Exhaustive Study of Median filter

[1]Anamika Sharma (sharma.anamika07@gmail.com), [2]Bhawana Soni (bhawanasoni01@gmail.com),
[3] Nikita Chauhan (chauhannikita39@gmail.com), [4]Rashmi Bisht (rashmi.bisht2000@gmail.com),
[*]Ritu Vijay(rituvijay1975@yahoo.co.in)
[1,2,3]M.Tech VLSI Design (students), [4]Researcher, *Banasthali University*
[*] *Associate Professor,, Banasthali University*

Abstract - Image filtering plays an important role to remove impulse (Salt and Pepper) noise from the images. The median filter which is a non-linear filter is very effective at removing noise while preserving image features and edges. In comparison to linear filter, they provide excellent noise reduction capabilities, with less blurring. The working of median filter is by removing the corrupted pixel value with the median value of neighboring pixel which is calculated by sorting all the pixel values from the window into ascending order. This paper presents a median filtering algorithm by using 3*3 windows.

Keywords – *Image filtering, Impulse noise (Salt and Pepper noise), Median filter, Window, Pixel, Matlab*

*****

I. Introduction

Image processing [4] is an ever expanding and dynamic area with applications reaching out into our everyday life such as medicine, space exploration, military surveillance, security and authentication, automated industry inspection, agriculture and many more. Applications such as these involve different processes like image enhancement, object detection and noise reduction. Implementing such applications on a general purpose computer can be easier, but not every time efficient due to additional constraints on memory and other peripheral devices. Application specific hardware implementation offers much greater speed than a software implementation. With advances in the VLSI (Very Large Scale Integrated) technology hardware implementation has become an attractive alternative. Implementing complex computation tasks on hardware and by exploiting parallelism and pipelining in algorithms yield significant reduction in execution times.

There are two types of technologies available for hardware design. Full custom hardware design known as Application Specific Integrated Circuits (ASIC) and semi custom hardware device. Semi custom hardware devices are programmable devices like Digital signal processors (DSPs) and Field Programmable Gate Arrays (FPGA's).

The importance and significance of image processing is to achieve better performance on hardware implementation, this work is done by image processing algorithms like median filter, convolution and smoothing operation and edge detection on FPGA using VHDL language. In recent times, Field Programmable Gate Array (FPGA) technology has become a reliable method for the implementation of algorithms suited to image processing applications because it provides the fast response as compare to other technologies. In recent years, VHSIC (Very High Speed Integrated Circuit) Hardware Design Language (VHDL) has become a sort of industry standard for high-level hardware design.

II. Filters

Image Filtering is used to remove noise [3, 5], sharpen contrasts, highlight contours, and detect edges. Image filters can be classified as linear and non-linear [1, 2].

A. Linear filters

Linear filters are also know as convolution filters as they can be represented using a matrix multiplication. It tends to blur edges and other image detail. Linear filter perform poorly with non-Gaussian noise. To overcome these disadvantages non-linear filters are used.

B. *Nonlinear filter*

Non linear filters can preserve edges and very effective at removing impulsive noise. In our work we used non-linear median filter.

- *Median filter*

Median filter is a non-linear filter. It sorts the pixels value in the neighborhood and then replaces the central pixel with the median value in the sorted group and it is known as best order-statistics filter [9].

$$f(x, y) = \underset{(s,t)\in S_{xy}}{\text{median}}\{g(s, t)\}$$

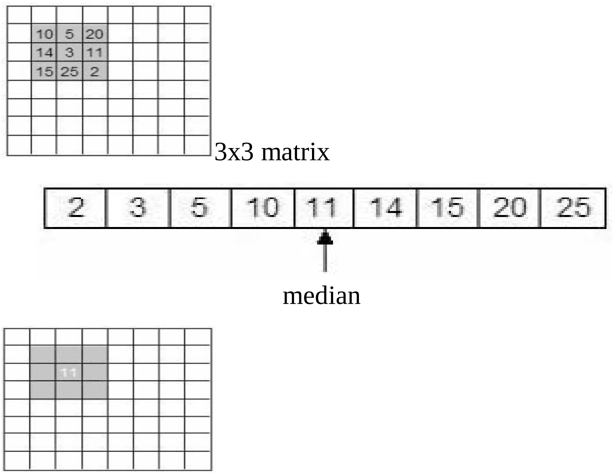

Figure-1 Representation of median value

Median filter plays an important role in image preprocessing applications and also in image edge preservation. So, in order to maintain a balance between noise removal and edge preservation, the size of the window should be carefully chosen.

