|x(i+1, j+n) - x(i+3, j+n+p)\right|^2 + \left|x(i-1, j+n) - x(i+1, j+n+p)\right|^2\right) \\ DV_u(p) = \sum_{n=-1}^{1}\left(\left|x(i-1, j+n) - x(i-1+p, j+1+n)\right|^2 + \left|x(i+1, j+n) - x(i+1+p, j+1+n)\right|^2\right) \\ DV_l(p) = \sum_{n=-1}^{1}\left(\left|x(i-1, j+n) - x(i-1+p, j-1+n)\right|^2 + \left|x(i+1, j+n) - x(i+1+p, j-1+n)\right|^2\right). \end{cases} \quad (8)$$

In (7) and (8), the HDVs are obtained by referencing upper two rows and lower two rows, i.e., i-3, i-1, i+1, and i+3, and the VDVs are obtained by referencing five columns, i.e., j-2, j-1, j, j+1, and j+2. Moreover, the search range, R, is set to 16 in both cases of horizontal and vertical directions. For the horizontal direction, we can calculate $DH_u(p)$ and $DH_l(p)$ at 33 points (from -16 to +16), and for the vertical direction, we can calculate $DV_u(p)$ and $DV_l(p)$ at 14 points due to the interlaced scan and the same DV as DH for p = 1.

The MSE (mean squared error) cost functions defined in (8) are used to find the HDVs and VDVs. For horizontal edge regions, the magnitudes of HDV_u and HDV_l are very similar to each other, while the directions (i.e., signs) are opposite. Thus the sum of HDV_u and HDV_l is close to zero. In the same manner, for vertical edge regions, VDV_u and VDV_l are applied to the case of the vertical edge regions. In non-edge or texture regions, the sum of HDV_u and HDV_l or the sum of VDV_u and VDV_l tends to be a large value. The sum is, therefore, a good measurement to be used to determine whether the current pixel to be interpolated is on the edge region or not. Figure 1 presents the HDVs, VDVs, and reference lines for them.

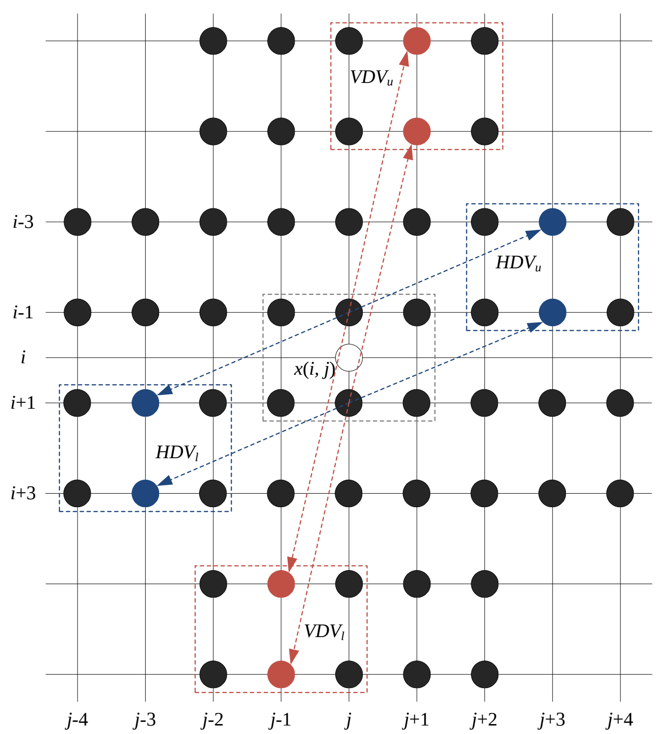

Figure 1. HDVs, VDVs, and Reference Lines for the Proposed Algorithm

To interpolate the current pixel, $x(i, j)$, we define the HDVs- and VDVs-related values as in (9) and (10).

$$\begin{cases} a = x(i-3,\ j+HDV_u) \\ b = x(i+1,\ j+HDV_l) \\ c = x(i-1,\ j+HDV_u) \\ d = x(i+3,\ j+HDV_l) \end{cases} \tag{9}$$

where a and c are the pixels chosen by HDV_u and b and d are the pixels chosen by HDV_l. Similarly, e and g are the pixels chosen by VDV_u and f and h are the pixels chosen by VDV_l in (10).

