ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

VLSI Architecture for Efficient Removal of Random-Valued Impulse Noise in Images

[1]Keerthana, M., [2]Arivu Selvam, B., [3]Sudha, S.

[1]PG Student, Easwari Engineering College, Department of Electronics and Communication Engineering, Chennai, India.
[2]Assistant Professor, Easwari Engineering College, Department of Electronics and Communication Engineering, Chennai, India.
[3]Professor, Easwari Engineering College, Department of Electronics and Communication Engineering, Chennai, India.

ARTICLE INFO

Article history:
Received 10 March 2015
Received in revised form 20 March 2015
Accepted 25 March 2015
Available online 10 April 2015

Keywords:
Image denoising, impulse noise, median filter, impulse detector, image processing.

ABSTRACT

Median Filters is a novel filter structure which gains its significance due to its edge preserving property. Speed of the filter plays a significant role in the filter design. Decision tree algorithm defines the algorithm to identify the pixel which is noisy in the image. Based on the level of the input noise level in the image decisions will be taken. When the noise level in the image is more, then the image is passed through the edge preserving filter to remove the noises. Impulse detector is used to find the noise pixels in the images. Edge preservation filter is used to remove the noise from the image. Median filter structure is used as the edge preservation filter in this architecture. PSNR values are compared with the existing filters. In the new proposed architecture the speed of the filter got increased significantly when compared to the existing filter structures.

To Cite This Article: Keerthana, M., Arivu Selvam, B., Sudha, S., VLSI Architecture for Efficient Removal of Random-Valued Impulse Noise in Images. ***Aust. J. Basic & Appl. Sci.,*** *9(15): 136-141, 2015*

INTRODUCTION

Imaging processing used in machine vision technology to identify the flaws occurring the manufactured products. Image obtained for machine vision are corrupted by the noise pixels which leads to escaping of flaws at output. Impulse noise frequently corrupts images during picture acquisition or transmission. Generally, two types of impulse noise exist: fixed-valued impulse noise and random valued impulse noise. The fixed value impulse noise (also known as salt-and-pepper noise) is usually reflected by a pixel which has either a minimum or a maximum value in grey-scale image. In contrast, the values of random-valued noisy pixels are distributed uniformly in the grey-scale image within the range of [0, 255]. The well known median filter removes impulse noise signals by changing the luminance value of the target pixel with the median value of those pixels in the filtering window.

The adaptive decision – tree based denoising method (DTBDM) and its VLSI architecture is used for the removal of random – valued impulse noise. Impulse noise often corrupts images in the imaging acquisition or transmission process. The noise may seriously affect the performance of image processing techniques, such as scanning techniques, image segmentation and face recognition. The standard median filter is a simple noise filter that attempts to remove impulse noise by changing the luminance value of the target pixel with the median value of those pixel values in the filtering window. Although the median filter is simple and provides a reasonable noise removal performance, it blurs image details and causes the useful information in the image to be lost. The most common solution to this problem is the switching median filter, which combines the median filter and the impulse detector. The impulse detector first determines whether a pixel in the image is corrupted or not. If the pixel is identified by the detector as a corrupted pixel, the pixel is sent to the median filter to remove the noise by taking median of neighboring pixels. The main advantage of these methods is that they employ an impulse detector to locate and filter the noisy pixels without processing the noise-free pixels.

In this paper we present a novel adaptive median filtering technique that achieves low complexity and high quality processing under low cost requirements on the image denoising hardware. We also present its FPGA implementation along with the images.

Architecture of Dtbdm:

DTBDM consists of two components: decision-tree-based impulse detector and edge-preserving image filter. The detector determines whether pi;j is a noisy pixel by using the decision tree and the correlation between pixel pi,j and its neighboring

Corresponding Author: Keerthana, M., Easwari Engineering College, Department of Electronics and Communication Engineering, Chennai.
E-mail:m.keerthanaece@gmail.com

pixels. If the result is positive, edge preserving image filter generates the reconstructed value. Otherwise, the value will be kept unchanged. The flow diagram of the DTBDM structure is shown in Fig. 1.

