Available online at www.elixirpublishers.com (Elixir International Journal)

Electrical Engineering

Elixir Elec. Engg. 93 (2016) 39672-39675

ISSN: 2229-712X

VLSI implementation and performance evaluation of adaptive filters for impulse noise removal

B.Elizabeth Caroline, G.Sheeba, J.Jeyarani and F.Salma Rosline Mary

Electronics and Communication Engineering, J.J. College of Engineering and Technology, Trichy – 09, India.

ARTICLE INFO

Article history:

Received: 24 July 2012;

Received in revised form: 14 April 2016;

Accepted: 20 April 2016;

Keywords

Adaptive Median Filter,
Adaptive Rank Order Filter,
Impulse Noise,
Image Enhancement Factor and Peak,
Signal to Noise Ratio.

ABSTRACT

An efficient VLSI implementation of Adaptive Rank Order Filter (AROF) and Adaptive Median Filter (AMF) is proposed in this paper. Impulse noise is introduced in digital images during image acquisition and transmission. The noise reduction algorithms remove noise without degrading image information. Linear filters tend to blur an image; hence they are not commonly used. Non-linear filters provide more satisfactory results in comparison to linear filters. The proposed paper adapts the filter based on the level of noise intensity in the image. AMF provides better filtering properties than standard median filters for images corrupted with 60% noise density. AROF provides better filtering properties than it is possible with AMF for images corrupted with higher noise densities (>60%). The VLSI architecture for AROF and AMF implements pipelining with parallel processing in order to speed up the filtering process. The performance of the proposed algorithm is compared with Peak Signal to Noise Ratio (PSNR) and Image Enhancement Factor (IEF).

Introduction

Digital images are usually transmitted through satellites, wired and wireless networks. An additive noise process may corrupt these digital images in both the acquisition and transmission stages. Image noise is the random variation of brightness or colour information in images produced by the sensor and circuitry of a scanner or digital camera. Due to the imperfections of image sensors, images are often corrupted by noise. The impulse noise is the most frequently referred type of noise. Two common types of impulse noise are the salt and pepper noise (commonly referred to as intensity spikes or speckle) and the random-valued impulse noise. For images corrupted by salt and pepper noise, the noisy pixels can take only the maximum (i.e. 255) or minimum (i.e. 0) values in the dynamic range. It is very difficult to remove this type of noise using linear filters because they tend to degrade the image quality of the resulting images.

Impulse noise removal in image processing often involves the removal of salt and pepper noise from images and it is a very important pre-processing step for most other subsequent processing tasks such as edge detection, segmentation and classification. Linear filters became the most popular filters in image signal processing. But, there exist many areas in which the nonlinear filters provide significantly better results. The advantage of nonlinear filter lies in their ability to preserve edges and suppress the noise without loss of details.

Impulse Filters – An Overview

Impulse noise is removed by a median filter which is the most popular nonlinear filter. The hardware implementation of median filter does not require many resources. But, the standard median filter gives a poor performance for images corrupted by impulse noise with higher intensity. A simple median filter utilizing 3x3 or 5x5-pixel window is sufficient only when the noise intensity is less than approx. 10-20%. When the intensity of noise is increasing, a simple median filter remains many shots unfiltered. Thus more advanced techniques have to be utilized.

Various approaches proposed to overcome this shortage are Switching Median Filter [1], Weighted median filters [2] and Adaptive Median filters [3]. The switching median filter, weighted median filter and weighted order statistic filters do not use the concept of a small filtering window. Therefore, their hardware implementations do not bring any benefits for a reasonable cost.

All alternatives to median filters have already been implemented in hardware [4], [5], [6], [7], [8], [9]. But, their performance is poor compared to AMF and AROF.

