

Edge-adaptive Spatial Video De-interlacing Algorithms Based on Fuzzy Logic

Piedad Brox, Iluminada Baturone, Santiago Sánchez-Solano, and Julio Gutiérrez-Ríos

Abstract — *Since the human visual system is especially sensitive to image edges, edge-dependent spatial interpolators have been proposed in literature as a means of successfully restoring edges while avoiding the staircase effect of linear spatial algorithms. This paper addresses the application of video de-interlacing, which constitutes an indispensable stage in video format conversion. Classic edge-adaptive de-interlacing algorithms introduce annoying artifacts when the edge directions are evaluated incorrectly. This paper presents two ways of exploiting fuzzy reasoning to reinforce edges without an excessive increase in computational complexity. The performance of the proposed algorithms is analyzed by de-interlacing a wide set of test sequences. The study compares the two proposals both with each other and with other edge-adaptive de-interlacing methods reported in the recent literature¹.*

Index Terms — Video Signal Processing, De-interlacing, Fuzzy Inference Systems, Edge adaptive.

I. INTRODUCTION

Fuzzy logic-based algorithms are able to model uncertainty and subjective concepts in video and image processing. Edges, for instance, are key concepts for improving the visual perception of an image. But deciding whether a pixel belongs to an edge or not is no simple task, especially if the images are noisy and/or contain a high number of details. Fuzzy logic theory provides a mathematical framework to deal with this kind of uncertain information.

This study focuses on the development of fuzzy logic-based algorithms for the particular task of video processing called ‘de-interlacing’ [1]. Current digital television transmission formats use an interlaced scan mode. For example, in the popular HDTV (high-definition television) 1080i the ‘i’ stands for interlaced scanning. Interlaced scanning is directly compatible with some CRT-based HDTV television sets on which video can be displayed natively in interlaced form, but for display on modern progressive-scan LCD and plasma TV sets, video must be de-interlaced and often scaled to the display resolution.

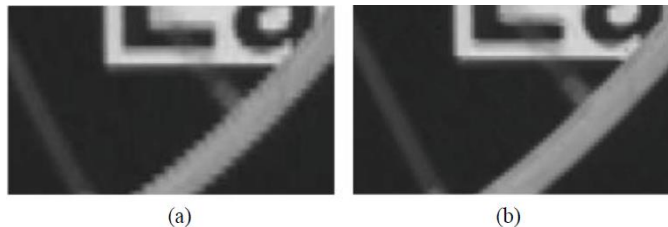


Fig. 1. De-interlaced results: (a) linear versus (b) edge-adaptive algorithm.

Fig. 1 illustrates the advantages of edge-adaptive de-interlacing algorithms. Linear spatial interpolation techniques such as Line Average produce an annoying effect known as the staircase effect (see Fig. 1(a)), but edges are correctly restored when the image is interpolated with a well-known edge-adaptive algorithm called ELA (Edge-based Line Average) [2] (see Fig. 1(b)).

The original version of the ELA algorithm works with 3+3 pixels from the upper and lower lines. ELA performs well when the edge direction agrees with the minimum differences of the local luminance values in the neighborhood, but otherwise introduces errors and degrades image quality. The errors usually appear when the edges are not clear, when the image is corrupted by noise, when there is a high number of details, etc. Such situations are described in this paper as areas of the picture with ‘unclear edges’.

Several proposals have been presented in literature to reduce the abovementioned ELA shortcomings [3]-[15]. The algorithm described by Chen *et al* [3] introduced two error measurements to reinforce the direction in which interpolation was being

¹ This work was partially supported by TEC2011-24319 and INNPACTO IPT-2012-0695-390000 projects from the Spanish Government (both with support from FEDER).

P. Brox is with the Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC/University of Seville, and with the Department of Electronics and Electromagnetism of the University of Seville, Spain, (e-mail: brox@imse-cnm.csic.es).

I. Baturone is with the Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC/University of Seville, and with the Department of Electronics and Electromagnetism of the University of Seville, Spain, (e-mail: lumi@imse-cnm.csic.es).

S. Sánchez-Solano is with the Instituto de Microelectrónica de Sevilla (IMSE-CNM), CSIC/University of Seville, Spain (e-mail: santiago@imse-cnm.csic.es).

J. Gutiérrez-Ríos is with the Department of Photonics Technology at the Technical University of Madrid (UPM), Spain. (e-mail: jgr@fi.upm.es).

performed. A direction-oriented spatial interpolator that used four edge orientations to increase ELA performance was presented by Chen and Lai [4]. Other approaches are focused on enhancing the reconstruction of horizontal edges. Lee *et al* [5] combined line doubling and an edge-adaptive ELA-based de-interlacing algorithm to interpolate horizontal edges. Kuo *et al* [6] proposed an edge-based line average which used the 3+3 pixels of ELA and the previously calculated interpolated pixel value. The algorithm presented by Lee *et al* [7] involved a modified ELA module and a contrast enhancement module. The ELA module used a processing window with 5+5 pixels whereas the second module reduced detail losses due to the interpolation. The neighborhood was enlarged up to 6+6 taps [8], 7+7 taps [9], [10], 11+11 taps [11], and 34+34 taps [12] in other proposals. A higher number of pixels in the aperture provides more information about edges, but at the expense of increasing the algorithm complexity. The algorithm proposed by Kim *et al* [13] considered three steps: pre-processing, content classification, and adaptive interpolation. After analysis of the local region in the pre-processing stage, the pixel was classified into four categories. Different interpolation strategies were then applied depending on the classification results. This strategy made it possible to study a high number of possible edge directions. Park *et al* [14] used not only luminance differences in pixel values but also a horizontal edge pattern to enhance edge interpolation. In the method proposed by Lee [15], edge direction and slope were detected with a locally adaptive-thresholded binary image.

Fuzzy logic-based approaches to video de-interlacing have also been reported in recent years [16]-[22]. Mancuso *et al* [16] used soft-decision fuzzy logic techniques to remove noise and de-interlace TV video signals. A fuzzy edge-direction detector was introduced by Michaud *et al* [17] to orient a conventional vertico-temporal filter for de-interlacing tasks. Van De Ville *et al* [18] used fuzzy inference rule bases to detect motion and to choose the most appropriate de-interlacing strategy according to the degree of motion detected. Brox *et al* [19] used two fuzzy logic-based systems, each one tackling a feature highly relevant to de-interlacing: motion and edges. More recently, several proposals have been reported by Jeon *et al* in which fuzzy techniques were used to de-interlace images by enhancing edges. The first of them presented a weighted fuzzy filter algorithm that computed weights using a fuzzy measurement from luminance differences in pixel values [20]. A new version of a fuzzy weighted filter capable of detecting six different edge directions was also considered [21]. The combination of a fuzzy rule-based de-interlacing algorithm and a rough sets-assisted optimization algorithm [22], as well as the mixture of spatio-temporal fuzzy filters [23] were also proposed to de-interlace video. The high computational costs involved in these approaches are not feasible for real-time processing.

