Comparative analysis of different algorithms of Median Filter with FPGA applications

[1]Bhavik A. Trivedi, [2]Jayesh Popat
[1]PG Student, [2]Assistant Professor
EC Department,
MEFGI, Rajkot, India
[1]bhavik.trivedi45@gmail.com , [2]jayesh.popat@marwadieducation.edu.in

***Abstract*— The selective median filter is a mixed filter which removes spike noise (or an impulse noise) from a noisy image while preserving sharp edges. For this purpose we can use a double derivative filter or Laplace filter which acts as a noise detector. If it is found noisy then we apply a general median filter. The median filter technique, the eight direct neighbours and centre point of a sliding 3-by-3 array are examined. A selective median filter which consumes less power can be designed and different logics for majority bit evaluation can be applied and simulate in VHDL. Different majority bit calculation method can be implement and the result sorting circuit can be analyze for power analysis and can be implement in FPGA like hardware.**

Keywords— selective; median filter; mixed filter; impulse noise; double derivative filter; Laplace filter; majority bit evaluation; FPGA; VHDL.

I. INTRODUCTION

Image processing is a wide area with tremendous applications including our everyday life such as medicines, space exploration, military surveillance, image authentication, automated industry inspection, automated control equipments and many more areas. In most of cases captured images from image sensors are affected by various types of noises. The impulse noise is among one of the most frequently referred type of noise.

This noise, commonly also known as salt & pepper noise, is caused by malicious pixels in camera sensors and in images, absurd memory locations in hardware, or errors in the data transmission and in many other forms affected by environment conditions.

Median filtering is considered a suitable method to remove impulse noises from images. This non-linear technique is a good substitute to linear filtering as it can efficiently subdue impulse noise while preserving edge information. The median filter operates for each pixel of the image and ensures that it fits with the pixels around it. It filters out samples that are not representative of their surroundings; in other words the impulses. Thus, median filter is very helpful in filtering out missing or damaged pixels of the image.

Median filter has complexity to implement, as large amount of data involved in representing image information in digital format. For easy implementation, general purpose processor like option can be used but it is not time efficient because of not time efficient.

In contrast to that, full custom hardware design for example, Application Specific Integrated Circuits (ASICs), is known for its providing best speed for various applications but at the same time they have very less scope for flexibility [1].

The fast growing demand for System on Chip (SOC) design requires much more developed. There is a heavy demand for reducing this power factors with same functioning capabilities. Most of these products include embedded microprocessors, Digital Signal Processors (DSPs) and ASICs.

Therefore low power design of any VLSI design is nowadays a challenging task. The median filter is widely used for noise reduction because it is more effective and less blurring than a linear smoothing filter when dealing with salt and pepper noise [6].

FPGA has been chosen because of its various features. FPGAs are reconfigurable devices, which enables simplified debugging and verification and rapid prototyping. Its parallel processing characteristic increases the speed of implementation [4].

In next section, basic of median filter is given, then proposed architecture reviews are given, then simulation results are shown, then conclusion is drawn and then further future work is depicted.

II. MEDIAN FILTER

Median filter can be categorized into spatial filter technique, so it can use 2-D mask which is applied to each pixel in the input image. The median value is determined by placing the brightness in ascending order and selecting the centre value [5].

The obtained median value will be the value for that pixel in the output image. Figure 1 shows one example for illustration [2].

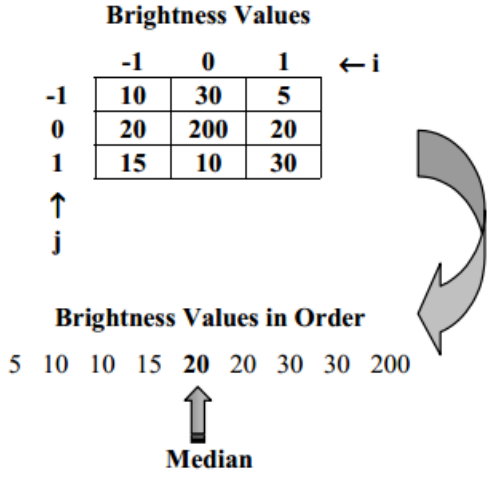

Figure 1: Illustration of median filter example

We use 3x3 window in median filter, as bigger size in window (5x5, 7x7, etc.) usually give lesser sharp edge than of 3x3 window median filter.

III. PROPOSED ARCHITECTURE OPTIMIZATION

We followed classic approach for sorting 9 pixels and selecting the median which proposes systolic array of optimized approach [2].

We have followed the technique shown in Figure 2, which is similar to the one proposed by Smith [3], and it uses only 19 nodes compare to other proposed in Vega-Rodríguez [2].

