INTERLACED TO PROGRESSIVE SCAN CONVERSION USING A FUZZY EDGE-BASED LINE AVERAGE ALGORITHM

P. Brox, I. Baturone, S. Sánchez-Solano

Instituto de Microelectrónica de Sevilla - Centro Nacional de Microelectrónica
Avda. Reina Mercedes s/n, (Edif. CICA),
E-41012, Sevilla, Spain.

Proc. IEEE International Symposium on Intelligent Signal Processing (WISP 2005), Faro, September 1-3, 2005.

Interlaced to Progressive Scan Conversion Using a Fuzzy Edge-based Line Average Algorithm

P. Brox, I. Baturone, S. Sánchez-Solano
Instituto de Microelectrónica de Sevilla. Centro Nacional de Microelectrónica
Avda. Reina Mercedes S/N, Edif. CICA, 41012 Seville, Spain

Abstract **– De-interlacing methods realize the interlaced to progressive conversion required in many applications. Among them, intra-field methods are widely used for their good trade off between performance and computational cost. In particular, the ELA algorithm is well-known for its advantages in reconstructing the edges of the images, although it degrades the image quality where the edges are not clear. The algorithm proposed in this paper uses a simple fuzzy system which models heuristic rules to improve the ELA rules. It can be implemented easily in software and hardware since the increase in computational cost is very low. Simulation results are included to illustrate the advantages of the proposed fuzzy ELA algorithm in de-interlacing non noisy and noisy images.**

Keywords **– De-interlacing, edge-dependent interpolation, fuzzy logic.**

I. Introduction

Interlaced scan is used in current NTSC, SECAM and PAL systems. However, high quality monitors, displays, projectors, and HDTV systems work with progressive scan formats. This is why several interlaced to progressive conversion (IPC) or de-interlacing algorithms have been proposed in the last few decades [1-8]. They can be classified into three categories: spatial (or intra-field) techniques, temporal (or inter-field) techniques, and hybrid methods. The first ones are the most advantageous in terms of hardware implementation since no memory is required to store previous fields. Among them, line doubling and line average methods have been widely used for its low computational cost. The drawback, however, lies in the blurring and step-edge effects they introduce. Several direction-dependent interpolation algorithms have been proposed to overcome such shortcomings. The simplest one, which is known as the edge-based line average (ELA) algorithm, uses the directional correlations between 3+3 pixels of the adjacent scan lines within the current field to interpolate the missing line linearly [1]. This algorithm works well when the edge directions are estimated correctly but, otherwise, it introduces errors and degrades the image quality. These problems appear when directional correlations are similar (in horizontal edges, for instance) and/or the image is corrupted with noise. To increase the robustness and reduce the sensitivity to noise, a three-point median filter that uses information from the previous field is proposed in [2], but this requires the use of an expensive field memory. To improve estimation of edge directions, the ELA algorithm is combined with the line doubling method in [3] and an adaptive ELA technique is proposed in [4], which are more costly computationally. Other authors resort to the use of a larger neighbourhood for the detection of edge directions (5+5 taps in [5], 7+7 taps in [6], up to 11+11 taps in [7], and 34+34 taps in [8]), with the resulting increase of hardware resources. Besides, the use of threshold values is proposed in [3] and [7] to ensure that estimated edges are dominant.

The algorithm proposed herein improves the ELA algorithm in estimating correctly the edge directions and processing noisy images. Besides, this is achieved with a low increase in computational cost. The idea is to use simple fuzzy rules to detect edge directions so that linear interpolation is applied between 3+3 taps of the current field when the edge is clear and non linear interpolation is carried out for non clear edges.

The paper is organized as follows. Section 2 describes the proposed fuzzy ELA algorithm. Several simulation and comparison results are shown in Section 3. Hardware implementation is presented in Section 4. Finally, some conclusions are given in Section 5.