Proposed Method

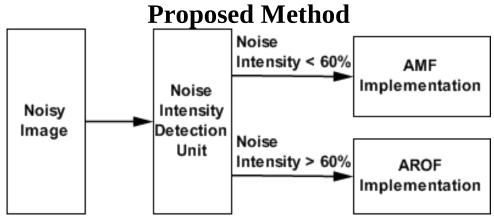

Fig.1 Block Diagram of Proposed Method

The proposed method adapts the filter based on the level of noise intensity in the image as shown in Fig.1. The proposed method provides hardware implementation of AROF if noise intensity is greater than 60% in Noise Intensity detection unit. Otherwise, the proposed method provides hardware implementation of AMF if noise intensity is within 60%. The Adaptive Rank Order Filter provides better filtering properties for images corrupted with higher noise intensity (>60%), while the Adaptive Median Filter provides better filtering properties for images corrupted within 60% noise intensity.

Adaptive Rank Order Filter

The AROF [11] provides better filtering properties than it is possible with AMF for images corrupted with high noise intensities. This method not only removes salt and pepper noise but also improves the visual quality of an image even at higher

Tele:
E-mail address: jeyaranijeyaseelan@gmail.com

noise densities. The improved performance of the AROF is measured by Peak Signal to Noise Ratio (PSNR) and Image Enhancement Factor (IEF). The goal is to provide a filter suitable for a high performance real-time processing of images corrupted by impulse noise of various intensities.

The proposed scheme overcomes the drawback of the AMF. The reconstructed images using AMF are generally not satisfactory for higher noise densities. For a fixed window size, there may be some pixels which are not noisy, when the median is noisy. In such case, if the centre pixel is replaced by some non-median pixel, then the filter becomes a kind of adaptive rank orders. The reconstructed image using this method provides better visual quality than that possible with AMF.

In AROF, the window adapts itself for two cases: (i) If all pixels within the current window are noisy. (ii) In order to replace centre pixel with non-median pixel if the median is noisy.

The proposed algorithm for AROF is as follows:

1. Read the image.
2. Select a window of size 3x3 at top left corner of the image.
3. Check if the centre pixel within the window is noisy. If yes, then go to step 4. Otherwise, slide the window to next pixel and repeat step 2.
4. Sort all pixels within the window in an ascending order and find the minimum p_{min}, median p_{med} and maximum p_{max}.
5. Determine if p_{med} is noisy by $p_{min}<p_{med}<p_{max}$. If it is true, p_{med} is not a noisy pixel and go to step 6. Otherwise, p_{med} is noisy pixel and go to step 7.
6. Replace the corresponding centre pixel in output image with p_{med} and got step 9.
7. If $p_{min}<p_{med}<p_{max}$ is false, check if all other pixels are noisy. If yes, then expand the window size by 2 and go back to step 4. Otherwise, go to step 8.
8. Replace corresponding centre pixel in output image with the noise free pixel which is the closest one to the median.
9. Reset window size and centre of window to next pixel.
10. Repeat the steps until all pixels are processed.

In order to apply AROF for colour image, the image is separated into red (R), green (G) and blue (B) colour components. The above algorithm is repeated for each of R, G and B colour components.

Adaptive Median Filter

The AMF [3] provides significantly better results especially for images corrupted with noise intensity less than 60%. The main advantage of AMF is that it modifies only corrupted pixels. But, the standard median filters modify almost all pixels of the image. Adaptive median filters can also be used as detectors of corrupted pixels.

AMF starts with an initial window size of 3x3 pixels with the following steps:

(1) Check if the centre pixel within the window is noisy or not. If the centre pixel is noisy, then sort pixels within window. Otherwise slide the window to next pixel and repeat this step.

(2) Check if the median is noisy or not. If the median is noisy then expand the window size and sort pixels within the window. This process is repeated until non-noisy median is found.

(3) Replace the centre pixel with the median value computed in step (2).

As the noise density increases, AMF employs a larger window to clean up the noise. But, the quality of the restored image degrades when larger window is employed. The hardware implementation of AMF [10] with filtering window 7x7 exhibits a very good performance/cost in comparison to standard median filters. But, this filter occupies approximately 30% of the chip area and is able to remove noise level up to 60%.