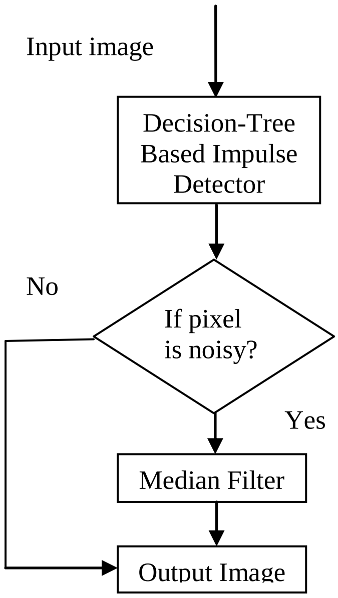


Fig. 1: Flow Diagram of DTBDM.

Decision-Tree-Based Impulse Detector:

The decision-tree-based impulse detector, has three modules—isolation module (IM), fringe module (FM), and similarity module (SM). Three concatenating decisions of these modules build a decision tree. The decision tree is a binary tree and can determine the status of pi;j by using the different equations in different modules. First, we use isolation module to decide whether the pixel value is in a smooth region. If the result is negative, we conclude that the current pixel belongs to noisy free. Otherwise, if the result is positive, it means that the current pixel might be a noisy pixel or just situated on an edge. The fringe module is used to confirm the result. If the current pixel is situated on an edge, the result of fringe module will be negative (noisy free); otherwise, the result will be positive. If isolation module and fringe module cannot determine whether current pixel belongs to noisy free, the similarity module is used to decide the result. It compares the similarity between current pixel and its neighboring pixels. If the result is positive, pi,j is a noisy pixel; otherwise, it is noise free. The following sections describe the three modules in detail.

Isolation Module:

Difference between the neighboring pixels in an image is small. Isolation point is a point where the difference between the two pixels values is large. Isolation point is obtained by observing the smoothness of the surrounding pixels. Maximum and minimum intensity pixels of top segment of the image are calculated using the first four pixels of the processing windows. The processing window is shown in Fig. 2. First four pixels in the processing window are selected and then values are compared with each other to obtain the top half minimum and maximum intensity pixel values. Similarly bottom four pixels are used to obtain the bottom half maximum and minimum intensity values. Difference between the center pixel and top half maximum and top minimum values are calculated. Similarly the difference between the centre pixel and bottom maximum and minimum values is obtained. Difference values obtained from the previous steps are compared with the threshold values. If the difference value exceeds the threshold value then the pixel value is considered to be a noisy pixel. The architecture of isolation module is shown in Fig. 3.

a	b	c
d	$f_{i,j}$	e
f	g	h

Fig. 2: A 3 X 3 Mask.

$$WTopHalf = \{a,b,c,d\} \quad (1)$$

$$WBottomHalf = \{a,b,c,d\} \quad (2)$$

$$Top\ Half\_diff = Top\ Half\_max - Top\ Half\_min \quad (3)$$

$$Bottom\ Half\_diff = Bottom\ Half\_max - Bottom\ Half\_min \quad (4)$$

$$\text{Decision I} = \begin{cases} \text{True}, & \text{if } (TopHalf\_diff \geq Th\_IMa) \text{ or } (BottomHalf\_diff \geq Th\_IMa) \\ \text{False}, & \text{otherwise} \end{cases} \quad (5)$$

$$IM\_TopHalf = \begin{cases} \text{True}, & \text{if } (|fi,j - TopHalf\_max| \geq Th\_IM_b) \text{ or } (|fi,j - TopHalf\_min| \geq Th\_IM_b) \\ \text{False}, & \text{otherwise} \end{cases} \quad (6)$$

$$IM\_BottomHalf = \begin{cases} \text{True}, & \text{if } (|fi,j - BottomHalf\_max| \geq Th\_IM_b) \text{ or } (|fi,j - BottomHalf\_min| \geq Th\_IM_b) \\ \text{False}, & \text{otherwise} \end{cases} \quad (7)$$

$$\text{Decision II} = \begin{cases} \text{True}, & \text{if } (IM\_TopHalf = true) \text{ or } (IM\_BottomHalf = true) \\ \text{False}, & \text{otherwise} \end{cases} \quad (8)$$