II. Fuzzy Ela Algorithm

Fig. 1 shows the pixels used by the ELA algorithm to interpolate the pixel value X. The pseudo code of this algorithm is as follows:

$$a=|A-F| \quad b=|B-E| \quad c=|C-D| \tag{1}$$

if $\min\{a,b,c\}=a$, $X=(A+F)/2$
elseif $\min\{a,b,c\}=c$, $X=(C+D)/2$
else $X=(B+E)/2$

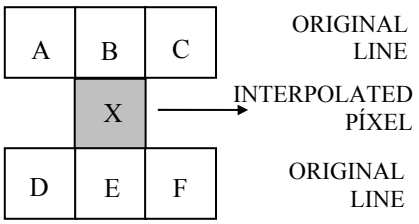

Fig. 1. 3x3 window for the ELA algorithm.

Selection of the edge direction is so crisp that errors appear easily when there is not an edge or it is not clear, as illustrated in Fig. 2. If we apply heuristics to solve these limitations, we will say that an edge is clear in direction *a* not only if *a* is small but also *b* and *c* are large, and something similar happens to an edge in direction *c*. If there is a strongly small correlation in directions *a* and *c*, and a large correlation in direction *b*, neither there is an edge nor vertical linear interpolation performs well, the best option is a linear interpolation between the neighbors with small correlations: A, B, C, F. In other cases, the best thing is to calculate a vertical linear interpolation. This heuristic knowledge is fuzzy since the concepts of small and large are not understood as threshold values but as fuzzy ones. Hence, our proposal is to model this knowledge by a fuzzy system.

The rule base of our fuzzy system is described in Table 1. Using fuzzy logic, the concepts of *SMALL* and *LARGE* are represented by membership functions that change continuously instead of abruptly between 0 and 1 membership values, as shown in Fig. 3a-b. The linguistic hedge '*strongly*' acting upon the concept of SMALL modifies its membership function as illustrated in Fig. 3c [9]. This fuzziness makes all the rules may be activated simultaneously, contrary to what happens in the ELA algorithm. Using the minimum operator to represent the connectives *and*, the activation degrees of the rules, α_i, are calculated as follows:

$$\begin{aligned}
\alpha_1 &= \min\{\mu_{SMALL\,a}(h), \mu_{LARGE\,b}(h), \mu_{LARGE\,c}(h)\} \\
\alpha_2 &= \min\{\mu_{LARGE\,a}(h), \mu_{LARGE\,b}(h), \mu_{SMALL\,c}(h)\} \\
\alpha_3 &= \min\{\mu_{strongly\ SMALL\,a}(h), \mu_{LARGE\,b}(h), \mu_{strongly\ SMALL\,c}(h)\} \\
\alpha_4 &= 1 - \alpha_1 - \alpha_2 - \alpha_3
\end{aligned} \tag{2}$$

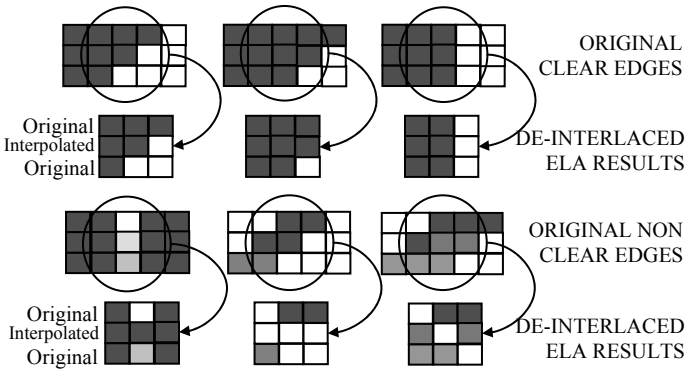

Fig. 2. Examples of edge detection by ELA algorithm.