$$\begin{cases} e = x(i-1+VDV_u,\ j+1) \\ f = x(i-1+VDV_l,\ j-1) \\ g = x(i+1+VDV_u,\ j+1) \\ h = x(i+1+VDV_l,\ j-1). \end{cases} \tag{10}$$

The index (i, j) in (7) - (10) is the current pixel position to be interpolated. Given the current line and nine reference lines, i-3, i-1, i+1, i+3, j-2, j-1, j, j+1, and j+2, the magnitude of the difference between $x(i\text{-}1, j)$ and $x(i\text{+}1, j)$ is first compared with a threshold T (=10), and $x(i, j)$ is selected as the average in the vertical direction. Otherwise, the median filtering is applied to a, b, c, d, e, f, and g as shown in the flow chart in Figure 2.

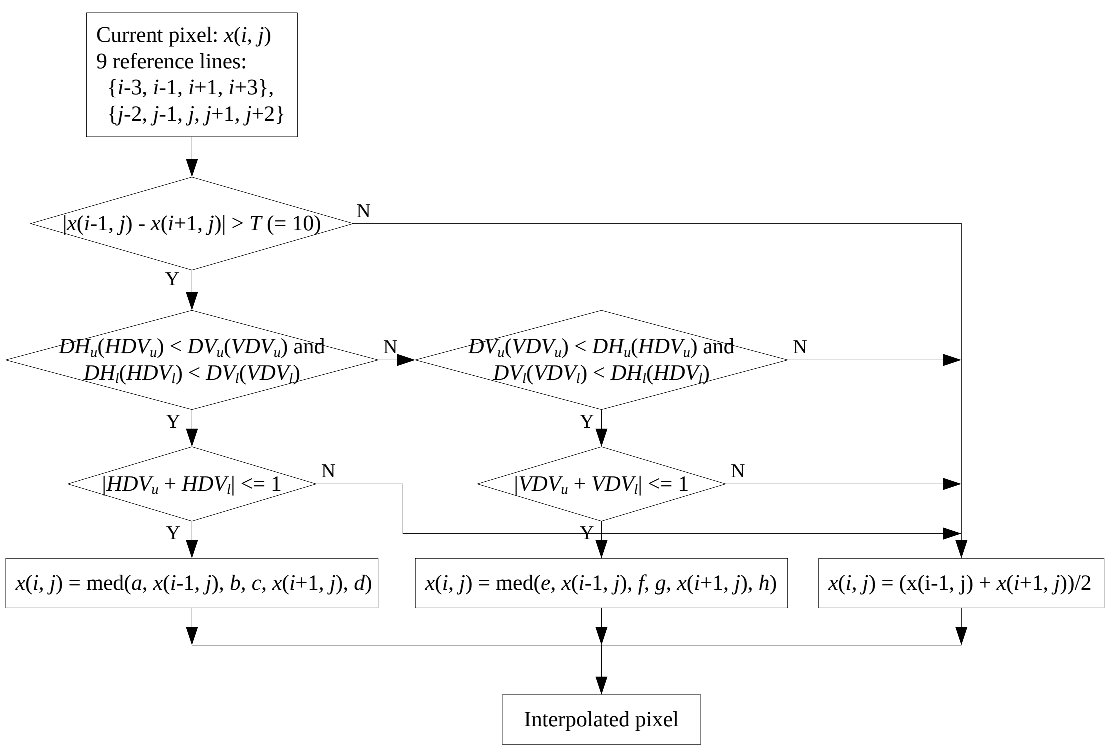

Figure 2. Flow Diagram of the Proposed Algorithm

In Figure 2, med(·) represents the median operation, which sorts 6 values and averages two median values. Table 2 shows the relationship between direction angles and VDVs. Note that the DOI method has only HDVs and it lacks directional resolution.

Table 2. VDVs and the Corresponding Direction Angles

VDV	±3	±5	±7	±9	…	±VDV
Degree	±72	±79	±82	±84	…	$\pm\tan^{-1}$(VDV)

4. Experimental Results

The performance of the proposed algorithm is evaluated in terms of the objective performance measured in PSNR and the subjective visual quality. The experiments are carried out with seven test images, including various edge patterns. The size of the test images is 512×512. These images are simulated after down-sampling, that is, a 512×256 image was made by deleting even lines from the original images. The proposed interpolation algorithm is applied to the pixels on odd lines of the down-sampled images.