Spatial de-interlacers often form part of more complex de-interlacing algorithms combining spatio-temporal interpolation. Improvements to the spatial de-interlacer therefore increase the overall performance of the spatio-temporal algorithm. This paper focuses on spatial edge-adaptive de-interlacing algorithms which do not significantly increase computational complexity. ELA-based algorithms that seek out edges in a 5+5 spatial window offer a good trade-off between complexity and performance, so the algorithms proposed in this paper are based on an ELA 5+5 scheme. To limit the shortcomings of conventional ELA, they apply simple fuzzy reasoning to ensure the presence of edges. Hence their denomination as ‘Fuzzy-ELA 5+5’ algorithms. The fuzzy reasoning rule bases were designed taking linguistically expressed, heuristic knowledge as a point of departure, thus exploiting the ability of fuzzy logic to cope with symbolic knowledge. Two ways of defining the rules employing different heuristics are proposed.

The paper is organized as follows. Section 2 describes two proposals called ‘Fuzzy-ELA 5+5 H1 (Heuristics1)’ and ‘Fuzzy-ELA 5+5 H2 (Heuristics2)’. Section 3 describes the performance of the algorithms when de-interlacing a wide battery of test sequences, in comparison with other state-of-the-art spatial de-interlacers. Finally, some conclusions are expounded in Section 4.

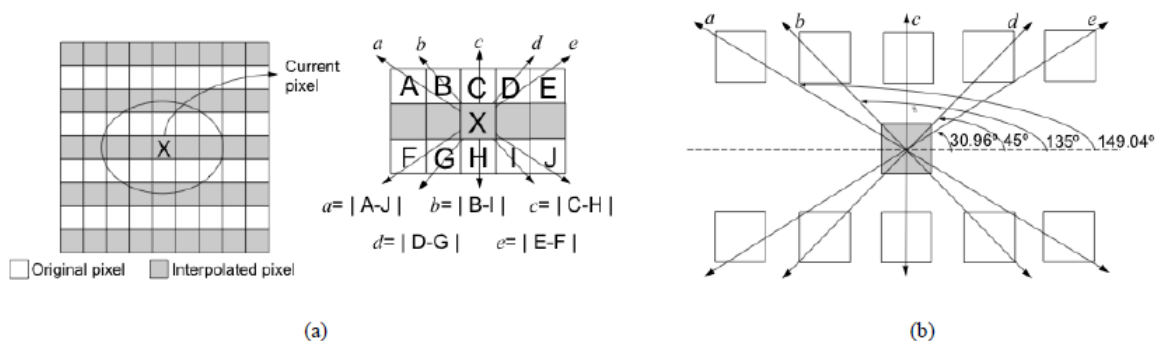


Fig. 2. ELA 5+5 algorithm with 5+5 pixels from the upper and lower lines: (a) ELA 5+5 scheme, (b) 5+5 Aperture ELA.

II. FUZZY-ELA 5+5 ALGORITHMS

This section describes two algorithms for interpolating the new pixel value, $I_p(x,y,t)$, which for the sake of simplicity will be annotated here as X .

A. Fuzzy-ELA 5+5 H1 algorithm

The ELA 5+5 algorithm interpolates X by analyzing the luminance differences in the 5+5 pixels in the upper and lower lines (see Fig. 2). ELA 5+5 looks for the most reliable edge direction (five directions are analyzed: 30.96° , 45° , 90° , 135° , 149.04°) and then applies line average along the selected direction. The pseudo-code of the ELA 5+5 algorithm is:

$$\begin{aligned}
 & \text{if } \min(a,b,c,d,e)=a \rightarrow X=(A+J)/2 \\
 & \text{elseif } \min(a,b,c,d,e)=b \rightarrow X=(B+I)/2 \\
 & \text{elseif } \min(a,b,c,d,e)=d \rightarrow X=(D+G)/2 \\
 & \text{elseif } \min(a,b,c,d,e)=e \rightarrow X=(E+F)/2 \\
 & \text{else } \rightarrow X=(C+H)/2
 \end{aligned} \quad (1)$$

ELA 5+5 produces good results if the minimum difference in luminance pixel values corresponds with the presence of an edge, which usually happens with clear edges.

However, problems can arise because minimum differences do not always indicate the direction of an edge. In Fig. 3, for example, ELA 5+5 introduces a clear error. Fig. 3(b), showing details of the progressive image in Fig. 3(a), has different examples of clear and unclear edges. ELA 5+5 was applied to interpolate the central pixel in the middle row of the examples shown in Fig. 3(c), and the results obtained can be seen in Fig. 3(d). As corroborated by this image, ELA 5+5 fails in the interpolation of the unclear edge.

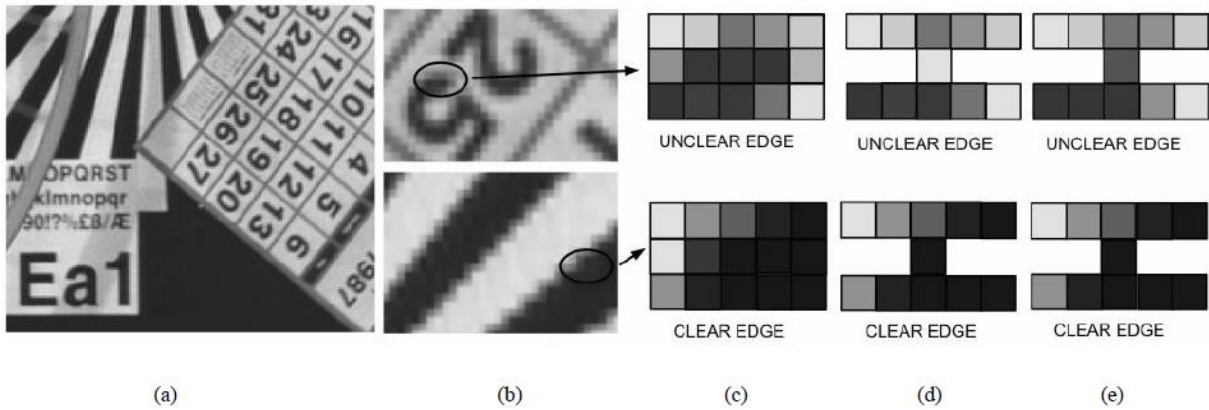


Fig. 3. (a) An interesting image for analysis, since it contains numerous areas with a high level of detail. (b) Areas with clear and unclear edges. (c) Close-ups of the corresponding edges. (d) ELA 5+5 results for clear and unclear edges. (e) Fuzzy-ELA 5+5 H1 results for clear and unclear edges.