TABLE 1

FUZZY RULE SET

if	*antecedents*	*then*	*consequent*
1)	*(a is SMALL) and (b is LARGE) and (c is LARGE)*		*(A+F)/2*
2)	*(a is LARGE) and (b is LARGE) and (c is SMALL)*		*(C+D)/2*
3)	*(a is strongly SMALL) and (b is LARGE) and (c is strongly SMALL)*		*(A+F+C+D)/4*
4)	*otherwise*		*(B+E)/2*

Since the consequents, c_i, of the rules are not fuzzy, the global conclusion provided by the system is calculated by applying the Fuzzy Mean defuzzification method, as follows:

$$X = \frac{\sum_{i=1}^{4} \alpha_i\, c_i}{\sum_{i=1}^{4} \alpha_i} = \frac{\alpha_1\, c_1 + \alpha_2\, c_2 + \alpha_3\, c_3 + \alpha_4\, c_4}{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4} \tag{3}$$

Substituting the consequents, c_i, by their values in Table 1, and applying that $\alpha_1+\alpha_2+\alpha_3$ is equal to 1, the above expression can be given as:

$$X = \alpha_1\left(\frac{A+F}{2}\right) + \alpha_2\left(\frac{C+D}{2}\right) + \alpha_3\left(\frac{A+F+C+D}{4}\right) + \alpha_4\left(\frac{B+E}{2}\right) \tag{4}$$

III. SIMULATIONS RESULTS

In order to obtain objective performance measurements of our algorithm compared to others, the even lines of original progressive images have been eliminated and calculated by applying different de-interlacing techniques. The line doubling, line average, ELA, the proposed algorithm with crisp instead of fuzzy descriptions of labels (crisp method), and our fuzzy ELA algorithms are compared in Table 2 for images taken from http://decsai.ugr.es. The MSE and PSNR values between the original progressive and de-interlaced images have been considered as figures of merit.

The results in Table 2 show how the ELA algorithm reduces

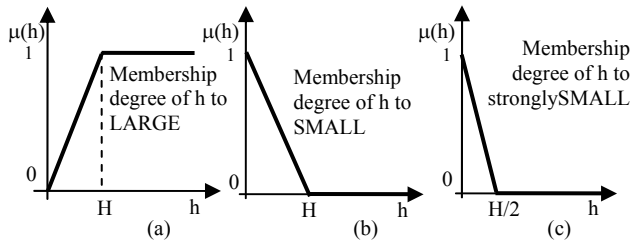

Fig. 3. Membership functions of the fuzzy sets used.