Australian Journal of Basic and Applied Sciences, 9(15) Special 2015, Pages: 136-141

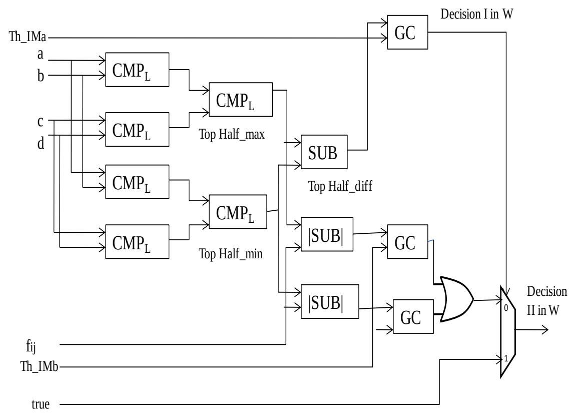


Fig. 3: Architecture of Isolation Module.

Fringe Module:

Fringe Module is used to identify whether the pixel is noisy pixel or the pixel present in the edges of the image. In order to determine whether the pixel is present in the edge the difference values are obtained in four directions. E1 difference is calculated using left wise diagonal pixels in the processing elements, E2 difference is calculated using right wise diagonal elements, E3 difference is calculated column wise center diagonal elements and E4 is difference is calculated row wise center diagonal elements. Diagonal wise difference values are greater than the threshold values then the pixel value is considered as the noisy pixel. The architecture for fringe module and FM_1 module is shown in Fig. 5. and Fig. 6. respectively.

$$FM\_E1 = \begin{cases} \text{False,} & \text{if } (|a-f_{i,j}| \geq Th\_FM_a) \text{ or } (|h-f_{i,j}| \geq Th\_FM_a) \text{ or } (|a-h| \geq Th\_FM_b) \\ \text{True,} & \text{otherwise} \end{cases} \quad (9)$$

$$FM\_E2 = \begin{cases} \text{False,} & \text{if } (|c-f_{i,j}| \geq Th\_FM_a) \text{ or } (|f-f_{i,j}| \geq Th\_FM_a) \text{ or } (|c-f| \geq Th\_FM_b) \\ \text{True,} & \text{otherwise} \end{cases} \quad (10)$$

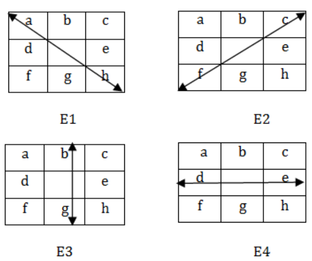


Fig. 4: Four Directions in DTBDM.

$$FM\_E3 = \begin{cases} \text{False,} & \text{if } (|b-f_{i,j}| \geq Th\_FM_a) \text{ or } (|g-f_{i,j}| \geq Th\_FM_a) \text{ or } (|b-g| \geq Th\_FM_b) \\ \text{True,} & \text{otherwise} \end{cases} \quad (11)$$

$$FM\_E4 = \begin{cases} \text{False,} & \text{if } (|c-f_{i,j}| \geq Th\_FM_a) \text{ or } (|f-f_{i,j}| \geq Th\_FM_a) \text{ or } (|c-f| \geq Th\_FM_b) \\ \text{Tr} \end{cases}$$

$$\text{Decision III} = \begin{cases} \text{False,} & \text{if } (FM\_E_1) \text{ or } (FM\_E_2) \text{ Or } (FM\_E_3) \text{ or } (FM\_E_4) \\ \text{True,} & \text{otherwise} \end{cases} \quad (13)$$

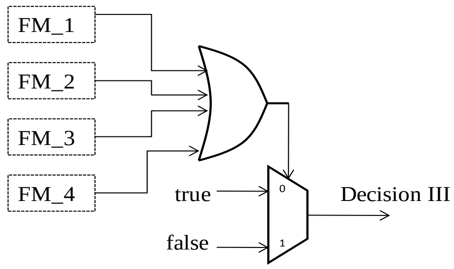


Fig. 5: Architecture of Fringe Module.