Hardware Implementation of AROF

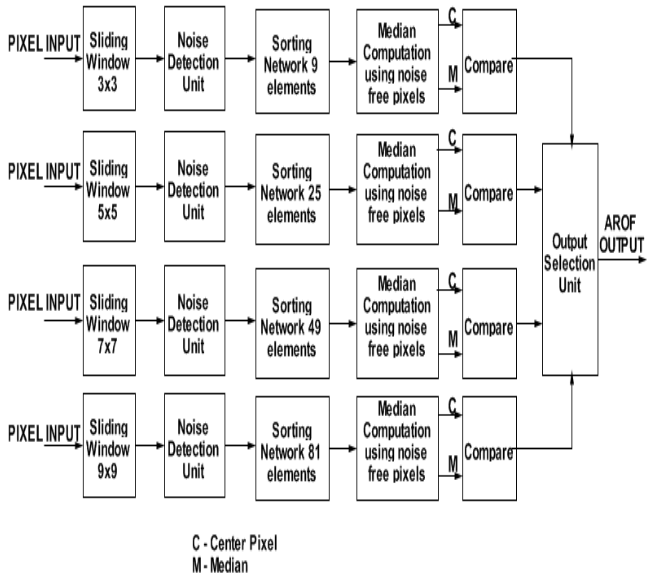

Fig.2 Hardware Implementation of Adaptive Rank Order filter

The hardware architecture of AROF as shown in Fig.2 comprises of five basic functional units – sliding window, noise detection, sorting network, median computation and the output selection unit. The image input data is placed in Random Access Memory (RAM) and placed into the sliding window module.

Sliding Window Module

Spatial filters operate with pixel values in the neighbourhood of the centre pixel called as filter window or observation window. The local neighbourhood function is implemented and it is called as a sliding window module. This sliding window module as shown in Fig.3 is applied independently on all pixel locations and is typically invariable for all locations (i.e. spatially invariant).

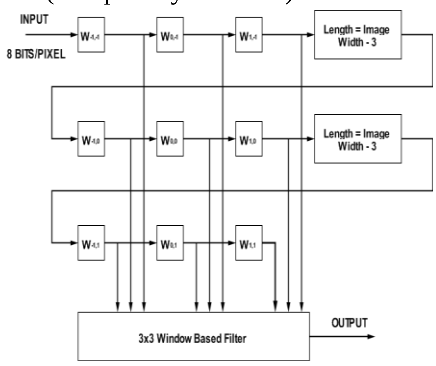

Fig.3 Implementation of 3x3filter window

The pixel values of the input image of width 8-bits are imported serially into the sliding window module. The hardware architecture of the sliding window module uses the row buffers. In this approach, one image pixel is read from memory in one clock cycle. The pixels are read row by row in a raster scan order. For a 3x3 sliding window, two First-In-First-Out (FIFO) buffers are used. The FIFO buffers are used to reduce the memory access to one pixel per clock cycle. The depth of the FIFO buffer is chosen as (W-3), where W is the width of the image. In order to access all values of the window for every clock cycle, the two FIFO buffers must be full.

The architecture places the lowest demand on external memory bandwidth, but the highest demand on internal memory bandwidth. This approach does not cause problems, as FPGA devices contain large amount of embedded memory.

Noise Detection Unit

The pixels from the sliding window are placed into the noise detection unit. The noise detection unit checks for noisy pixels within the window. If the pixels within the window are 0 or 255, then it is considered as salt and pepper noise. The noise detection unit generates output as zero for noisy pixel input. The output of the noise detection unit is fed into the sorting network.

Sorting Network

Sorting network consists of compare & swap elements called as comparators, that sorts all input pixels. A compare & swap unit of two elements (A,B) as shown in Fig.4 compares A and B and exchanges the elements in order to obtain the sorted sequence.