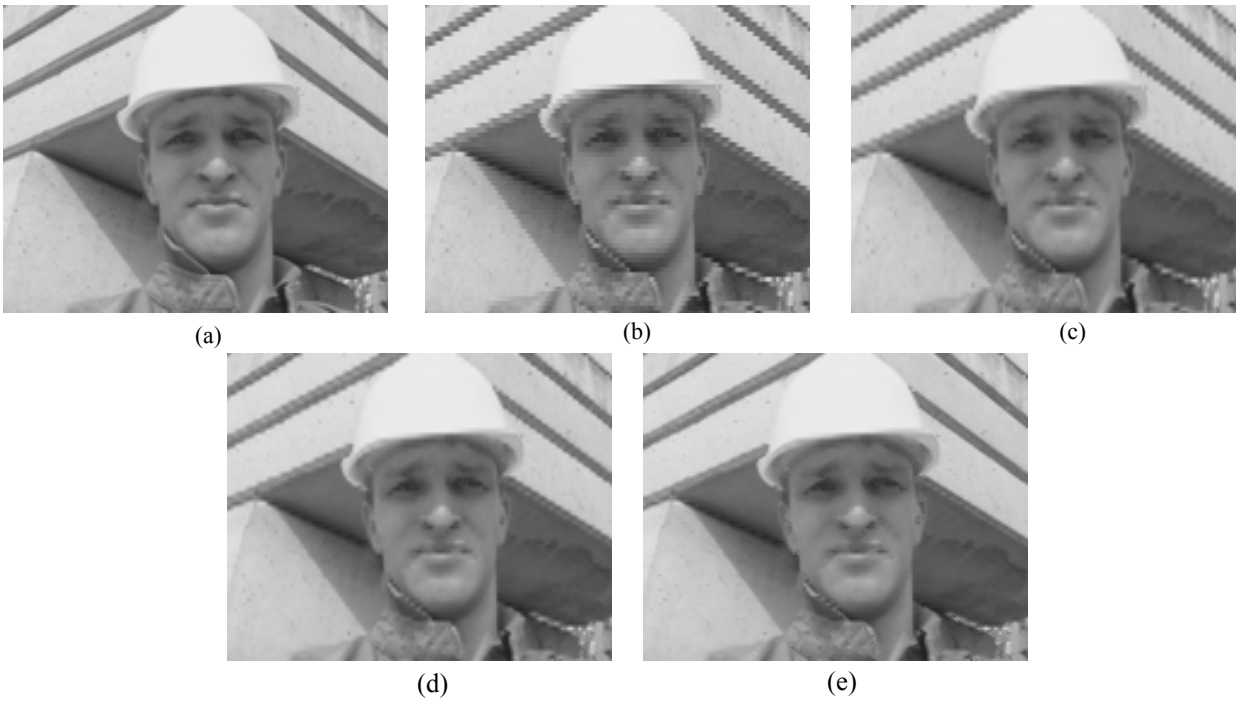

Fig. 4. (a) Original progressive image. De-interlaced images obtained by: (b) doubling line, (c) line average, (d) ELA and (e) Fuzzy ELA method.

the global PSNR of many images compared to the simple average method (although it improves the edge quality of the de-interlaced images). The proposed algorithm works as well as ELA regarding clear edges and better in the rest of the image. It can be seen also that the use of fuzzy labels allows better results. Qualitative performance and advantages of the proposed algorithm can be also seen in Fig. 4, which corresponds to Foreman image.

Different image sequences have been also analyzed. The results in Fig. 5 and 6 correspond to Susie sequence with an original resolution of 240x352 pixels. Again the proposed algorithm works considerably better than ELA. Although PSNR values of line average method are close to our results, our technique de-interlaces edges better than it, as shown in Fig. 5.

The robustness of the proposed algorithm against noise has been analyzed by adding impulse and Gaussian noise to original frames. Fig. 7 and 8 show the results corresponding to Susie sequence. Performance of the fuzzy ELA algorithm remains better than the other ones, especially for fields corrupted by Gaussian noise.

The threshold value H used in the descriptions of the fuzzy labels (Fig. 3) has been fixed heuristically to a value that has been the same in all the analysis. It has been proven that the results are not very sensitive to this value.

The proposed algorithm can be also extended to work with a larger neighbourhood of pixels but the improvements are not very significant.

TABLE 2

RESULTS APPLYING DIFFERENT DE-INTERLACING TECHNIQUES

De-interlacing algorithm	Figure of merit	Elaine 512x512 pixels	Butterfly 512x512 pixels	Bird 512x512 pixels	Einstein 256x256 pixels	Mrchest 256x256 pixels	Zelda 512x512 pixels	Missa 288x360 pixels	Foreman 144x176 pixels
Doubling	MSE PSNR (dBs)	58,22 30,48	95,08 28,35	47,98 31,32	55,22 30,71	21,48 34,81	14,06 36,65	15,31 36,28	106,75 27,85
Average	MSE PSNR (dBs)	36,74 32,48	44,47 31,65	21,58 34,79	25,76 34,02	3,78 42,35	4,27 41,82	5,82 40,48	38,55 32,27
ELA	MSE PSNR (dBs)	29,65 33,41	50,13 31,13	26,31 33,93	33,12 32,93	5,27 40,91	8,09 39,05	7,08 39,63	25,42 34,08
Crisp method	MSE PSNR (dBs)	35,65 32,61	44,47 31,65	21,63 34,78	25,77 34,02	3,78 42,35	4,42 41,68	5,88 40,48	23,81 34,37
Proposed method	MSE PSNR (dBs)	27,61 33,72	43,06 31,79	20,89 34,93	24,43 34,25	3,47 42,71	4,12 41,98	5,59 40,65	22,42 34,62