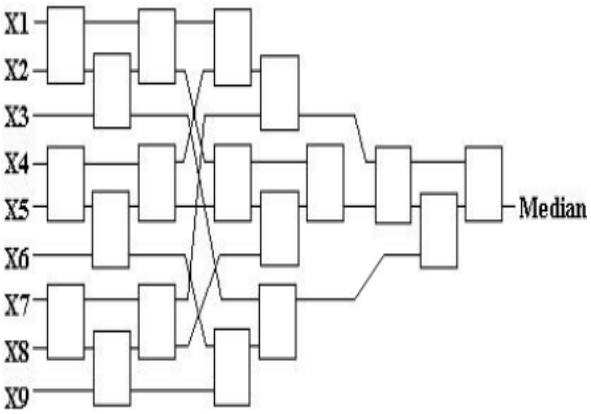

Figure 2: Proposed systolic array of optimized approach

Each basic node allows sorting two elements. To do that, each node compares two elements by means of an 8-bit comparator, using its output in two 2:1 multiplexers, as shown in Figure 3 [2].

We reviewed Bethina, Premkumar [7], and showed its flowchart of MDBUTMF (Modified Decision Based Unsymmetrical Median Filter) algorithm. Each and every pixel of the image is checked for the presence of salt and pepper noise.

MDBUTMF provides better Peak Signal-to-Noise Ratio (PSNR) than the existing methods. MDBUTMF replaces the noisy pixel by trimmed median value when some of the elements with values 0s and 255s are present in the selected window. If all the pixel values in the selected window are 0s and 255s means then the noisy pixel is replaced by mean value of all the elements present in that selected window [7].

Based on Figure 3, we can mentioned that, If processing pixel of 3x3 window has salt & pepper noise and neighboring pixel values contains all pixels that adds salt and pepper noise to the image, then mean of that window element is the answer of center pixel of that window.

If processing pixel's window contains all 0's or 255's or both than it simply convert to mean of all element of it to processing pixel, and if that window doesn't contain all 0's or 255's or both, that means contain some other values, then first that window's all 0's & 255's and processing pixel get removed and then median of remaining values is taken and that becomes final median value for that processing pixel.

Where "255" is processing pixel, i.e., (Pij), and that matrix is as shown as below:

$$\begin{pmatrix} 0 & 255 & 0 \\ 0 & <255> & 0 \\ 255 & 0 & 255 \end{pmatrix}$$

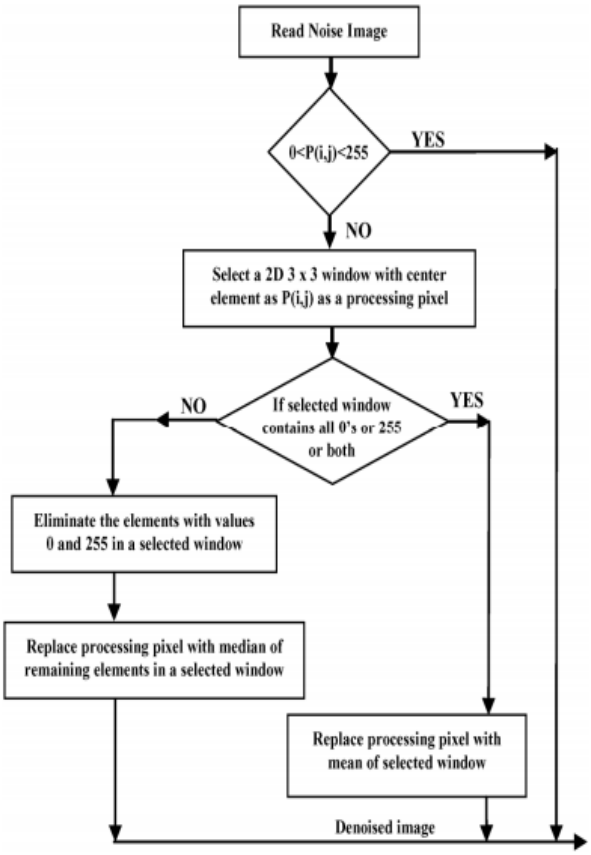

Figure 3: flowchart of MDBUTMF algorithm

Then comparison of different parameters of different algorithms can be take for analysis, as we showed here of in Table I [7].

Noise in %	PSNR in dB					
	MF	AMF	PSMF	DBA	MDBA	MDBUTMF
10	26.34	28.43	30.22	36.4	36.94	37.91
20	25.66	27.40	28.39	32.9	32.69	34.78
30	21.86	26.11	25.52	30.15	30.41	32.29
40	18.21	24.40	22.49	28.49	28.49	30.32
50	15.04	23.36	19.13	26.41	26.52	28.18
60	11.08	20.60	12.10	24.83	24.41	26.43
70	9.93	15.25	9.84	22.64	22.47	24.30
80	8.68	10.31	8.02	20.32	20.44	21.70
90	6.65	7.93	6.57	17.14	17.56	18.40

Table I: Comparison of PSNR for different median filter algorithms

As shown in Table I, different algorithms like Median Filter (MF), Adaptive Median Filter (AMF), Progressive Switched Median Filter (PSMF), Decision Based Algorithm (DBA), Modified Decision Based Algorithm (MDBA) and MDBUTMF's PSNR values for different noise densities for particular image is shown.