- *Structure of Standard median filter*

For 3x3 windows, hardware implementation is depicted in Figure 2

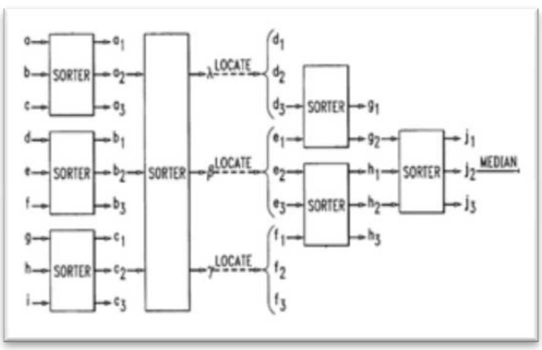

Figure-2 Architecture of median filter [8]

In Figure 2, the input data of the filter is the output of the filtering window generating module. It consists of 4 stages. After each stage, we get the min, median and max values. Out of these three values, only median value is transmitted to the next stage. This process goes on continuously and finally we get the final median value after stage 4. This median value is replaced by corrupted pixel value of the image so that we get the noise-free images as output.

III. BLOCK DIAGRAM

The block diagram shown below illustrates the complete process of presence of noise and its removal. Consider an image as input form. Suppose due to certain reasons noise is introduce in the image. This type of image is termed as Noise image. In order to get a Noise free image median filtering is used.

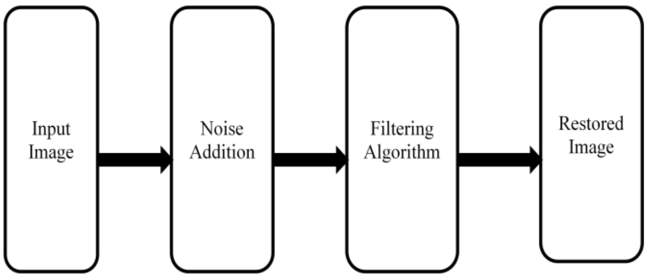

Figure-3 Block diagram for image filtering [12]

A. *Stage 1*

Input Image: We provide the input image in the form of 3x3 matrixes from the MATLAB.

B. *Stage 2*

Noise Addition: After providing the input image, we add the 'salt and pepper' noise in our image.

C. *Stage 3*

Filtering Algorithm: Salt and Pepper noise is removed from the noisy image by using the following algorithm [6].

Step 1*:* A two dimensional window size 3x3 is selected and centered on the processed pixel p(x, y) in the corrupted image.

Step 2*:* Sort the pixels in the selected window according to the ascending order and find the median pixel value denoted by (P_{med}), maximum pixel value (P_{max}) and minimum pixel value (P_{min}) of the sorted vector V0. Now the first and last elements of the vector V0 is the P_{min} and Pmax respectively and the middle element of the vector is the Pmed.

Step 3: If the processed pixel is within the range $P_{min} < P(x, y) < P_{max}$, $P_{min} > 0$ and $P_{max} < 255$,it is classified as uncorrupted pixel and it is left unchanged. Otherwise p(x, y)is classified as corrupted pixel.

Step 4: If p(x, y) is corrupted pixel, then we have the following two cases:

Case 1: If $P_{min} < P_{med} < P_{max}$ and $0 < P_{med} < 255$, replace the corrupted pixel p(x, y) with P_{med}.

Case 2: If the condition in case 1 is not satisfied then P_{med} is a noisy pixel. In this case compute the difference between each pair of adjacent pixel across the sorted vector V0 and obtain the difference vector VD .Then find the maximum difference in the VD and mark its corresponding pixel in the V0 to the processed pixel.

Step 5: Step 1 to Step 4 are repeated until the processing is completed for the entire image.

In *median filter*, window size varies according to the heavy impulsive noise density.

There are various median filter algorithms used to remove the salt and pepper noise. DBMF and DBUTM are extended form of Simple median filters [10, 11]. DBMF and DBUTM used a fixed window size as Simple median filter. The Adaptive median filter has variable window size for removal of impulses while preserving sharpness [7].