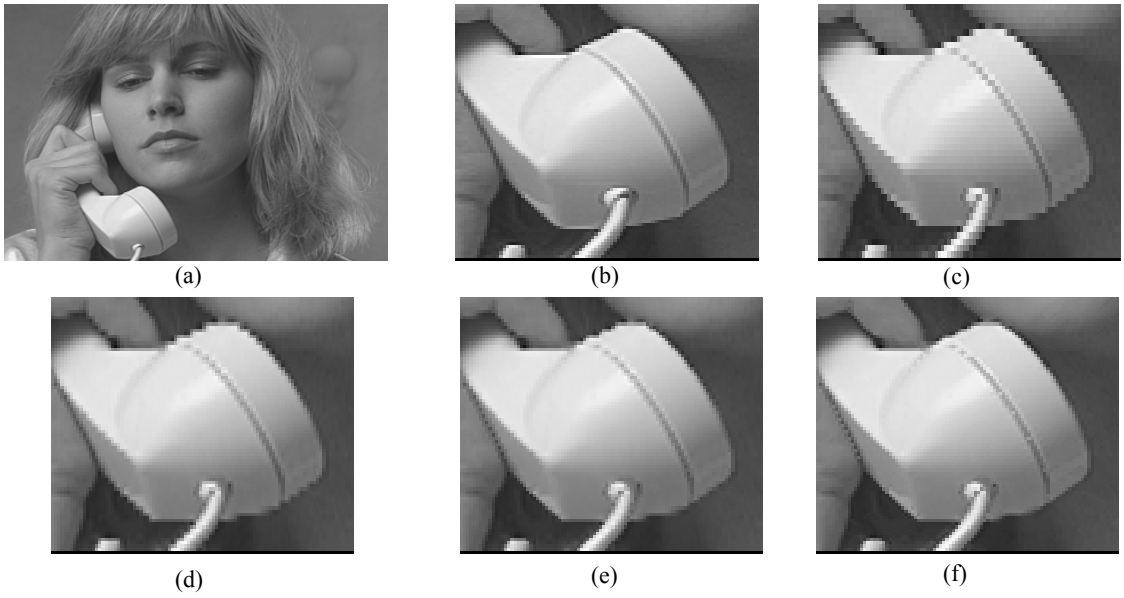

Fig. 5. (a) One original progressive frame from Susie sequence. (b) Zoom of the progressive frame. Zooms of the de-interlaced frame obtained by: (c) doubling line, (d) line average, (e) ELA and (f) fuzzy ELA.

IV. HARDWARE IMPLEMENTATION

The hardware implementation of the proposed algorithm has been realized on a FPGA. It has been implemented with two architectures which differ in the parallelism level employed to calculate the fuzzy rule's activation degrees. The parallel design requires a higher number of resources (three multipliers to implement (4) taking into account (2)) and provides a new pixel value every clock cycle. The number of the resources can be minimized by using a sequential architecture. It only requires one multiplier but three clock cycles to provide the result. Both implementations have been developed with SysGen (System Generator), which is a tool from Xilinx integrated in the Matlab environment. SysGen accelerates the design process since allows generating the VHDL description. Comparing the simulations of this design (running SysGen model in the Simulink environment) with those obtained from the algorithms programmed in Matlab language, the results are very similar. The block diagram of the sequential architecture is shown in Fig. 10.