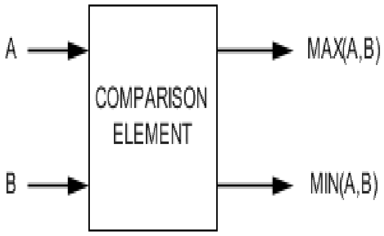

Fig.4 Symbol for a Comparison element

A sequence of compare & swap operations depends only on the number of elements to be sorted, not on the values of the elements. The main advantage of the sorting network is that the sequence of comparisons is fixed. Sorting networks are best suitable for parallel processing and pipelined hardware implementation. In hardware, sorting network is implemented using the Median Sorter module as shown in Fig.5.

The Median Sorter module uses the compare and swap elements to sort the input pixels. The Median Sorter module is based on the parallel sorting strategy. The Median Sorter module arranges the pixels in the descending order.

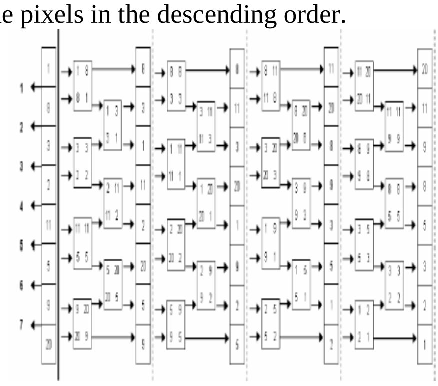

Fig.5 Median Sorter Module

The number of compare and swap operations depends on the number of elements to be sorted. The Median Sorter module requires 36 comparators (9 stages of 4 comparators each stage) for a 3x3 region, 300 comparators (25 stages of 12 comparators each stage) for a 5x5 region, 1176 comparators (49 stages of 24 comparators each stage) for a 7x7 region and 3240 comparators (81 stages of 40 comparators each stage) for a 9x9 window.

Median Computation Unit

The task of the median computation unit is to compute the median value from noise free pixels. The noise detection unit discussed above converts all noisy pixels into 0 if its value is equal to 255. However the pixels with 0 value remains as zero. The median computation unit operates after the sorting unit, therefore the noisy pixels or pixels with 0 values are ignored by this module and computes median from the noise free pixels.

The Output Selection Unit

The output selection unit is implemented in the hardware to replace the corrupted pixel with the median value and to retain the original pixel value if the pixel is uncorrupted.

Hardware Implementation of AMF

The hardware architecture of AMF is similar to the architecture of AROF. The hardware implementation of AMF as shown in Fig.6 consists of sliding window, sorting network, median computation unit and output unit. The AMF provides best results only for images corrupted with noise intensity less than 60%. The main difference between AMF and AROF lies in the computation of median value. The AMF uses all pixel values within the window to compute the median value while the AROF uses only the noise free pixels within the window.

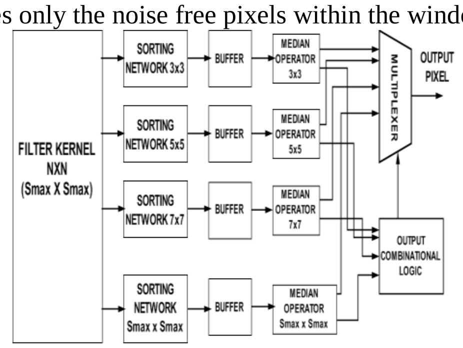

Fig.6 Hardware Implementation of Adaptive Rank Order filter

The Adaptive Median Filter operates with a kernel of Smax x Smax pixels. Let Sxy denote the centre pixel to be processed. The kernel is processed by a set of SNs with 3x3, 5x5, 7x7 and 9x9 inputs. Each Sorting network provides the minimum, maximum and median value. Buffers are used to synchronize the outputs of all Sorting Networks and Sxy. The implementation is pipelined and tries to minimize the number of stages. The output block (which is a simple combinational circuit) selects the output value according to the algorithm of Adaptive Median Filter.

Experimental Results

The proposed Adaptive filter is implemented in Verilog, simulated using Xilinx ISE Simulator and synthesized using Xilinx ISE tools version 9.2i to Vertex XC2VP50-7 FPGA device. Fig.7 shows the AMF and AROF results for Lena image corrupted with 70% noise intensity.