TABLE I
RULE BASE OF THE FUZZY-ELA 5+5 H1 ALGORITHM

Rule	Antecedents	Consequent
1.	a is <i>SMALL</i> and b is <i>LARGE</i> and c is <i>LARGE</i> and d is <i>LARGE</i> and e is <i>LARGE</i>	$X=(A+J)/2$
2.	a is <i>LARGE</i> and b is <i>SMALL</i> and c is <i>LARGE</i> and d is <i>LARGE</i> and e is <i>LARGE</i>	$X=(B+I)/2$
3.	a is <i>LARGE</i> and b is <i>LARGE</i> and c is <i>LARGE</i> and d is <i>SMALL</i> and e is <i>LARGE</i>	$X=(D+G)/2$
4.	a is <i>SMALL</i> and b is <i>LARGE</i> and c is <i>LARGE</i> and d is <i>LARGE</i> and e is <i>LARGE</i>	$X=(E+F)/2$
5.	b is <i>VERY_LARGE</i> and c is <i>LARGE</i> and d is <i>VERY_LARGE</i>	$X=(B+D+G+I)/4$
6.	<i>Otherwise</i>	$X=(C+H)/2$

The proposed fuzzy approach applies heuristic knowledge to overcome the limitations of unclear edges. The following heuristic rules are employed to obtain a correct estimation of the edge direction:

- An edge is clear in one direction if the luminance difference in that direction is small and the differences in the other directions are large.
- If luminance differences b and d are very small and difference c is large, there is no edge and the vertical linear interpolation performance is poor; the best option is a linear interpolation between the neighbors with small differences: B , D , G and I .
- Otherwise, a vertical linear interpolation would be the most appropriate solution.

These rules are fuzzy because the concepts *SMALL*, *LARGE* and *VERY_SMALL* are understood as fuzzy, rather than crisp, values. This proposal translates the heuristic knowledge into fuzzy reasoning. *SMALL*, *LARGE* and *VERY_SMALL* are represented by fuzzy sets the membership functions of which change continuously, instead of abruptly, between 0 and 1

membership values (μ), as shown in Fig. 4. The fuzzy reasoning rule base is described in Table I.

The product operator is used as the “and” connective for antecedents to calculate the activation degree of rules 1 to 5, (β_i , $i=1, \dots, 5$). The activation degree of the sixth rule is defined as follows:

$$\beta_6 = 1 - (\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5) \quad (2)$$

With the product operator as the connective, β_6 never takes negative values. Since the rule consequents (c_i) are not fuzzy, the global conclusion provided by the rule base is calculated by applying the Fuzzy Mean defuzzification method as follows:

$$X = \frac{\sum_{i=1}^6 \beta_i \cdot c_i}{\sum_{i=1}^6 \beta_i} \quad (3)$$

Since the sum of the rule activation degrees is equal to one, no division is required:

$$X = \beta_1 \cdot \left(\frac{A+J}{2} \right) + \beta_2 \cdot \left(\frac{B+I}{2} \right) + \beta_3 \cdot \left(\frac{D+G}{2} \right) + \beta_4 \cdot \left(\frac{E+F}{2} \right) + \beta_5 \cdot \left(\frac{B+D+G+I}{4} \right) + \beta_6 \cdot \left(\frac{C+H}{2} \right) \quad (4)$$

The Fuzzy-ELA 5+5 H1 algorithm applies Line Average in the directions a , b , d or e if there is a clear edge (β_1 , β_2 , β_3 , or β_4 takes the value 1 and the others are 0). Otherwise, several rules are active and the new pixel is interpolated using all or several of the values of the 5+5 pixels (A, B, C, D, E, F, G, H, I, J). Consequently, with clear edges the performance of the proposed method should be as good as the results obtained by the ELA 5+5 algorithm, whereas with unclear edges it should surpass the ELA 5+5 performance. This is illustrated in Fig. 3(e), where the examples in Fig. 3(c) have been processed with the Fuzzy-ELA 5+5 H1 algorithm.

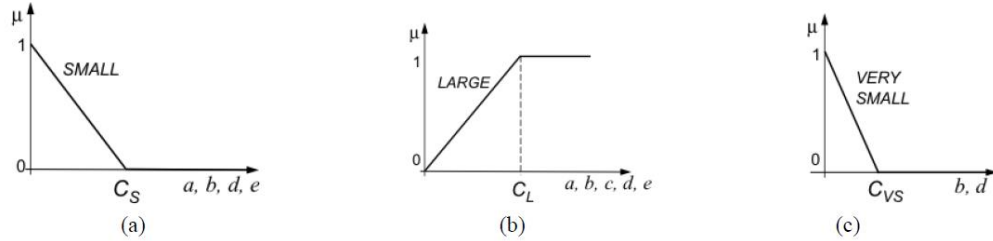


Fig. 4. Piecewise linear membership functions that are employed: (a) SMALL, (b) LARGE, (c) VERY_SMALL.

Algorithm performance can be improved by tuning parameters. Shapes of piecewise linear functions in Fig. 4 were chosen heuristically, but there are many possibilities for selecting parameters that functionally describe the concepts *SMALL*, *LARGE*, and *VERY_SMALL* (C_S , C_L , and C_{VS}). In fact, only one condition needs to be fulfilled to make the concepts *SMALL* and *VERY_SMALL* meaningful: C_{VS} should be smaller than C_S . Of the many values that meet this requirement, some will unquestionably provide better results than others. If a set of data can be obtained from progressive images, an error function can be used to evaluate the difference between using one parameter or another, and supervised learning algorithms can be applied to minimize error. This is the approach that was followed to select good values for the membership function parameters.

