Hardware Implementation of Modified Weighted Median Filtering on FPGA

Akula Venkata Subba Rao
M.Tech Student DECS
Gudlavalleru Engineering College
Gudlavalleru, Krishna Dist, AP, India

Penta Anil Kmumar
Assistant Professor
Gudllavalleru Engineering College
Gudlavalleru, Krishna Dist, AP, India

ABSTRACT
In real time applications, most of the times, image is subjected to the noise due to the transmission of images through the channels, scanning, digitizing, and storing at the output. If this noisy image is used for the further processing, it may leads to the incompetent decision for the human interpretation. So the image filtering is essential to keep away from the noise present in the image. The performance of an image filtering system depends on its capability to detect the occurrence of noisy pixels in the image. The earlier image filtering methods has a drawback of pure image quality i.e. low PSNR and high MSE values. And also these filtering methods are designed using PC. Generally PC based image filtering system uses general purpose processor. This type of implementation takes more amount of time to provide the filtered result. Because this PC based filtering system executes the instructions in step by step manner. In order to eradicate the drawbacks associated with existing filtering methods, a new filtering technique is implemented by means of FPGA. Because FPGA's supports parallelism. This new proposed technique gives high image quality that means High PSNR and low MSE values. And also this FPGA based hardware implementation takes less amount of time to get the filtered result.

Keywords
Image, noise, FPGA, PC, PSNR, MSE

1. INTRODUCTION
Image filtering is used in many areas for taking the decisions. Common occurrence of noise in digital images is salt & pepper noise. The salt & pepper noise occurred due to the transmission of pictures through the channels, scanning, digitizing, and storing at the output [1]. Some amount of output is degraded. Salt & pepper noise can corrupt the images where the corrupted pixel takes the minimum or maximum gray level. The basic filter is the median filter for removing the salt & pepper noise in digital images. This filter is good at lower percentages of noise in images. After that so many filters are implemented but those are not sufficient for real time implementation.

FPGA's are used in modern digital image applications like image filtering, medical imaging & image compression [3]. FPGA's are good at flexibility, reprogram and parallelism. The earlier image filtering operations are implemented on PC. But this approach of implementation takes more time. Because PC executes the operations in step by step order. It consumes more time to get the output. Because FPGAs supports parallelism [3]. So the speed of operation is good and gives the fast output response compared to the personal computer.

Xilinx Platform Studio is one of the integrated software environment (ISE) tools. By using Xilinx platform studio (XPS) and visual basic (VB), image filtering operation is implemented on FPGA. By using corresponding hardware description language to the respected image filter technique i.e.is impulse C code. Xilinx platform studio consists of mainly three panels. Those are project information panel, system assembly panel and connectivity panel. The project information panel provides offers control over and information about the project. And also provides the Project, Applications & IP Catalog tabs. The system assembly panel is where you view and configure system block elements. The connectivity panel is a graphical representation of the hardware platform interconnects. Bit stream is generated by using these 3 panels. Download the Executable & Linkable Format file on FPGA. By using Rs232 serial cable and parallel JTAG is connected to the target board by using PC.

2. LITERATURE SURVEY
2.1 Why FPGA is Particularly Selected for the Implementation of Image Filtering
The literature survey has confirmed that a number of hardware platforms exist to implement the real time signal processing solutions. Each platform has its own strengths and weaknesses. The hardware platforms available for implementing the real time signal processing applications; are Application-Specific Integrated Circuit (ASIC), DSP (Digital Signal Processor), FPGA (Field programmable Gate Array), MCU (Micro Controller Unit), RISC (Reduced Instruction Set Computer).

Among the above mentioned hardware platforms, FPGAs and DSPs offer unique and different options for signal processing applications. The FPGAs will continue to be used for many of today' challenging signal processing application. The reason for this are assured below.

Digital signal processors are a specialized form of microprocessor, while FPGAs are a form of highly configurable hardware. FPGAs have reconfigurable gate structure which consumes more power. In case of DSPs, the hardware has been already configured, this requires less power consumption [4].

Currently, the primary reason most engineers choose use a FPGA over digital signal processors is driven by the MIPS requirements of an application [5]. Performance is measured in terms of number of digitized samples that can be processed in a given amount of time and also in MIPS (Millions of instructions executed per second) & MMACS (Millions of Multiply-Accumulate Operations per Second). Generally an

FPGA can process samples at rates that are an order of magnitude higher then DPSs. Due to this FPGA provide high performance compared to DSPs. Both are comparable in product reliability and maintainability.

The FPGA design implementation cycle requires less hardware-specific awareness than most DSP chips or ASIC design solutions. Smaller design groups, with less experienced engineers, can design larger, more composite DSP systems in less time than larger design groups with more experienced engineers who are required to know device-specific programming languages. The FPGA-based DSP system-level design team can design, test, verify, and ready a complex DSP system for production in a few weeks.

After a careful evaluation of each and every performance factor, it is conclude that FPGA is the best preference for implementation of image processing applications.