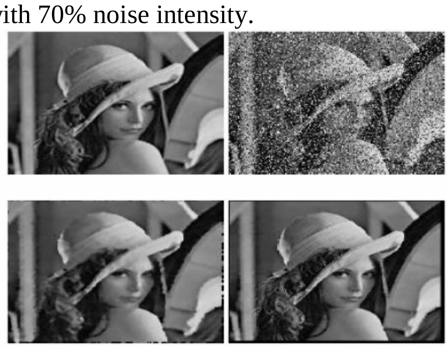

Fig.7 Results of filtering with adaptive median filter and adaptive rank order filter. First row : Original image, noisy image; Second row : Adaptive median filter, Adaptive Rank Order filter

The AROF and AMF filters are implemented in FPGA and the simulation results for 3x3 window is as shown in Fig.8

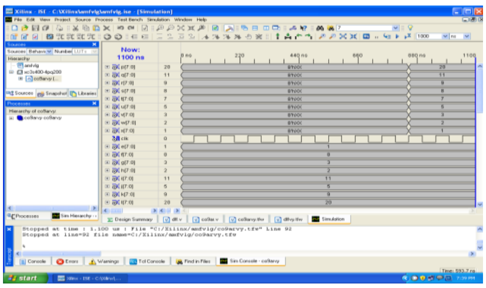

Fig.8(a) Median Sorter Module for 3x3 window

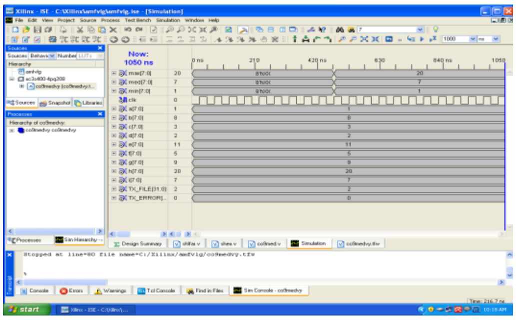

Fig.8(b) Median Computation for 3x3 window

Objective comparisons on the performances of the proposed filter on image corrupted to various levels of impulse noise ratios are made with the Mean Absolute Error (MAE) values, the Peak Signal to Noise Ratio (PSNR) and Image Enhancement Factor (IEF) of the image restored by them.

The PSNR is given by

$$PSNR = 10\log_{10}[255^2/MSE] \quad (1)$$

The Mean Square Error is given by

$$MSE = \sum_{x=1}^{p}\sum_{y=1}^{q}\frac{[E(x,y)-I(x,y)]^2}{pq} \quad (2)$$

Where E(x,y) is the enhanced gray pixel at position (x,y), I(x,y) is the original gray pixel at position (x,y) and, p and q denote the size of the gray image.

The IEF is expressed as:

$$IEF = \sum_{x=1}^{p}\sum_{y=1}^{q}\frac{[n(x,y)-I(x,y)]^2}{[f(x,y)-I(x,y)]^2} \quad (3)$$

n(x,y) is the noisy image, I(x,y) is the original image and f(x,y) is the filtered image.

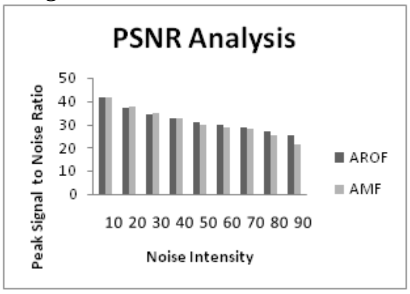

Fig.9 (a) PSNR analysis for AMF and AROF

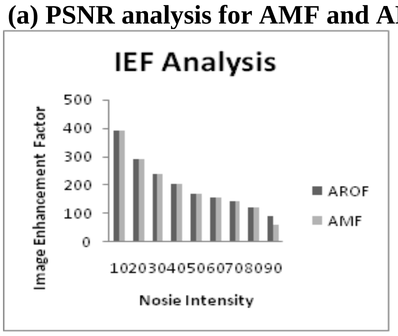

Fig.9 (b) Image Enhancement Factor analysis for AMF and AROF

The objective improvement of the proposed AROF is established by comparing the restored images of AROF with AMF as shown in Fig.7. The better PSNR and IEF for the restored Lena image is obtained for images corrupted with high noise intensity as shown in Fig.9.