We developed MATLAB code in an efficient way for median filter. We used MATLAB R2012a for this purpose.

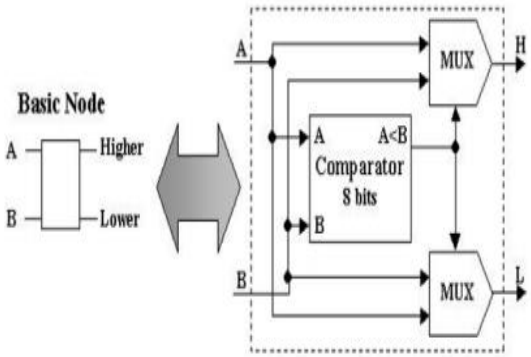

Figure 3: Scheme for each basic node

We mentioned our developed MATLAB code's algorithm steps with flowchart and showed schematic way of scanning 3x3 window of median filter.

MATLAB code algorithm:

Step 1. Read an input image
Step 2. Define 3x3 array for 3x3 window to take 9 corresponding pixels from input image.
Step 3. In this step, all three rows scan in 3x3 window for finding maximum, median and minimum values from all three rows.
Step 4. Similarly all three columns then scan for finding maximum, median and minimum values.
Step 5. Now elements of 3x3 window scan in principle diagonal of 3x3 square window to find median from those three values.
Step 6. Similarly in other diagonal all three values scan except a center value in that diagonal replace by result of previous step.
Step 7. Now this 3x3 window continues to scan input image until all pixels of image scanned.
Step 8. Finally, we get output image without impulse noise by using median filter.

Figure 3 shows flowchart of our MATLAB code. It gives us one of the efficient way to get median value from 3x3 window.

In Figure 5, steps of scanning input image by 3x3 window of median filter is shown. Here, (d) shows that one of the value being scanned is output of (c), that means output of (c) (median value from n1, n5 & n9), will scan with n7 & n9. Here in (a), (b) & (c), it is assumed that after sorting values of nine pixel values, all values' positions are unchanged, for illustration purpose.

Flowchart of Developed MATLAB code for median filter

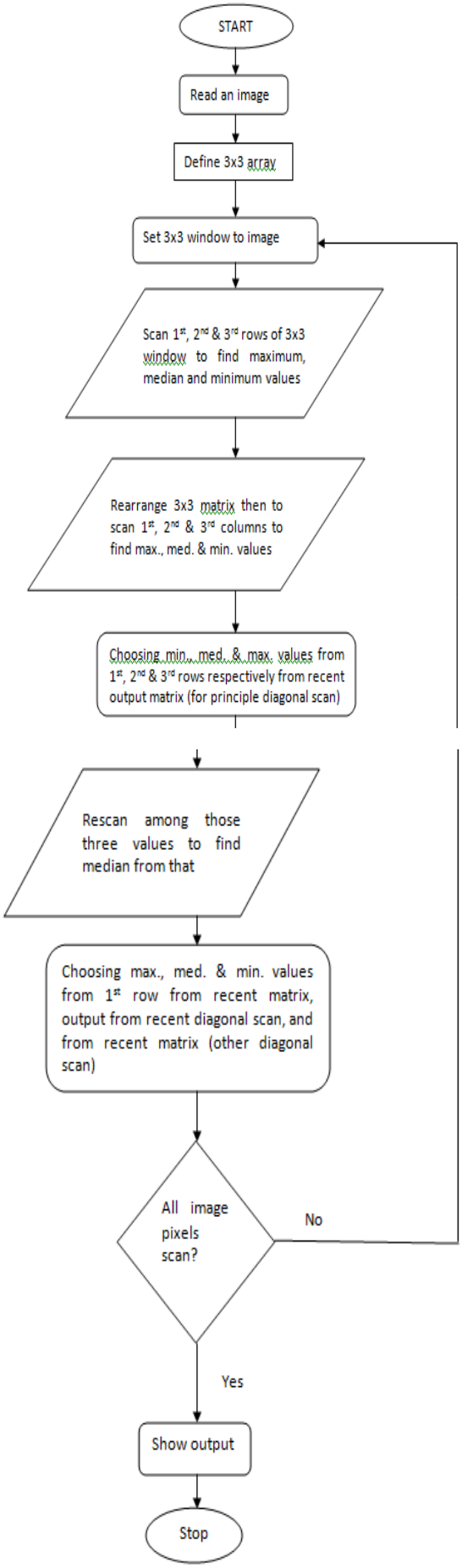

Figure 3: flow chart of median filter MATLAB Code

IV. Simulation Results

We get following result of simulation from our developed MATLAB code, as shown in figure 6 & 7.