To facilitate the tuning of the parameters, the *Xfuzzy_3* environment was employed. *Xfuzzy_3* is a fuzzy system development environment with a set of tools that cover all the stages involved in designing fuzzy logic-based systems [24]. The fuzzy system used in this algorithm was described in *Xfuzzy_3* as a first-order Takagi-Sugeno system, since its rule consequents can be considered as linear functions of the inputs. This fuzzy system was tuned by supervised learning using the *Xfuzzy_3* CAD tool called *xfsl*. From the wide set of learning algorithms available in this tool, the Marquardt-Levenberg method was chosen. The *xfsl* tool allows the user to configure the tuning process, and therefore only those parameters involved in the description of the membership functions were enabled to participate in this process. A set of input/output training patterns was generated using 2 or 3 frames from several video test sequences presented in such a manner as to consider the highest number of different situations (from *VERY_SMALL* to *LARGE* luminance differences). The result of the learning process was that parameter C_{VS} of the *VERY_SMALL* membership function took a value close to 8; parameter C_L of *LARGE* was approximately 64 for input variable b , and parameters C_S and C_L for the *SMALL* and *LARGE* functions respectively were close to 16 for the rest of the input variables.

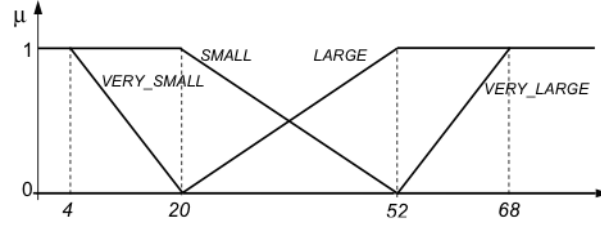


Fig. 5. Membership functions used by the Fuzzy-ELA 5+5 H2 algorithm.

B. Fuzzy-ELA 5+5 H2 algorithm

This second approach also applies heuristic knowledge, but this time in a more structured fashion because the five possible directions are treated separately and more fuzzy concepts are considered. The heuristic rules of the previous algorithm are applied for the three most relevant directions (45° , 90° , and 135°), as follows:

- An edge is clear in one of the three directions if the luminance difference (b , c , or d) in that direction is small and the differences in the other two directions are large.
- If differences b and d are very small and difference c is large, there is no edge and the vertical linear interpolation performance is poor; the best option is linear interpolation between neighbors with small differences: B, D, G, I.

This heuristics is considered by three of the fuzzy rules described in Table II (rules 1 to 3). The most extreme directions (30.96° and 147.04°) are evaluated as follows:

- There is an edge in one of the extreme directions (a , e) when the difference in that direction is small, the differences in the two closest directions are large, and the differences in the other two directions are very large.

This heuristics is considered by rules 4 and 5 in Table II. Otherwise, a vertical linear interpolation would be the most adequate solution, in accordance with rule 6.

TABLE II
RULE BASE OF THE FUZZY-ELA 5+5 H2 ALGORITHM

Rule	Antecedents	Consequent
1.	b is <i>SMALL</i> and c is <i>LARGE</i> and d is <i>LARGE</i>	$X=(B+I)/2$
2.	b is <i>LARGE</i> and c is <i>LARGE</i> and d is <i>SMALL</i>	$X=(D+G)/2$
3.	b is <i>VERY_SMALL</i> and c is <i>LARGE</i> and d is <i>VERY_SMALL</i>	$X=(B+D+G+I)/4$
4.	a is <i>SMALL</i> and b is <i>LARGE</i> and c is <i>LARGE</i> and d is <i>VERY_LARGE</i> and e is <i>VERY_LARGE</i>	$X=(A+J)/2$
5.	a is <i>VERY_LARGE</i> and b is <i>VERY_LARGE</i> and c is <i>LARGE</i> and d is <i>LARGE</i> and e is <i>SMALL</i>	$X=(E+F)/2$
6.	Otherwise	$X=(C+H)/2$

To ensure that rules 4 and 5 are activated whenever an edge really appears in directions a and e , it is considered that edges are not an isolated feature of the image: that is to say, they are not considered as belonging to one specific pixel in the image. Since directions b and d are close to a and e , it seems logical that the rules to interpolate along these edge directions (the first and fourth rules for b and a , and the second and fifth rules for d and e) should frequently be activated simultaneously.

To implement the above considerations, the following condition was imposed on the fourth rule:

$$\begin{aligned}
 & \text{if } \left(\begin{aligned} & (\max(\beta_4|_c, \beta_1|_c) > 0) \text{ and } (\max(\beta_4|_p, \beta_1|_p) > 0) \\ & \text{and } (\max(\beta_4|_c, \beta_1|_c) > \max(\beta_5|_c, \beta_2|_c)) \end{aligned} \right) \\
 & \text{then } (\beta_4|_c = \beta_4|_c) \\
 & \text{else } (\beta_4|_c = 0)
 \end{aligned} \tag{5}$$

where ' c ' signifies the spatio-temporal coordinates of the current pixel (x,y,t) and ' p ' signifies the spatio-temporal coordinates of the previous pixel $(x-1,y,t)$.

The corresponding condition for the fifth rule is as follows:

$$\begin{aligned}
& \text{if } \left(\begin{aligned} & (\max(\beta_5|_c, \beta_2|_c) > 0) \text{ and } (\max(\beta_5|_p, \beta_2|_p) > 0) \\ & \text{and } (\max(\beta_5|_c, \beta_2|_c) > \max(\beta_4|_c, \beta_1|_c)) \end{aligned} \right) \\
& \text{then } (\beta_5|_c = \beta_5|_c) \\
& \text{else } (\beta_5|_c = 0)
\end{aligned} \quad (6)$$

The fuzzy concepts employed in this algorithm are modeled by the membership functions shown in Fig. 5. The overlapping degree of membership functions was selected to ensure that no more than two rules were activated simultaneously. Since a positive value for the activation degree of the sixth rule is always obtained, this strategy allows the minimum operator to be used as the ‘and’ connective. The particular breakpoints of the membership functions (4, 20, 52, and 68 in Fig. 5) were obtained after applying a learning stage, as with the Fuzzy-ELA 5+5 H1 algorithm. The output provided by the algorithm is again obtained using the Fuzzy Mean defuzzification method:

$$\begin{aligned}
X = & \beta_1 \cdot \left(\frac{B+I}{2} \right) + \beta_2 \cdot \left(\frac{D+G}{2} \right) + \beta_3 \cdot \left(\frac{B+D+G+I}{4} \right) + \\
& \beta_4|_c \cdot \left(\frac{A+J}{2} \right) + \beta_5|_c \cdot \left(\frac{E+F}{4} \right) + \beta_6 \cdot \left(\frac{C+H}{2} \right)
\end{aligned} \quad (7)$$

Fig. 6 qualitatively compares the performance of the fuzzy approaches with Line Average and ELA 5+5. As can be seen, edge-adaptive algorithms considerably reduce the staircase effect introduced by Line Average, and Fuzzy-ELA 5+5 H2 slightly improves the Fuzzy-ELA 5+5 H1 performance in this picture.