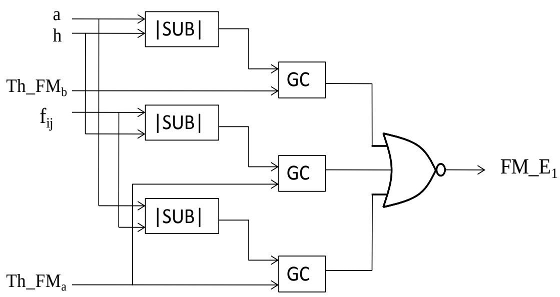


Fig. 6: Architecture of FM_1 Module.

Similarity Module:

The intensity values in the window are located in noise free area is closer in values. The median values are located in the center and the noisy pixels are located in the edges. A smaller and bigger value implies the possibility of noise signal. Nine pixel values are sorted to obtain fourth, sixth values and median values. Minimum and maximum are used to obtain the status of the pixel. Difference values are greater than the threshold values then the pixel value is considered as the noisy pixel. The architecture for similarity module is shown in Fig. 7.

$$Max_{i,j} = 6\text{th in } W_{i,j} + Th\_SM_a$$
$$Max_{i,j} = 4\text{th in } W_{i,j} - Th\_SM_a \quad (14)$$

$$Nmax = \begin{cases} Max_{i,j}, & \text{if } (Max_{i,j} \leq MedianInW_{i,j} + Th\_SM_b) \\ MedianInW_{i,j} + Th\_SM_b & \text{otherwise} \end{cases} \quad (15)$$

$$N_{min} = \begin{cases} Min_{i,j}, & \text{if } (Min_{i,j} \leq MedianInW_{i,j} - Th\_SM_b) \\ MedianInW_{i,j} - Th\_SM_b & \text{otherwise} \end{cases} \quad (16)$$

$$\text{Decision IV} = \begin{cases} \text{True}, & \text{if } (f_{i,j} \geq N_{max}) \text{ or } (f_{i,j} \geq N_{min}) \\ \text{False} & \text{otherwise} \end{cases} \quad (17)$$

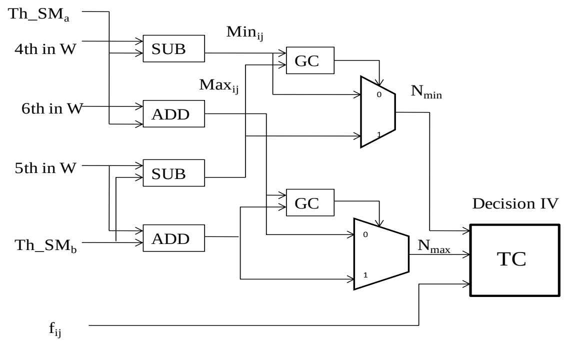


Fig. 7: Architecture of Similarity Module.

Filter Structure:

Filter structure is used to remove the noisy pixel in the image. Noisy pixel determined from the above steps is removed by using the filter structure. Noisy pixel is replaced with the median value calculated from the original image. Initially the values are sorted and the median value is obtained from the image. The architecture for sorting and M0 module is shown in Fig. 8. and Fig. 9. respectively.

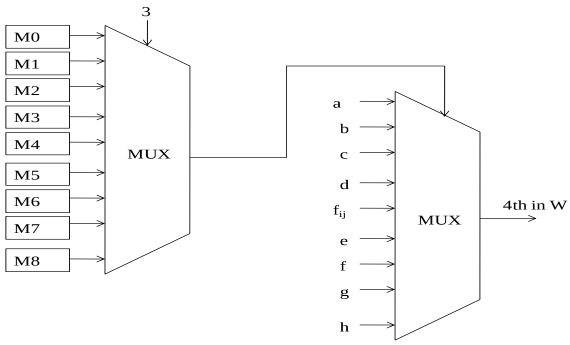


Fig. 8: Architecture of Sorting (4th in W).