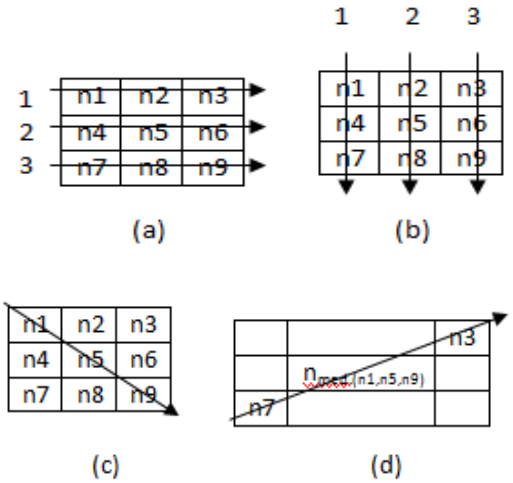

Figure 5: steps of scanning input image by 3x3 window

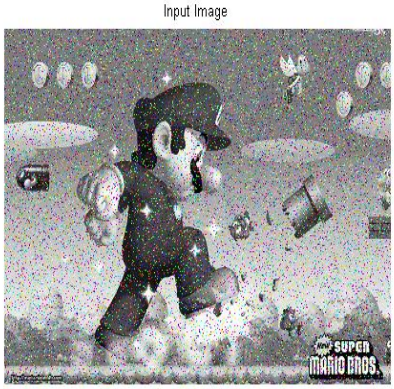

Figure 6: Input noisy image

Figure 7: Output image with filtered out impulse noise

So, as shown in figure 6, we used original image then manually distorted it with 10% of noise density of salt & pepper noise and then use our MATLAB code to smoothing out and we got relevant result, as shown in figure 7.

V. Conclusion

So, from figure 6,7 we can say that from developing MATLAB code and to optimizing it, instead of using software MATLAB code (medfilt2) for FPGA like hardware, we reach to better results for real-time applications.

Future Work

Our MATLAB developed median filter code can be used in FPGA implementation, as it is very useful for parallel processing and gives real-time results. We may follow following schematic diagram for our FPGA implementation of median filter.

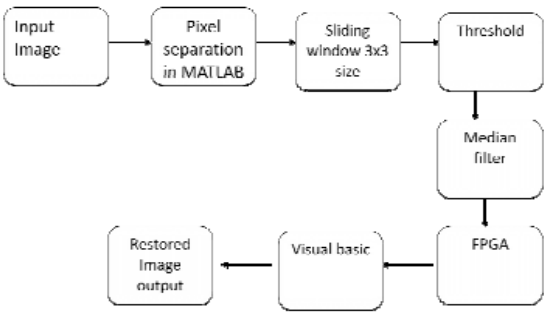

Figure 8: Basic schematic diagram of workflow of median filter implementation for FPGA using Visual Basic

References

[1] S.S. Tavse, P.M. Jadhav, M.R. Ingle, "Optimized median filter implementation on FPGA including soft processor.", International Journal of Emerging Technology and Advanced Engineering, Vol. 2, Issue 8, pp. 236-239, August 2012.

[2] Miguel A., Vega-Rodríguez, Juan M., Sánchez-Pérez, Juan A.Gómez-Pulido, "An FPGA-based implementation for median filter meeting the real-time requirements of automated visual inspection systems." Proc. Of 10th Mediterranean Conference of Control and Automation. - MED 2002 Lisbon, Portugal, July 9-12, 2002.

[3] J. L. Smith, "Implementing median filters in xc4000e FPGAs," XCell, vol. 23, no. 1, p. 16, 1996.

[4] Yueli Hu, Huijie Ji., "Research on image median filtering algorithm and its FPGA implementation." IEEE Global Congress on Intelligent Systems, 2009.

[5] G.A. Baxes, "Digital Image Processing Principles & Applications", Wiley & Sons, (1994).

[6] Xiaoyang Li, Jie Jiang, Qiaoyun Fan, "An improved real-time hardware architecture for Canny edge detection based on FPGA", 3rd Int. Conf. Intelligent Control and Information Processing, pp. 445-449, July 2012.

[7] Chaitanya Bethina, M. Premkumar, "An FPGA Implementation of Modified Decision based Unsymmetrical Trimmed Median Filter for the removal of Salt and Pepper Noise in Digital Images", IJESS, 2012.