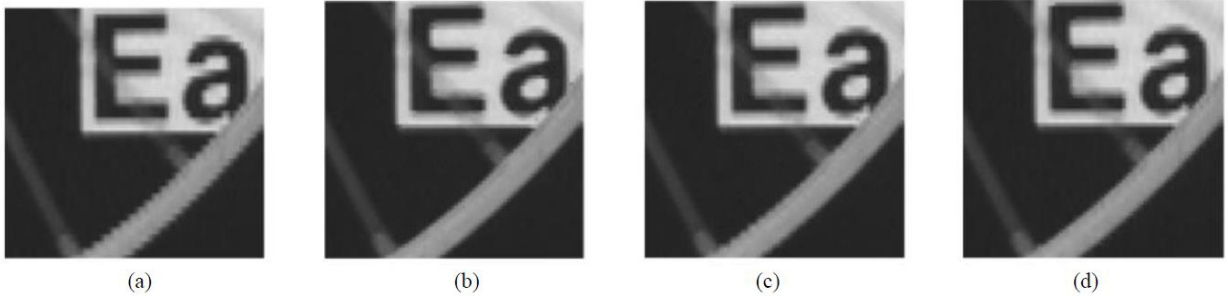


Fig. 6. Examples of clear edges after de-interlacing with: (a) Line Average, (b) ELA 5+5 [2], (c) Fuzzy-ELA 5+5 H1, (d) Fuzzy-ELA 5+5 H2.

III. SIMULATION RESULTS

The performance of the algorithms described above was analyzed by de-interlacing several sequences that have been widely used as benchmarks in video processing applications. The test material was a set of progressive sequences that had been interlaced in order to measure the difference between the obtained interpolated frames and the original frames. Numerous error measures have been proposed as figures of merit for evaluating digital video quality. One very popular quality criterion is the MSE (Mean Squared Error):

$$MSE(t) = \frac{1}{M \cdot N} \sum_{x,y} (I_p(x, y, t) - I_{original}(x, y, t))^2 \quad (8)$$

where I_p is the interpolated pixel and $I_{original}$ is the pixel value in the original progressive image, with a resolution of $M \times N$ pixels. Closely related to the MSE is the PSNR (Peak Signal to Noise Ratio):

$$PSNR(t) = 20 \log \frac{255}{\sqrt{MSE(t)}} \quad (9)$$

An improvement of 0.4 dBs in PSNR is quite perceptible by the human visual system in de-interlacing pictures.

A. Comparison between fuzzy proposals

Table III shows the average PSNR values obtained when de-interlacing 50 frames from a battery of progressive standard

sequences of different formats and origin. Video material came from video camera recordings, whereas the film material was provided by movie camera.

TABLE III
PSNR RESULTS (IN DBS) FOR FUZZY PROPOSALS

	News	Mother	Carphone	Missa	Paris	Trevor	PMenace	FireRose	Fargo1	Fargo2	Tokyo
Material	Video	Video	Video	Video	Video	Video	Video	Film	Film	Film	Film
Format	QCIF	QCIF	QCIF	CIF	CIF	CIF	DVD	PAL	PAL	PAL	PAL
FELA 5+5 H1	28.17	35.92	32.66	40.45	26.54	33.07	34.96	35.01	35.76	34.08	31.81
FELA 5+5 H2	29.21	36.21	32.92	40.71	26.66	33.52	35.24	35.12	35.89	34.34	32.19

When the two proposals are compared, the highest PSNR values are obtained by the Fuzzy-ELA 5+5 H2 algorithm. Improvements range from 0.06 dBs in the Phantom Menace sequence to 1.04 dBs in the News sequence, with an average improvement of 0.3 dBs.

Fig. 7 qualitatively compares the two proposals and ELA 5+5 in one illustrative image. Both fuzzy approaches clearly improve unclear edges and detailed areas of the picture (for example, see the stars in the flag) where ELA fails. The Fuzzy-ELA 5+5 H2 algorithm offers the best performance in the reconstruction of clear edges (in striped lines) and unclear edges. After examining the quantitative measurements (PSNR values) and qualitative figures of merit (visual inspection of de-interlaced pictures), it is concluded that the Fuzzy-ELA 5+5 H2 algorithm is a better de-interlacer than the Fuzzy-ELA 5+5 H1 algorithm.

Both algorithms were designed taking into account their hardware implementation. Hence the piecewise linear shape of their membership functions and the rounding of their parameters to avoid complex divisions. Also, no division was required to implement the defuzzification method (see equations (4) and (7)). A hardware demonstrator was developed on a medium-size FPGA to validate real-time implementation with HDTV resolution (30 frames/sec). This work was recently published by Brox *et al* [25].

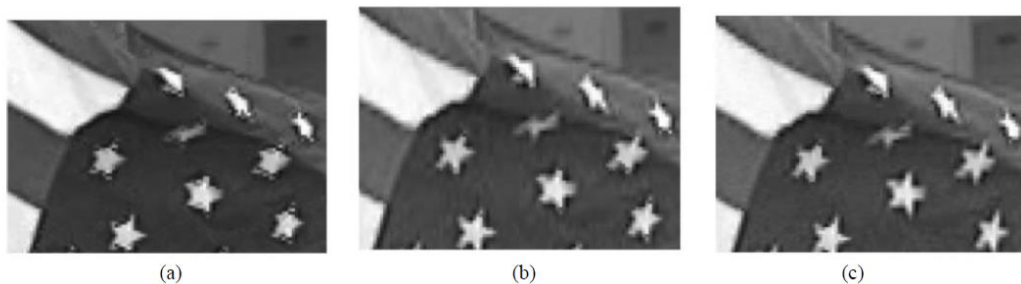


Fig. 7. Qualitative comparison between the two proposals in the Flag sequence: (a) ELA 5+5 [2], (b) Fuzzy-ELA 5+5 H1, (c) Fuzzy-ELA 5+5 H2.

B. Comparison with state-of-the-art edge-adaptive de-interlacers

A quantitative comparison was carried out by de-interlacing five well-known progressive test sequences in CIF format (all the sequences had a total number of 1240 frames). The set of methods used for this comparison included several recently reported edge-adaptive de-interlacers and the two ELA algorithms (ELA with 3+3 and 5+5 taps).

From the results shown in Table IV, it is concluded that the Fuzzy-ELA 5+5 H2 algorithm clearly surpasses not only the majority of the recently reported algorithms (ELA 3+3 [2], ELA 5+5 [2], NEDI [14], EELA [3], NEDD [12], FDED [17], but also the algorithm reported by de Haan [26]), although, in terms of average value, it offers no significant improvements over three of them: MELA [13], FWW [20], and IFRA [21].