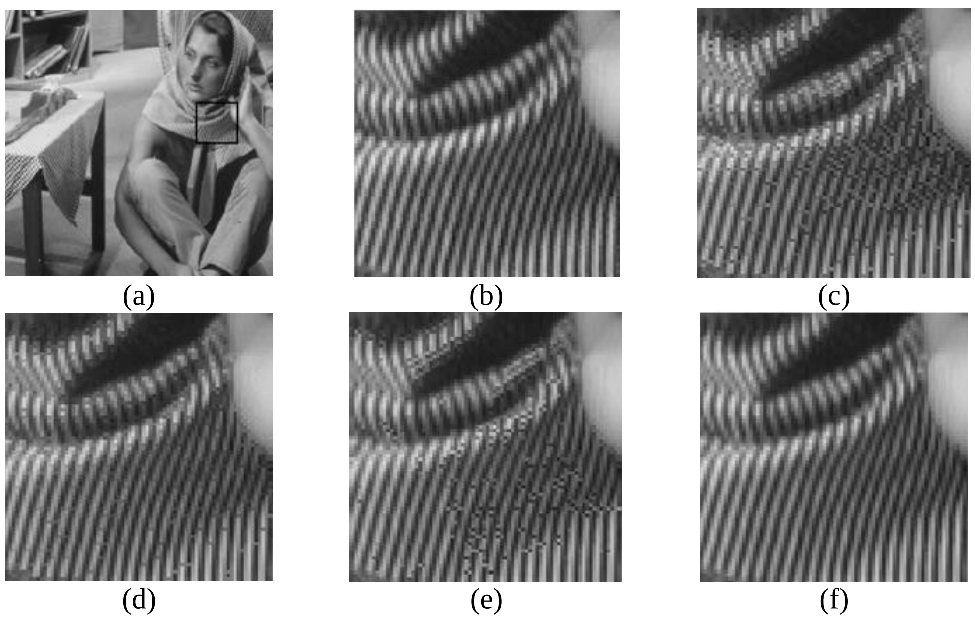

Figure 3. Subjective Visual Quality Comparison on *Barbara*. (a) Original Image, (b) Original Image in Part with 5 Times Magnified, (c) ELA, (d) EELA, (e) DOI, (f) Proposed Algorithm

The interpolated images resulting from the application of various methods are shown in Figure 3. The proposed method shows better visual quality than other methods. For the regions of vertical angle directions, the conventional methods suffer from artifacts, resulting in unsatisfactory image quality. The proposed interpolation method, however, does not produce artifacts, and thus provides satisfactory image quality.

The results of various deinterlacing methods including the proposed algorithm are presented in Table 3 for the various test images. The proposed algorithm indicates a substantial improvement of performance compared to conventional algorithms in terms of PSNR as shown in the table. The proposed method increased PSNR up to 2.89dB for *Barbara*, and up to 2.58dB for *Finger*, compared with the DOI method. It shows that the proposed algorithm presents better performance in PSNR for more complicated images due to the nature of the proposed algorithm.

Table 3. Performance Evaluation Results Measured in PSNR (dB)

Image	ELA	EELA	DOI	Proposed
Airplane	31.15	31.30	31.71	31.86
Baboon	22.99	23.18	23.43	23.62
Barbara	25.20	30.62	29.58	32.47
Finger	28.94	29.41	29.30	31.88
Zelda	34.05	34.18	33.93	34.25

5. Conclusions

In this paper, we proposed an adaptive interpolation algorithm for effective picture format conversion technique, namely deinterlacing based on total edge directions estimation, i.e., horizontal and vertical angles, and a median operation. By using vertical direction vectors (VDVs), the edge directions of vertical angles can be estimated more accurately than the conventional algorithms. Moreover, some errors and artifacts can be reduced by using a median operation. Experimental results demonstrate that the proposed algorithm is superior to the conventional deinterlacing algorithms in both the objective performance measured in PSNR and subjective image quality.

References

[1] G. Haan and E. B. Bellers, "Deinterlacing – an overview", Proceedings of the IEEE, vol. 86, no. 1839, **(1998)**.

[2] G. Jeon, "Artifact assessment approaches for images", Int. J. Multimedia and Ubiquitous Engineering, vol. 8, no. 327, **(2013)**.