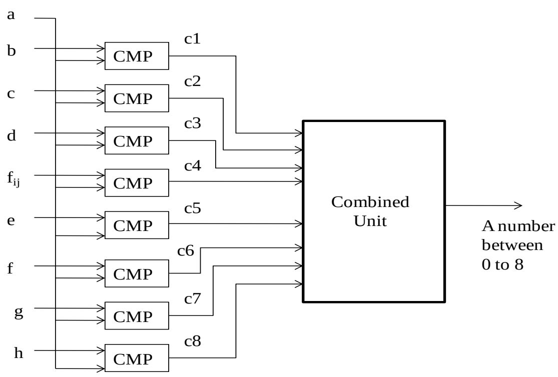
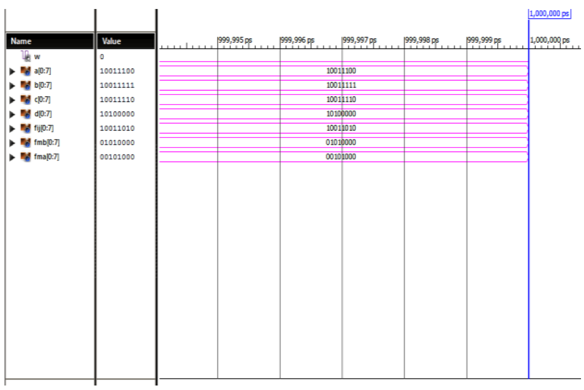


Fig. 9: Architecture of M0 Module.

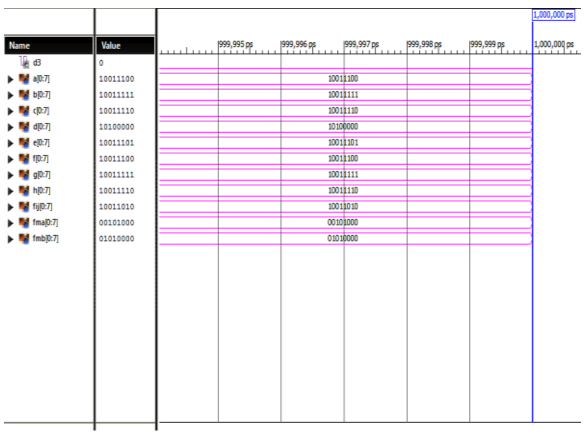
Results:

An image is given as input in MATLAB and the hex values are obtained and it is written as a text file. Test bench is written using HDL to read the hex values from the text file. The obtained hex values are passed as inputs to the Isolation module, Similarity module, Fringe module and the median filter to remove the noisy pixel. HDL coding are simulated using XILINX. The PSNR values are obtained from the noise free image.

ISOLATION MODULE

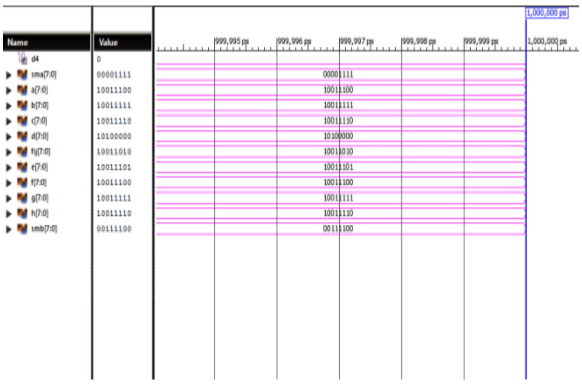


FRINGE MODULE

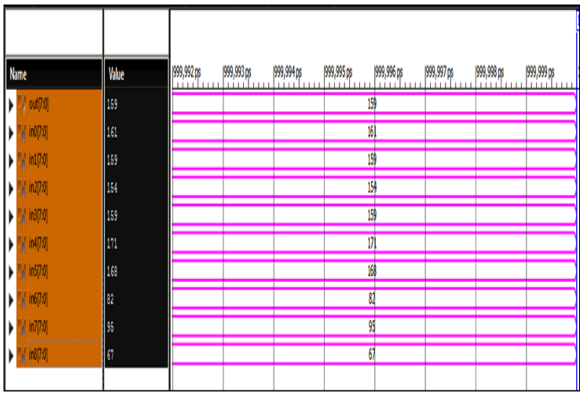


Australian Journal of Basic and Applied Sciences, 9(15) Special 2015, Pages: 136-141

SIMILARITY MODULE



MEDIAN FILTER




Input Image


Random Noise Added Image


Reconstructed Image

Conclusion:

PSNR values obtained from the above filter structure are better when compared to the existing filter structures. Efficiency of the filter structure is more even though the noise levels present in the image are high. The approach of DTDM is used to find the noise and also to detect the edge pixel efficiently.