2.2 Various Existing Image Filtering Techniques

Noise removal is one of the most imperative areas with in digital image processing. One type of noise that can be come out in images is salt & pepper noise, which can be produced during image acquisition, storage, or transmission and can affect later stages of processing if not removed properly while preserving the other original details of the image. So image must go through the denoising phenomenon. Here the denoising phenomenon is used to remove the noise while retaining the edges and other features of the original image as much as possible.

The noise-free images are very much essential to analyze and take the decision about the targets in so many areas like radar, military, navy, air force and medical applications. So, denoising is a very important pre-processing task before further processing of image like segmentation, feature extraction, texture analysis etc. Suppose if the image is directly subjected to the further processing without denoising process, it may leads to incorrect decision making in the identification of the target.

Several algorithms are available for denoising the image. But each algorithm has its own strengths & weaknesses. Number of quality or performance evaluation factors has been identified to check the effectiveness of an image filtering algorithm. Some of these measures include PSNR, MSE & SSIM. So it is concluded that in the area of image filtering, selection of the good image filtering algorithm is also a very important obsession. This selection is made in such a way that the algorithm must have high PSNR value and low MSE value. Filtering action, merits and demerits of the some image filtering algorithms are furnished below.

2.2.1 Mean Filter

This filter eradicates the unnecessary noise in images by a local averaging operation where the value of each pixel is replaced by the average of all the values in the local neighbourhood. This filter is good at lower masking for maintenance the image details, without any defeat. But this filter has an imperative disadvantage. All image structures, including points, edges and lines are blurred. So the quality of the image is not good.

2.2.2 Median Filter

This filter has an advantage over the mean filter. This filter eradicates the salt & pepper noise at edges, points and lines, but not clearly [7]. This filter performs the operation according to the selected masking window. After selecting the masking, it sorts the pixels in ascending or descending order. After arranging in one of the order, it selects the middle value as median. If number of pixels is odd, then it takes the middle value as median. If number of pixels is even it takes the average of two middle pixel values as median. The median value is place at the centre of the selected masking window. This masking operation is done in all the pixels located in the image by moving the pixel pointer (i, j). It is comparatively expensive and complex to compute are the constraint to this filter.

2.2.3 Weighted Median Filter

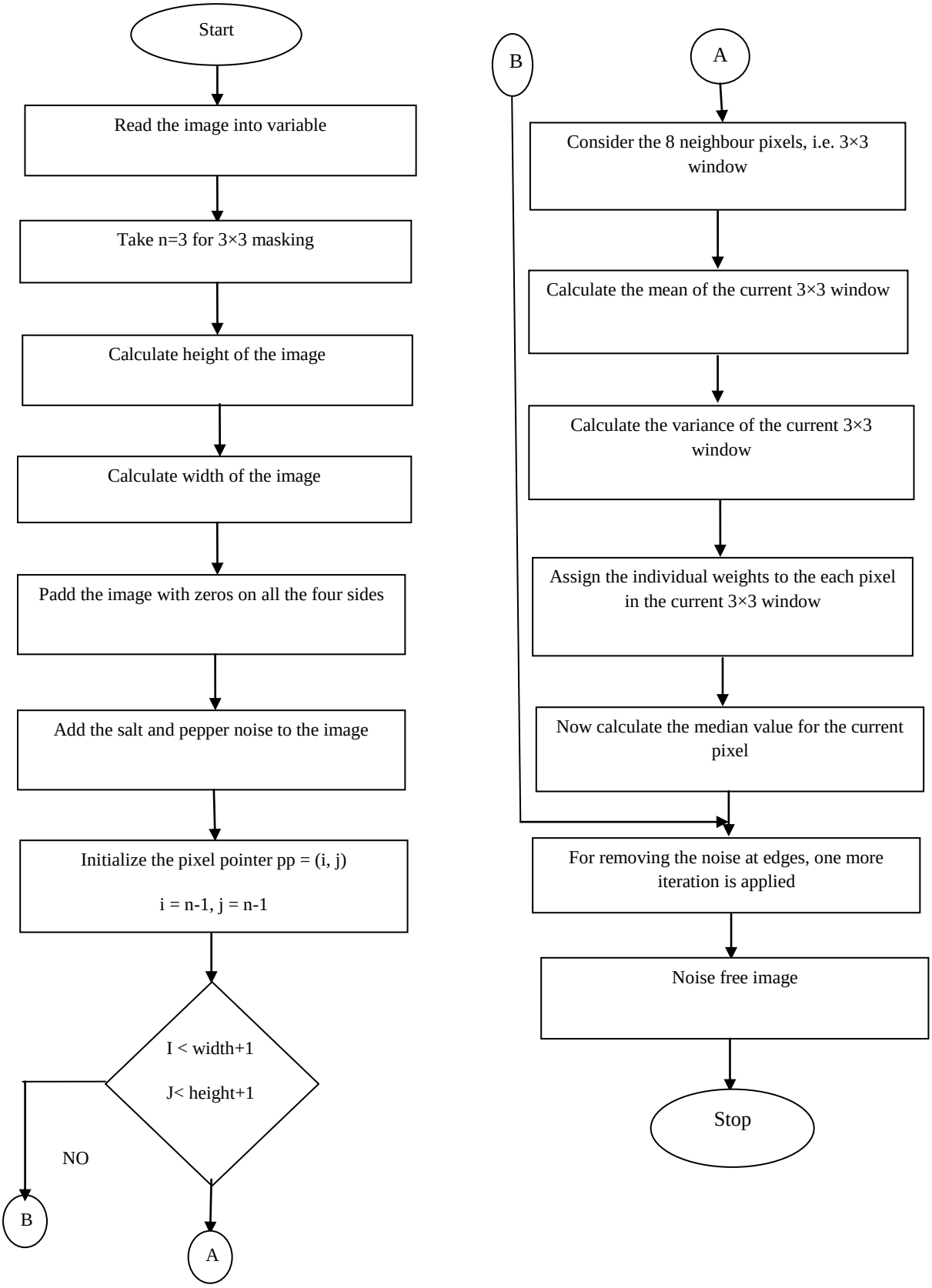
This filter overcomes the drawback of previous filter techniques. In standard median filter, each pixel in the filter region has the same influence in spite of its distance from the centre. In this filtering, weights are assigned to the each individual pixel in the image, which can be interpreted as the number of votes for the corresponding pixel values. After assigning the suitable weights to the each pixel in the image, then median filtering operation is continued. It eradicates the noise effectively compared to the median filter.