IV. METHODOLOGIES

Our proposed method is to implement the median filter in order to remove the salt and pepper noise (0 to 255). For removing this noise we use 3x3 window sizes and use the array sorting method. In this method we sort the pixels of image in ascending order to find the median value then replace this value by the centre value of the window.

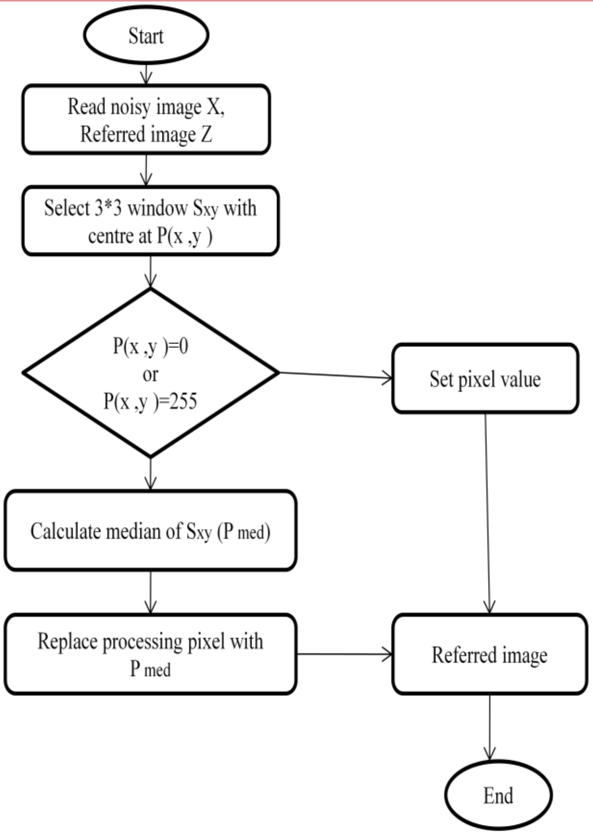

Figure-5 Flow graph of median filter

V. EXPERIMENTAL RESULTS

We have used Mat lab R2009b as the simulation tool. The 8-bit images of Lena with dimensions M X N (256 * 256) pixels are used for simulations. The pixels Q' (i, j) for $1 \leq i \leq M$ and $1 \leq j \leq N$, of the image is corrupted by adding impulse noise, with noise density ranging from 0.1 to 1. The Peak signal to noise ratio (PSNR) is used to compare the relative filtering performance of various filters. The PSNR between the filtered output image Q (i, j) and the original image Q' (i, j) of dimensions M X N pixels is defined as

$$PSNR = 10log10(255 * 2/\sqrt{mse}) \text{ .... (1)}$$

Where MSE stands for mean square error and given as

$$MSE = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}[Q'(i,j)-Q(i,j)]^2}{M \times N} \text{ .... (2)}$$

It can be seen that Peak signal to noise ratio (PSNR) is closely related to mean square error (MSE).

TABLE I. COMPARISON OF PSNR VALUES OF DIFFERENT ALGORITHMS FOR LENA IMAGE AT DIFFERENT NOISE DENSITIES

Noise Density	SMF	DBMF	DBUTM	AMF
10	33.24	41.64	43.02	41.51
20	29.15	37.50	39.31	37.60
30	23.51	34.71	36.62	35.23
40	18.95	32.43	34.51	33.53
50	15.39	30.23	32.26	32.22
60	12.39	27.93	30.08	30.75
70	10.03	25.84	27.89	29.15
80	8.13	23.01	24.68	27.44
90	6.65	19.72	20.25	25.13

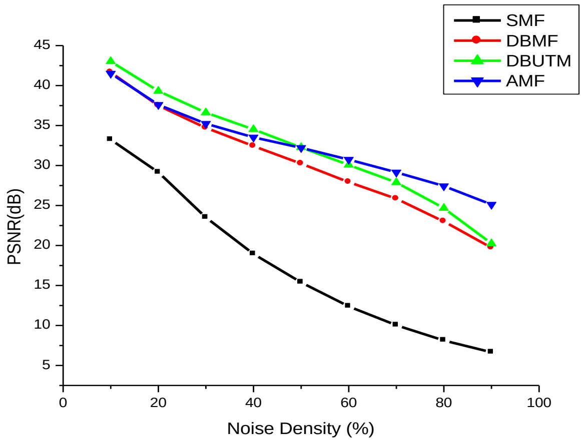

Figure-6 Noise densities versus PSNR (db) for Lena Image

The table I and figure 6 shows the PSNR value of simple, decision-based, Decision Based Unsymmetrical Trimmed and Adaptive median filter. It has been observed that DBMF and AMF give same output at low noise density. DBUTM give best result at low noise density. But at high noise density AMF gives better result.