[3] S. Park, "GOP level rate-control for real-time video transmission", Int. J. Multimedia and Ubiquitous Engineering, vol. 8, no. 207, **(2013)**.

[4] K. Sugiyama and H. Nakamura, "A method of deinterlacing with motion compensated interpolation", IEEE Trans. Consumer Electronics, vol. 45, no. 611, **(2000)**.

[5] T. Doyle, "Interlaced to sequential conversion for HDTV applications", Proceedings of the 2nd International Workshop on Signal Processing of HDTV, Aquila, Italy, **(1988)** February 29-March 2.

[6] T. Chen, H. R. Wu and Z. H. Yu, "Efficient deinterlacing algorithm using edge-based line average interpolation", Optical Engineering, vol. 39, no. 2101, **(2000)**.

[7] H. Yoo and J. Jeong, "Direction-oriented interpolation and its application to de-interlacing", IEEE Trans. Consumer Electronics, vol. 48, no. 954, **(2002)**.

[8] M. K. Park, M. G. Kang, K. Nam and S. G. Oh, "New edge dependent deinterlacing algorithm based on horizontal edge pattern," IEEE Trans. Consumer Electronics, vol. 49, no. 1508, **(2003)**.

[9] M. K. Park, M. Byun, J. H. Park and M. G. Kang, "Edge-dependent interpolation-based deinterlacing using edge patterns", J. Electronic Imaging, vol. 15, no. 043003, **(2006)**.

[10] P.-Y. Chen and Y.-H. Lai, "A low-complexity interpolation method for deinterlacing", IEICE Trans. Information and Systems, E90-D, no. 2, no. 606, **(2007)**.

[11] D.-H. Lee, "A new edge-based intra-field interpolation method for deinterlacing using locally adaptive-thresholded binary image", IEEE Trans. Consumer Electronics, vol. 54, no. 110, **(2008)**.

[12] F. Michaud, C. T. Le Dinh and G. Lachiver, "Fuzzy detection of edge-direction for video line doubling", IEEE Trans. Circuits and Systems for Video Technology, vol. 7, no. 539, **(1997)**.

[13] G. Jeon and J. Jeong, "Fuzzy rule and Bayesian network based line interpolation for video deinterlacing", IEICE Trans. Communications, E90-B(6), vol. 1495, **(2007)**.

[14] M.-H. Lee, J.-H. Kim, K.-K. Ryo and D.-I. Song, "A new algorithm for interlaced to progressive scan conversion based on directional correlation and its IC design", IEEE Trans. Consumer Electronics, vol. 40, no. 119, **(1994).**

[15] C. J. Kuo, C. Liao and C. C. Lin, "Adaptive interpolation technique for scanning rate conversion", IEEE Trans. Circuits and Systems for Video Technology, vol. 6, no. 317, **(1996)**.
[16] T. Chen, H. R. Wu and Z. H. Yu, "An efficient edge line average interpolation algorithm for deinterlacing", Proceedings of SPIE Visual Communications and Image Processing 2000, Perth, Australia, **(2000)** June.
[17] H. Oh, Y. Kim, Y. Jung, A. W. Morales and S. Ko, "Spatio-temporal edge-based median filtering for deinterlacing", Proceedings of IEEE International Conference on Consumer Electronics 2000, Los Angeles, CA, **(2000)** Jun. 13-15.
[18] M. Chen, C. Huang and C. Hsu, "Efficient de-interlacing technique by inter-field information", IEEE Trans. Consumer Electronics, vol. 50, no. 1202, **(2004)**.
[19] K. Sugiyama, Y. Yamada and N. Sagara, "Improvement of motion compensated inter-field interpolation method for deinterlacing", Proceedings of 2006 IEEE Region 10 Conference (TENCON 2006), Hong Kong, China, **(2006)** November 14-17.
[20] B. Choi, J. Han, C. Kim and S. Ko, "Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation", IEEE Trans. Circuits and Systems for Video Technology, vol. 17, no. 407, **(2007)**.

Author

Jongho Kim, received B.S., M.S., and Ph.D. degrees in electronic communications engineering from Hanyang University, Seoul, Korea, in 1998, 2000, and 2008, respectively. From 2008 to 2009, he worked at Samsung Electronics Company, Korea. He is currently a Professor at Dept. of Multimedia Engineering, Sunchon National University. His research interests include image/video coding and multimedia signal processing.