In order to evaluate the complexity of the different algorithms a Virtex2 FPGA from Xilinx has been employed. It provides a wide variety of flexible features (block RAMS, multipliers), 500k system gates and an internal clock speed of 420 MHz. Table 3 shows the post-synthesis results in terms of area and speed of the implementations. The last row indicates the maximum resolution per frame that could be provided for

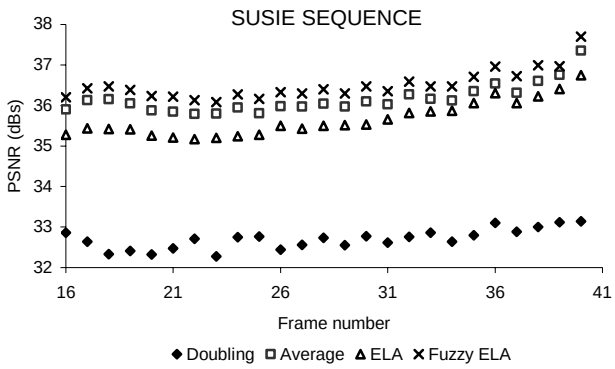

Fig. 6. PSNR values (in dBs) frame by frame in Susie sequence.

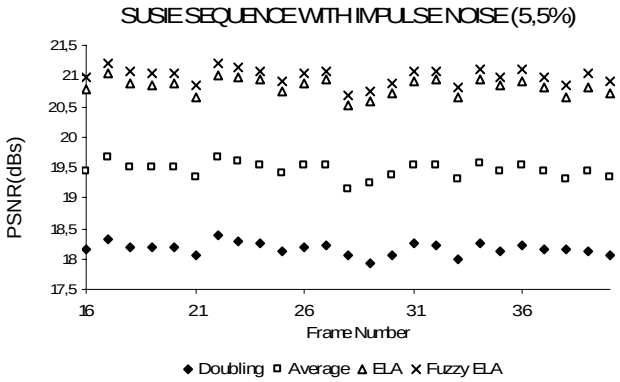

Fig. 7. PSNR values (in dBs) frame by frame in Susie sequence corrupted by impulse noise (5,5% density noise).

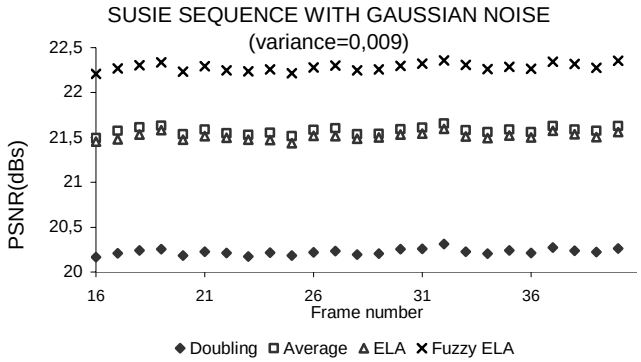

Fig. 8. PSNR values (in dBs) frame by frame in Susie sequence corrupted by Gaussian noise (variance=0,009).

frame sequences displayed at a rate of 30 per second. Comparing with the implementations of the original ELA the increase in consumed slices is low (especially in the sequential implementation, which only includes one multiplier and an accumulator). In terms of processing time the increase is also low, and, as shown the last row of Table 3, both fuzzy ELA implementations achieve resolution over HDTV.

V. CONCLUSIONS

The fuzzy ELA algorithm proposed in this paper works as well as ELA in avoiding the blurring and staircase effects that the line average method introduces in the edges of the images but improves the ELA results in the rest of the image where there are no edges or they are not clear. In addition, the proposed algorithm also performs better in de-interlacing images corrupted by noise.

These advantages are obtained at expense of a low increase in computational cost so that its hardware implementation is very simple. This has been proven by extensive simulation results with static images, image sequences, and noisy images.