The implementation costs are expressed in terms of number of slices. Table I provides synthesis results for AROF and AMF.

Conclusion

In this paper, a new FPGA implementation of AROF and AMF for removal of impulse noise is proposed. The filter logic is implemented on a novel reconfigurable device. The architecture provides the capability of implementing the reconfigurable fabric in a pipelined fashion and also includes the implementation of the pipelined version of the filter for improved speed. Adaptive Rank Order Filter provides better filtering properties than AMF for higher noise densities. The experimental results show that the restored images with higher noise densities based on AROF are better in visual quality than that is achieved using AMF.

Table I : SYNTHESIS RESULTS

Number of Inputs	Number of Slices	
	AMF	AROF
9	268	446
25	1506	2568
49	4815	8468
81	10315	15590

The performance of the proposed method is verified by applying peak signal to noise ratio and image enhancement factor. Future work involves implementation of AROF and AMF filters on XUPV5-LX110T, a feature-rich general purpose evaluation and development platform with on-board memory and industry standard connectivity interfaces.

References

[1] W. Zhou and Z. David, “Progressive switching median filter for the removal of impulse noise from highly corrupted images,” IEEE Trans On Circuits and Systems: Analog and Digital Signal Processing, vol. 46, no. 1, pp. 78–80, 1999.

[2] D. R. K. Brownrigg, “The weighted median filter,” Commun. ACM, vol. 27, no. 8, pp. 807–818, 1984.

[3] H. Hwang and R. Haddad, “Adaptive median filters: new algorithms and results,” IP, vol. 4, no. 4, pp. 499–502, April 1995.

[4] S. A. Fahmy, P. Y. K. Cheung, and W. Luk, “Novel fpga-based implementation of median and weighted median filters for image processing,” in FPL, T. Rissa, S. J. E. Wilton, and P. H. W. Leong, Eds. IEEE, 2005, pp. 142–147.

[5] D. Caban, “FPGA implementation of positional filters,” in Design of Embedded Control Systems. Springer-Verlag, 2005, pp. 243–249.

[6] R. Maheshwari, S. S. S. P. Rao, and E. G. Poonach, “FPGA implementation of median filter,” in VLSI Design. IEEE Computer Society, 1997, pp. 523–524.

[7] C. Chakrabarti, “Sorting network based architectures for median filters,” Transaction on Signal Processing, 1994.

[8] P. Lakamsani, R. Yang, B. Zeng, and M. Liou, “Design and implementation of a programmable stack filter,” in ICIP94, 1994, pp. 664–667.

[9] C. Chakrabarti, “Novel sorting network-based architectures for rank order filters,” IEEE Transactions on Very Large Scale Integration Systems, vol. 2, no. 4, pp. 502–507, 1994.

[10] Zdenek Vasicek, Lukas Sekanina, “Novel Hardware Implementation of Adaptive Median Filters”, IEEE,2008.

[11] M.C.Hanumantha Raju, M.Ravishankar, D.R.Ramesh Babu and S.B.Satish, “An efficient VLSI Architecture for Adaptive Rank Order Filter for Image Noise Removal”, International Journal of Information and Electronics Engineering, July 2011.

[12] Lu, Yan Dai, Ming Jiang and Shi, “Sort optimization algorithm of median filtering based on FPGA”, International Conference Machine Vision and Human-Machine Interface (MVHI), pp.250-253, 24-25 April, 2010.

[13] Meena S.M and Linganagouda K, “Rank based merge sorting network architecture for 2D median and morphological filters”, IEEE International Conference on Advance Computing pp.473-479,6-7 March,2009.