Relevant improvements are produced in the Foreman sequence (0.57 dBs over MELA [13], 0.62 dBs over FWW [20], and 0.64 dBs over IFRA [21]) where the proposed method achieves the highest PSNR result (see the first row in Table IV). The analysis of this sequence is particularly interesting since it contains numerous edges and other spatial details. One example of how the proposed approach offers better image quality than IFRA is shown in Fig. 8.

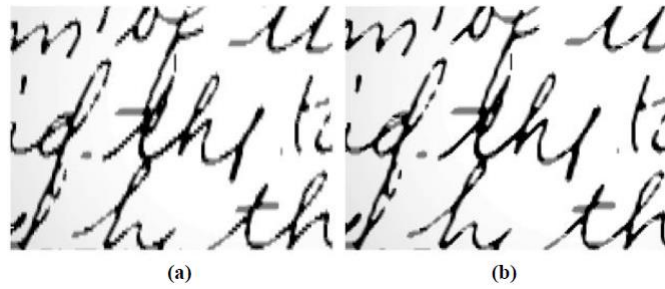


Fig. 8. Qualitative comparison between (a) IFRA [21] and (b) the proposed method (Fuzzy-ELA 5+5 H2).

TABLE IV
PSNR (IN DBS) COMPARISON FOR CIF SEQUENCES

Sequence	ELA 3+3[2]	ELA 5+5[2]	NEDI [14]	MELA [13]	EELA [3]	NEDD [12]	FDED [17]	FWW [20]	IFRA [21]	Alg. [26]	FELA 5+5 H1	FELA 5+5 H2
Foreman	31.39	30.51	30.94	31.28	30.40	30.64	30.78	31.23	31.21	31.61	31.58	31.85
Stefan	25.97	24.45	27.15	27.21	26.36	27.00	27.12	27.57	27.03	27.03	27.01	27.28
Flower	21.93	21.35	22.25	22.46	21.96	22.34	22.44	23.18	22.45	22.14	22.43	22.65
News	31.53	28.65	32.96	33.73	31.92	32.69	32.84	33.10	33.32	32.85	33.02	33.21
Mobile	23.34	21.98	24.73	25.30	23.27	24.76	24.87	25.41	25.34	24.95	24.89	25.15
Average	26.83	25.38	27.60	27.99	26.78	27.48	27.61	28.09	27.87	27.71	27.78	28.02

The MELA and FWW methods, which produce good PSNR results, have a high computational cost. Table V shows the CPU time required for de-interlacing one CIF frame on a conventional processor. MELA is 2.9 times slower than the Fuzzy-ELA 5+5 H2 algorithm, whereas FWW is 3.3 times slower than the proposed method. Since spatial de-interlacers are often part of a more complex de-interlacing technique, the selection of a simple technique is advisable for successful real-time implementations.

This comparison of state-of-the-art algorithms proves that the two proposals offer the best tradeoff between complexity and performance.

TABLE V
COMPARISON IN TERMS OF COMPUTATIONAL COST

Algorithm	MELA [13]	FWW [20]	FELA 5 5+5 H1	FELA 5+5 H2
CPU Time (s)	1.125	1.282	0.555	0.3892

C. Robustness of the fuzzy proposal against noise

The robustness of the Fuzzy-ELA 5+5 H2 algorithm against noise was explored by adding noise to the original image sequences. Two kinds of noise were added: Gaussian and impulse.

The results show that conventional ELA is able to avoid impulse noise and performs better than the fuzzy proposal for low values of noise density (1% and 4%). However, the fuzzy approach gradually reduces these differences as noise density increases. Line Average obtains poor results, losing more than 1 dB in PSNR values. After analyzing the results obtained from images corrupted with Gaussian white noise, it is concluded that the fuzzy approach achieves the best performance. The improvement is more evident as the noise level increases (see Table VI).

TABLE VI
PSNR (IN DBS) FOR NOISE SIMULATIONS

Kind of noise	Impulse D=1%	Impulse D=4%	Impulse D=10%	Impulse D=16%	Gaussian V=0.002	Gaussian V=0.004	Gaussian V=0.006	Gaussian V=0.008
Line Average	25.46	20.06	16.23	14.22	26.87	24.62	23.06	21.99
ELA 3+3	26.82	21.46	17.32	15.01	26.78	24.44	22.93	21.86
FELA 5+5 H2	26.61	21.21	17.37	15.19	27.04	24.74	23.29	22.22

D. Comparison with professional edge-adaptive de-interlacers

Robust edge-adaptive de-interlacing algorithms implemented on consumer electronics devices incorporate post-filtering processing to preserve favorable results on clearly defined edges and improve interpolation in areas where conventional edge-adaptive algorithms fail [27]. One of the most widely used post-filtering operators is the median operator, which easily removes

artifacts introduced by conventional edge-adaptive algorithms. This technique contributes to the improved reconstruction of unclear edges by ELA algorithms (see Fig. 9(a)).

Strictly speaking, fuzzy approaches do not require median filtering since they naturally eliminate annoying errors in ELA algorithms, and therefore there is no need to include a post-filtering stage with this purpose. However, the edge-adaptive interpolation implemented by the fuzzy algorithms may bring about a loss of image contrast in areas with edges, and the inclusion of contrast improvement post-filtering improves overall picture quality (Fig. 9(b)).

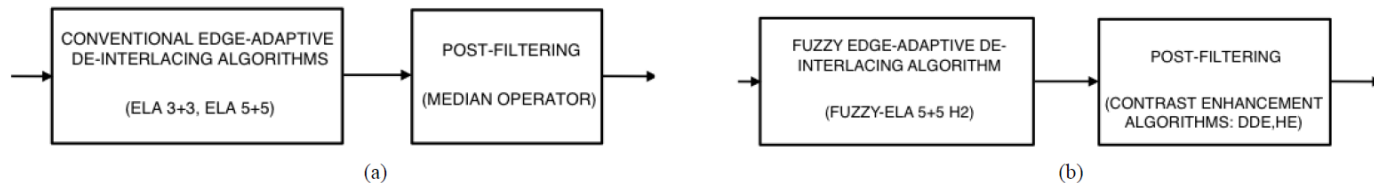


Fig. 9. Post-filtering to increase the overall performance of edge-adaptive de-interlacing algorithms: (a) a post-filtering based on the median operator, (b) a post-filtering based on a contrast algorithm.