TABLE II. COMPARISON OF RMSE VALUES OF DIFFERENT ALGORITHMS

FOR LENA IMAGE AT DIFFERENT NOISE DENSITIES

Noise Density	SMF	DBMF	DBUTM	AMF
10	5.55	2.14	1.75	1.83
20	8.89	3.40	2.73	2.91
30	17.02	4.74	3.80	3.93
40	28.77	6.16	4.78	5.09
50	43.33	7.87	6.22	6.17
60	61.21	10.03	8.00	7.26
70	80.90	13.04	10.49	8.67
80	99.83	17.68	14.72	10.55
90	118.77	26.81	24.88	12.65

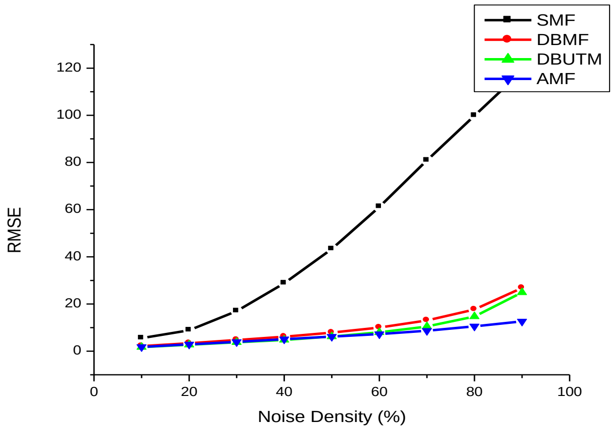

Figure-7 Noise densities versus RMSE for Lena Image

The table II and figure 7 shows the RMSE value of SMF, DBMF, DBUTM and AMF. It is observed that AMF and DBUTM have lower RMSE at higher noise density as compared to SMF and DBMF which indicate the improvement in filtering.

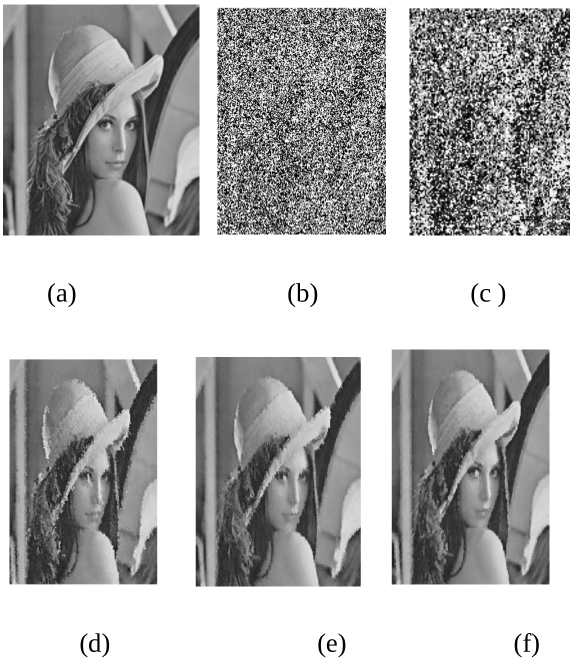

Figure-8. Results for 80% noise corrupted Lena image (a) original image (b) 80% noise corrupted image. Restoration results of (c) Simple median filter (d) Decision based median filter (e) Decision Based Unsymmetrical Trimmed Median Filter (f) Adaptive median filter