ACKNOWLEDGEMENT

I would like to thank Dr.D.Ebenezer and Dr.S.Sudha for their support and encouragement. I would like to express my sincere gratitude to Mr.B.Arivu Selvam for his guidance in the course of my research.

REFERENCES

Abreu, E., M. Lightstone, S.K. Mitra and K. Arakawa, 1996. ‘A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images’, IEEE Trans. Image Processing, 5(6): 1012-1025.

Bednar, J.B. and T.L. Watt, 1984. ‘Alpha-trimmed means and their relationship to median filters’, IEEE Trans. Acoust., Speech, Signal Process., ASSP-32(1): 145-153.

Bin Yu, Martin Vetterli and Grace Chang, 2000. ‘Spatially Adaptive Wavelet Thresholding with Context Modeling for Image Denoising’, IEEE Transactions On Image Processing, 9(9): 1522-1531.

Boskovitz, V. and H. Guterman, 2002. ‘An Adaptive Neuro-Fuzzy System for Automatic Image Segmentation and Edge Detection’, IEEE Transactions on Fuzzy Systems, 10(2): 247-262.

Chen, P.Y. and C.Y. Lien, 2008. ‘An Efficient Edge-Preserving Algorithm for Removal of Salt-and-Pepper Noise’, IEEE Signal Processing Letters, 15: 833-836.

Chen, P.Y., C.Y. Lien and H.M. Chuang, 2010. ‘A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise’, IEEE Trans. Very Large Scale Integration Systems, 18(3): 473-481.

Chen, T. and H.R. Wu, 2001. ‘Adaptive Impulse Detection Using Center-Weighted Median Filters’, IEEE Signal Processing Letters, 8(1): 1-3.

Chih-Yuan Lien, Chien-Chuan Huang, Pei-Yin Chen, Member, IEEE, and Yi-Fan Lin, 2013. ‘An Efficient Denoising Architecture for Removal of Impulse Noise in Images’, IEEE Transactions On Computers, 62(4).

Choi, Y.S. and R. Krishnapuram, 1997. ‘A Robust Approach to Image Enhancement Based on Fuzzy Logic’, IEEE Transaction on Image Processing, 6(6): 808-825.

Hsia, S.C., 2003. ‘Parallel VLSI design for a real-time video-impulse noise-reduction processor’, IEEE Trans. Very Large Scale Integer (VLSI) Syst., 11(4): 651-658.

Irphan Ali Shaik, Mirzashafishahsavar, K.J. Silva Lorraine, 2012. ‘Impulse Noise Detection and Filtering Based on Adaptive Weighted Median

Australian Journal of Basic and Applied Sciences, 9(15) Special 2015, Pages: 136-141

Filter” in ISBN: 2319-6483, ISSN: 2278-4721, 1(8): 49-54.

Ko, S.J. and Y.H. Lee, 1991. ‘Center Weighted Median Filters and Their Applications to Image Enhancement’, IEEE Trans. Circuits Systems, 38(9): 984-993.

Matsubara, T., V.G. Moshnyaga and K. Hashimoto, 2010. ‘A FPGA Implementation of Low-Complexity Noise Removal’, Proc. 17th IEEE Int’l Conf. Electronics, Circuits, and Systems (ICECS ’10), pp: 255-258.

Neelavathi, P., 2012. ‘A New Detection Statistics for Removal of Impulsive Noise’, International Journal of Computer Application, 1(2): ISSN: 2250-1797.

Nodes, T. and N. Gallagher, 1982. ‘Median Filters: Some Modifications and Their Properties’, IEEE Trans. Acoustics, Speech, Signal Processing, ASSP-30(5): 739-746.

Sun, T. and Y. Neuvo, 1994. ‘Detail-Preserving Median Based Filters in Image Processing’, Pattern Recognition Letters, 15: 341-347.

Zhang, S. and M.A. Karim, 2002. ‘A new impulse detector for switching median filter’, IEEE Signal Process. Lett., 9(11): 360-363.