Fig. 10 shows de-interlaced images obtained after applying two contrast improvement algorithms: Dynamic Detail Enhancement (DDE) and Histogram Enhancement (HE). DDE improves contrast in areas with details. HE re-calculates the luminance values of the pixels in the picture by means of histogram analysis [27]. Reconstruction of clear edges is improved by including contrast enhancement algorithms (see Fig. 10). Fig. 11 shows the superior performance of Fuzzy-ELA 5+5 H2 following the de-interlacing of a picture with numerous fine details. The ELA 5+5 algorithm cannot reconstruct the details in the picture even if the post-filtering stage is included. Fuzzy-ELA 5+5 H2, however, is competitive even against a more complex algorithm, called EDDI, which de-interlaces by combining motion vectors and information on picture edges [10]. The calculation of motion vectors considerably increases the complexity of EDDI in comparison with the proposed algorithm. The PSNR results obtained after de-interlacing and post-filtering are shown in Table VII. From these results, it can be concluded that the inclusion of a contrast algorithm does not significantly improve the resulting PSNR.

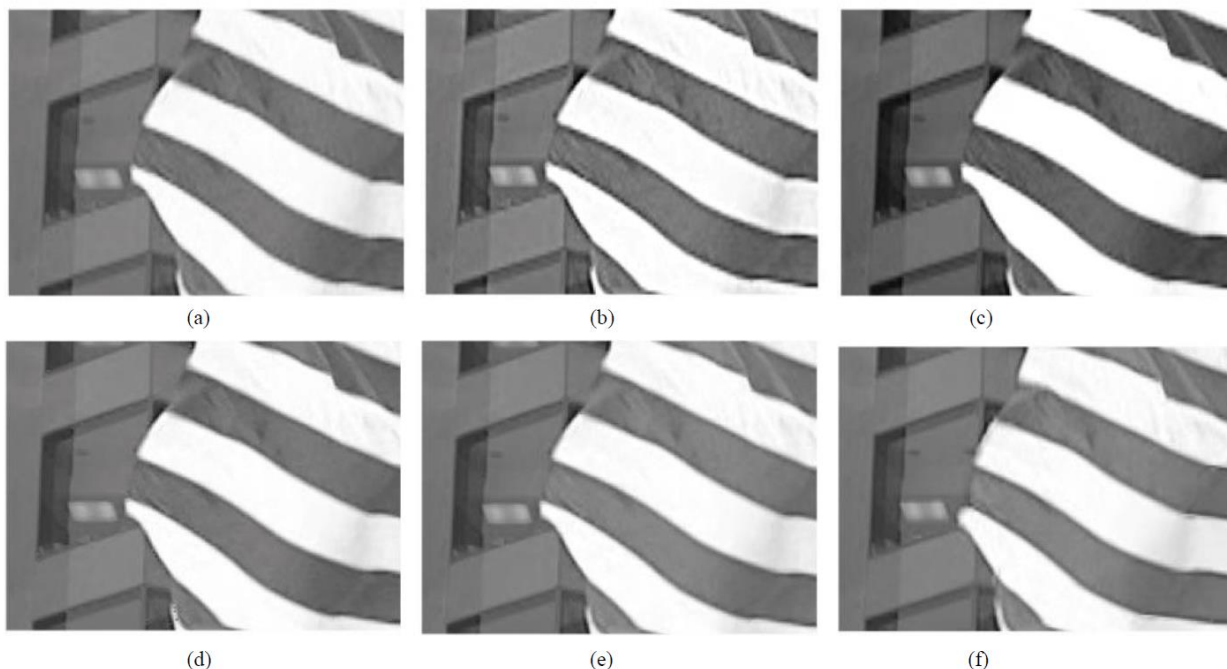


Fig. 10. Reconstruction of clearly defined edges in the Flag sequence after the inclusion of post-filtering techniques: (a) Fuzzy-ELA 5+5 H2, (b) Fuzzy-ELA 5+5 H2 + DDE, (c) Fuzzy-ELA 5+5 H2 + HE, (d) ELA 3+3 + MEDIAN, (e) ELA 5+5 + MEDIAN, (f) EDDI.

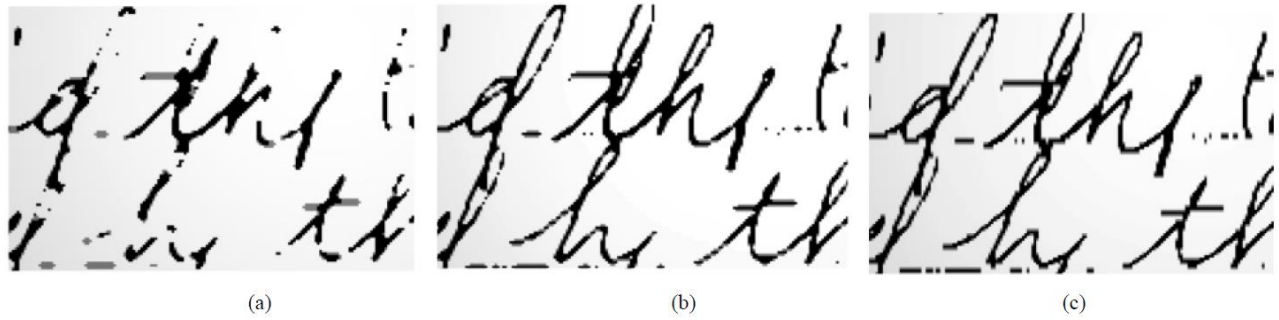


Fig. 11. The inclusion of the median operator cannot properly reconstruct the letters in the image if ELA 5+5 is used as the de-interlacer: (a) ELA 5+5 + MEDIAN, (b) Fuzzy-ELA 5+5 H2 + HE, (c) EDDI.

IV. CONCLUSION

This paper presents two edge-adaptive de-interlacing algorithms that use fuzzy logic to detect edge directions. After extensive analysis, the conclusion is that the Fuzzy-ELA 5+5 H2 algorithm is superior to the other proposed algorithm (in terms of quantitative PSNR values and qualitative visual sensation) in both clear and unclear edges. Compared to other state-of-the-art edge-adaptive algorithms, the proposed algorithm offers a good trade-off between performance and computational cost, and there exists a strong case for its inclusion in low-cost hardware solutions.