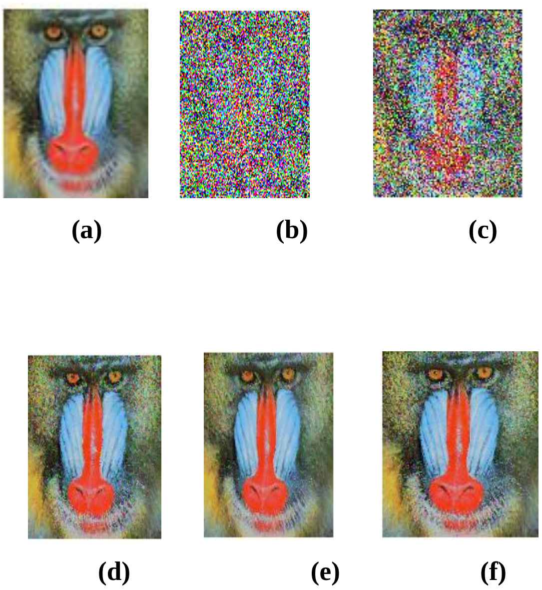

Figure-9 Results for 80% noise corrupted baboon image (a) original image (b) 80% noise corrupted image. Restoration results of (c) Simple median filter (d) Decision based median filter (e) Decision Based Unsymmetrical Trimmed Median Filter (f) Adaptive median filter.

Figure 8 show the result when these four methods are applied to 80 % corrupted Lena image by impulse noise, which is considerably high level of noise. SMF is not able to remove noises and DBMF removes some of the noises. The results show that visually DBUTM and AMF produced the sharp output.

Figure 9 show restoration results of 80% corrupted baboon (color) image. In high dense noisy image AMF gives much better filtering compared to SMF, DBMF and DBUTM.

VI. Conclusion

Median Filter is a powerful non-linear filter which is based on order statistics [12]. We have studied various techniques for removal of impulse noise i.e. salt n pepper noise. In this paper different non linear algorithms for the removal of salt and pepper noise are compared and analyzed on the bases of PSNR, RMSE value and visual inspection. Both the DBUTM and AMF work similar and give better performance as compared to traditional median filter and DBMF at different noise density. At higher density AMF works better as compared to DBUTM. In all above filtering algorithms except AMF fixed size window approach is adopted as it results in less blurring in filtered images. The results obtained are partially noise-free but further improvements can be possible for both highly and lowly corrupted images. The proposed platform is FPGA and description language is VHDL in order to implement median filter. Model-Sim will be used for simulation and Leonardo Spectrum for synthesis purpose. Matlab is used for image generation. Because of its simplicity, this method is used for noise removal purpose.

References

[1] Maria Petrou, and Panagiota Bosdogianni, “Image Processing: The Fundamental”, John Wiley & Sons Ltd, 2000.

[2] Rafael C.Gonzalez and Richard E. woods, “Digital Image Processing”, Pearson Education, Second Edition, 2005

[3] Dr.Philippe Cattin (2012-04-24). "Image Restoration: Introduction to Signal and Image Processing". MIAC, University of Basel. Retrieved 11 October 2013

[4] Raymond H. Chan, Chung-Wa Ho, and Mila Nikolova, "Salt-and pepper noise removal by median-type noise detectors and detail preserving regularization", IEEE Trans. Image Processing, vol. 14, no. 10, pp. 1479-1485, October 2005.

[5] Zhou Wang, and David Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images", IEEE Trans. Circuits and Systems –II: Analog and Digital Signal Processing, vol. 46, no. 1, pp. 78-80, January 1999.

[6] H. Hwang, and R. A. Haddad, "Adaptive median filters: New algorithms and results", IEEE Trans. Image Processing, vol. 4, no. 4, pp. 499-502, April 1995.

[7] Takeaki Matsubara, Vasily G. Moshnyaga, and Koji Hashimoto," A Low-Complexity and Low Power Median Filter Design", 2010 International Symposium on Intelligent Signal Processing and Communication Systems (lSPACS 2010) December 6-8, 2010.

[8] International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-1, Issue-5, November 2011.

[9] K.S. Srinivasan and D. Ebenezer," A New Fast and Efficient Decision-Based Algorithm for Removal of High-Density Impulse Noises", IEEE Signal Processing Letters, Vol. 14 Issue 3, pp. 189-192, 2007.

[10] Dodda Shekar and Rangu Srikanth,"Removal of High Density Salt & Pepper Noise in Noisy Images Using Decision Based Unsymmetrical Trimmed Median Filter", International Journal of Computer Trends and Technology, Vol. 2 Issue1, pp. 19-114, 2011

[11] Pandey R, "An Improved Switching Median filter for University Distributed Impulse Noise Removal", WASET, vol.28, pp. 349-351, 2008.

[12] Manju Chouhan, C.D Khare," Implementation of Median Filter for CI Based on FPGA", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 2, Issue 12, December 2013.