REFERENCES

[1] T. Doyle and M. Looymans, "Progressive scan conversion using edge information", *Proc. 3rd Int. Workshop on HDTV*, pp. 711-721, Torino, Italy, Aug. 1989.

[2] T. Doyle, "Interlaced to sequential conversion for EDTV applications", *Proc. 2nd Int. Workshop Signal Processing of HDTV*, pp. 412-430, L'Aquila, Italy, Feb. 1988.

[3] 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, n. 2, pp. 119-129, May 1994.

[4] 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, n. 3, pp. 317-321, June 1996.

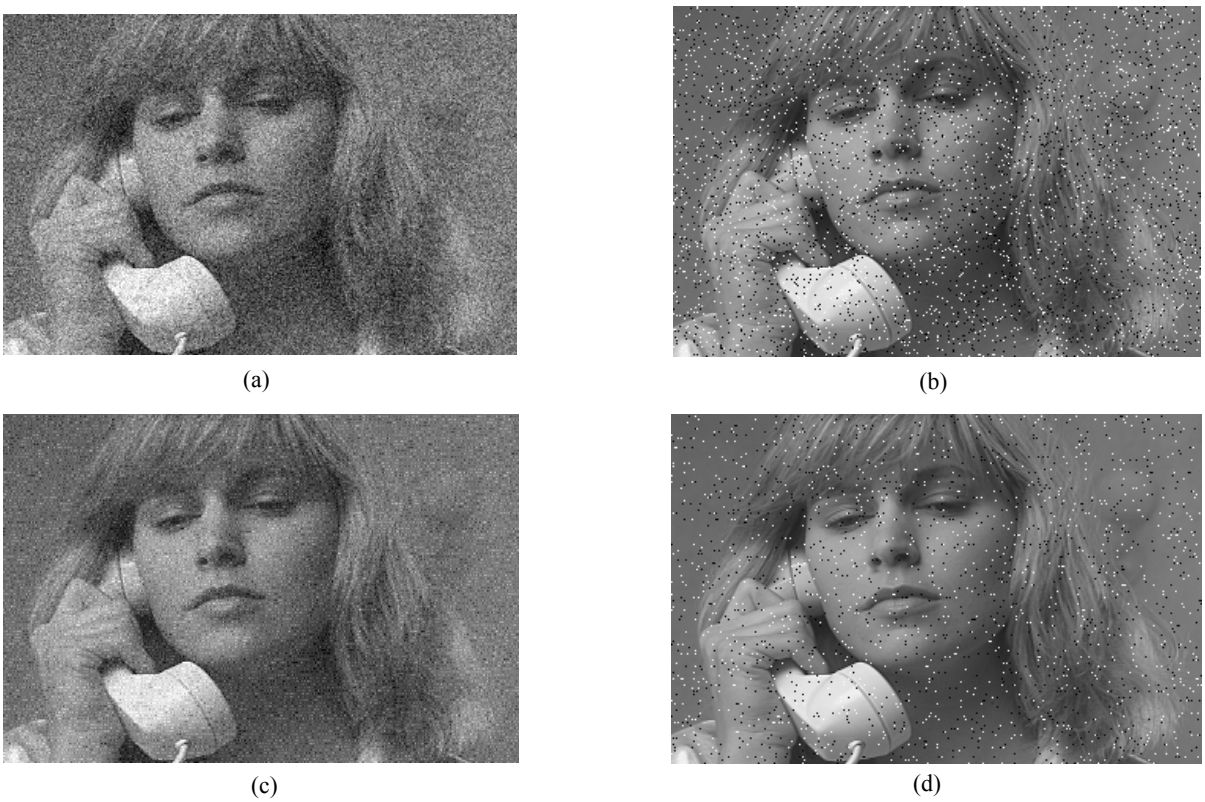

Fig. 9. (a) Original progressive frame with Gaussian noise and (b) the de-interlaced frame obtained with fuzzy ELA method. (c) Original progressive frame with impulse noise and (d) the de-interlaced frame obtained with fuzzy ELA method.