TABLE VII
PSNR (IN DBS) COMPARISON FOR CIF SEQUENCES

Sequence	ELA 5+5 +Median	EDDI	FELA 5+5 H2+HE	FELA 5+5 H2+DDE
Foreman	31.06	31.74	31.08	31.79
Stefan	25.41	26.84	27.12	27.31
Flower	22.06	22.45	22.37	22.46
News	30.39	32.83	33.17	32.97
Mobile	22.93	24.36	24.92	25.21

REFERENCES

- [1] G. de Haan and E. B. Bellers, "De-interlacing - an overview," *Proceedings of the IEEE*, vol. 86, no. 9, pp. 1839-1857, Sep. 1998.
- [2] T. Doyle and M. Looymans, "Progressive scan conversion using edge information," in *Signal Processing of HDTV*, II, Elsevier, pp. 711-721, 1990.
- [3] T. Chen, H. R. Wu, and Z. H. Yu, "Efficient deinterlacing algorithm using edge-based line average interpolation," *Optical Engineering*, vol. 39, no. 8, pp. 2101-2105, Aug. 2000.
- [4] P.-Y. Chen and Y.-H. Lai, "A low-complexity interpolation method for deinterlacing," *IEICE Trans. on Information & Systems*, vol. E90-D, no. 2, Feb. 2007.
- [5] M. H. Lee, J. H. Kim, J. S. Lee, K. K. Ryu, and D. Song, "A new algorithm for interlaced to progressive scan conversion based on directional correlations and its IC design," *IEEE Trans. on Consumer Electronics*, vol. 40, no. 2, pp. 119-129, May 1994.
- [6] C. J. Kuo, C. Liao, and C. C. Lin, "Adaptive interpolation technique for scanning rate conversion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 317-321, Jun. 1996.
- [7] 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 Trans. on Consumer Electronics*, vol. 46, no. 4, pp. 999-1006, Nov. 2000.
- [8] J. Salonen and S. Kalli, "Edge adaptive interpolation for scanning rate conversion," in *Signal Processing of HDTV*, IV, Elsevier, pp. 757-764, 1993.
- [9] R. Simonetti, A. P. Filisan, S. Carrato, G. Ramponi, and G. Sicuranza, "A deinterlacer for IQTV receivers and multimedia applications," *IEEE Trans. on Consumer Electronics*, vol. 39, no. 3, pp. 234-240, Aug. 1993.
- [10] G. de Haan and R. Lodder, "De-interlacing of video data using motion vector and edge information," in *Proc. IEEE Int. Conf. on Consumer Electronics*, Los Angeles, USA, Jun. 2002.
- [11] Y. L. Chang, S. F. Lin, and L. G. Chen, "Extended intelligent edge-based line average with its implementation and test method," in *Proc. IEEE Int. Symposium on Circuits and Systems*, vol. 2, pp. 341-344, Vancouver, Canada, May 2004.
- [12] H. Yoo and J. Jeong, "Direction-oriented interpolation and its application to de-interlacing," *IEEE Trans. on Consumer Electronics*, vol. 48, no. 4, pp. 954-932, Nov. 2002.
- [13] W. Kim, S. Jin, and J. Jeong, "Novel intra deinterlacing algorithm using content adaptive interpolation," *IEEE Trans. on Consumer Electronics*, vol. 53, no. 3, pp. 1036-1043, Aug. 2007.
- [14] M. K. Park, M. G. Kang, K. Nam, and S. G. Oh, "New edge dependent deinterlacing algorithm based on horizontal edge pattern," *IEEE Trans. on Consumer Electronics*, vol. 49, no. 4, pp. 1508-1512, Nov. 2003.
- [15] D.-H. Lee, "A new edge-based intra-field interpolation method for deinterlacing using locally adaptive-thresholded binary image," *IEEE Trans. on Consumer Electronics*, vol. 54, no. 1, pp. 110-115, Feb. 2008.
- [16] M. Mancuso, V. D'Alto, and R. Poluzzi, "Fuzzy edge-oriented motion adaptive noise reduction and scanning rate conversion," in *Proc. IEEE Asia-Pacific Conference on Circuits and Systems*, pp. 652-656, Taipei, Taiwan, Dec. 1994.

- [17] F. Michaud, C. T. Le Dinh, and G. Lachiver, "Fuzzy detection of edge direction for video line doubling," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 7, no. 3, pp. 539-542, Jun. 1997.
- [18] D. Van De Ville, W. Philips, and I. Lemahieu, "Fuzzy-based motion detection and its application to de-interlacing," in *Fuzzy techniques in image processing*. Studies in Fuzziness and Soft Computing, Physica-Verlag Editorial, Heidelberg, Germany, 2000.
- [19] P. Brox, I. Baturone, S. Sánchez-Solano, J. Gutiérrez-Ríos, and F. Fernández-Hernández, "A fuzzy edge-dependent motion adaptive algorithm for deinterlacing," *Fuzzy Sets and Systems*, vol. 158, no. 3, pp. 337-347, Feb. 2007.
- [20] G. Jeon, R. Lee, D. Kim, J. Lee, and J. Jeong, "Weighted fuzzy filter on interlaced-to-progressive conversion," in *Proc. IEEE Int. Conf. on Multimedia & Expo*, pp. 173-176, Hannover, Germany, Jun. 2008.
- [21] G. Jeon, S.-J. Park, J. Lee, R. Lee, S. Kim, and J. Jeong, "Enhancement of interlaced images by fuzzy reasoning approach," in *Proc. IEEE Int. Conf. on Image Processing*, pp. 3117-3120, Cairo, Egypt, Nov. 2009.
- [22] G. Jeon, M. Anisetti, D. Kim, V. Bellandi, E. Damiani, and J. Jeong, "Fuzzy rough sets hybrid scheme for motion and scene complexity adaptive deinterlacing," *Image and Vision Computing*, vol. 27, no. 4, pp. 425-436, Mar. 2009.
- [23] G. Jeon, J. You, and J. Jeong, "Weighted fuzzy reasoning scheme for interlaced to progressive conversion," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 842-855, Jun. 2009.
- [24] I. Baturone, F. J. Moreno-Velo, S. Sánchez-Solano, A. Barriga, P. Brox, A. Gersnoviez, and M. Brox, "Using Xfuzzy environment for the whole design of fuzzy systems," in *Proc. IEEE Int. Conf. on Fuzzy Systems*, pp. 517-522, London, United Kingdom, Jul. 2007.
- [25] P. Brox, I. Baturone, S. Sánchez-Solano, "Fuzzy logic-based embedded system for video de-interlacing," *Applied Soft Computing*, vol. 14, Part C, pp. 338-346, Jan. 2014.
- [26] P. Brox, I. Baturone, and S. Sánchez-Solano, "Interlaced to progressive scan conversion using a fuzzy edge-based line average algorithm," in *Proc. IEEE Int. Workshop on Intelligent Signal Processing*, pp. 10-15, Faro, Portugal, Sep. 2005.
- [27] G. de Haan, "Image enhancement," in *Digital Video: Post Processing*, University Press Eindhoven, pp. 81-112, 2005.