[5] 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, n. 4, pp. 999-1006, Nov. 2000.

[6] G. De Haan and R. Lodder, "De-interlacing of video data using motion vector and edge information", *Proc. IEEE Int. Conf. on Consumer Electronics (ICCE)*, pp. 70-71, Los Angeles, USA, June 2002.

[7] Y.L. Chang, S.F. Lin and L.G. Chen, "Extended intelligent edge-based line average with its implementation and test method", *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, vol. 2, pp. II-341-344, Vancouver, Canada, May 2004.

[8] H. Yoo and J. Jeong, "Direction-oriented interpolation and its application to de-interlacing", *IEEE Trans. on Consumer Electronics*, vol. 48, n. 4, pp. 954-962, Nov. 2002.

[9] M. Sugeno and T. Yasukawa, "A fuzzy-logic-based approach to qualitative modeling", *IEEE Trans. Fuzzy Systems*, vol. 1, n. 1, pp. 7-31, Feb. 1993.

TABLE 3

POST-SYNTHESIS RESULTS OF THE FPGA IMPLEMENTATIONS

FPGA Virtex 2 xc2v500_6fg256				
	ELA		FUZZY_ELA	
	Sequential	Parallel	Sequential	Parallel
No. Slices	60 (1,95%)	77 (2,5%)	283 (9,2%)	504 (16,4%)
Processing time (ns)	10,35	5,89	17,73	7,14
Processing frequency (MHz)	96,61	169,8	56,49	140,06
Maximum resolution (pixels)	6.441.222	11.318.619	3.760.104	9.337.068

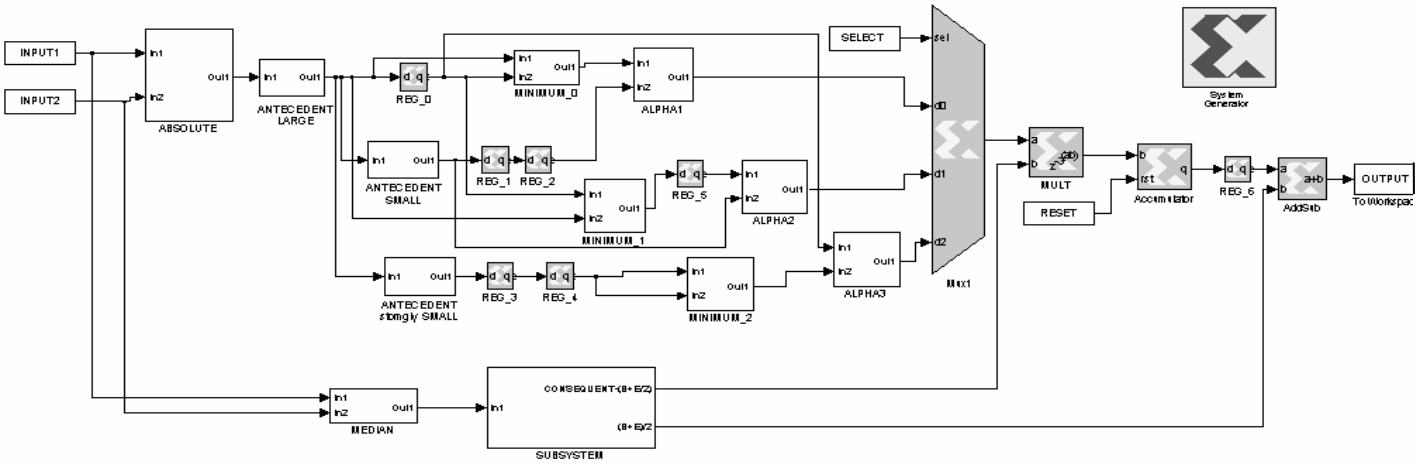

Fig. 10. SysGen sequential architecture for fuzzy ELA