3. MATLAB IMPLEMENTATION OF PROPOSED TECHNIQUE

Extraction of noise from the images is the most challenging problem in the area of image processing. However the performance of the image filtering algorithm depends heavily on its performance evaluation factors. The earlier image filtering algorithms has a drawback of poor PSNR and MSE values which results in the inefficient extraction of noise from the images. So in order to eradicate the problems associated with earlier techniques, a new technique named *Modified Weighted Median* filtering is introduced in this paper. This proposed method follows the *effective weights assignment* procedure to reduce the high density of the salt & pepper noise in the images. Earlier algorithms do not effectively remove the noise content which is present in the edges of the image. But in the proposed algorithm, zero padding is applied on all the sides of the image to remove the noise content present in the image. The proposed method has been designed and implemented in MATLAB using image processing toolbox. Different kinds of the images are taken to validate the performance of the proposed algorithms. Comparative analysis has shown significant improvement over the available methods. The various implantation steps of the proposed algorithm are provided below.

1. Read the input gray scale image by using proper Matlab command.
2. Compute the width & height of the input image, and then pad the image with zeros on all sides of the image without any inconvenience to the edges.
3. Initialize the pixel pointer with appropriate value (which initiates the masking process by taking the first window).
4. Compute the mean and variance values for the current window. This current window is pointed out by pixel pointer.
5. After calculating the mean and variance of current block, each individual pixel in is distinction with mean and multiplies with the reciprocal of variance.
6. After assigning the weights to the pixels and then sorts the pixels in ascending order, it replaces the centre value as median value in the masking.
7. One more iteration is applied by using basic median filter for eradicate the left over any unnecessary noise at edges.

Flow Chart


Start
Read the image into variable
Take n=3 for 3×3 masking
Calculate height of the image
Calculate width of the image
Padd the image with zeros on all the four sides
Add the salt and pepper noise to the image
Initialize the pixel pointer pp = (i, j)
i = n-1, j = n-1
I < width+1
J< height+1
NO
B
A
B
A
Consider the 8 neighbour pixels, i.e. 3×3 window
Calculate the mean of the current 3×3 window
Calculate the variance of the current 3×3 window
Assign the individual weights to the each pixel in the current 3×3 window
Now calculate the median value for the current pixel
For removing the noise at edges, one more iteration is applied
Noise free image
Stop

4. HARDWARE IMPLEMENTATION

The main aim of hardware implementation is to condense the processing time. The personal computer executes the instructions in consecutive order. So the processing time is increased. For overcome this difficulty, the proposed filtering techniques is implemented on FPGA.

The entire hardware implementation development is separated into 2 modules those are:

Module1: The digital gray scale image is transformed into header file format.

Module2: System C code for Proposed filtering Technique & procedure to implement on FPGA by using Xilinx Plat Form Studio.

Module1: Conversion of NOISY input image into corresponding HEADER file.

Step1: Development of Mat lab code with GUI for header files format conversion

Open the MATLAB and create a new M file. In the new M file, write the corresponding MATLAB code to convert the given noisy image into the header file. Here the matlab code is written in such way that it supports graphical user interface. With this GUI, the user can easily understand the step by step procedure of the file conversion from *noisy image file* to header file (having only one dimensional pixel values). Here the GUI is created with two radio buttons. One button is used to select the desired input noisy image which is to be converted into header file, and another button is used for the generation of header file.

Step2**:** The following GUI based window appears on the screen when RUN option is selected for the M- file created in the step1.

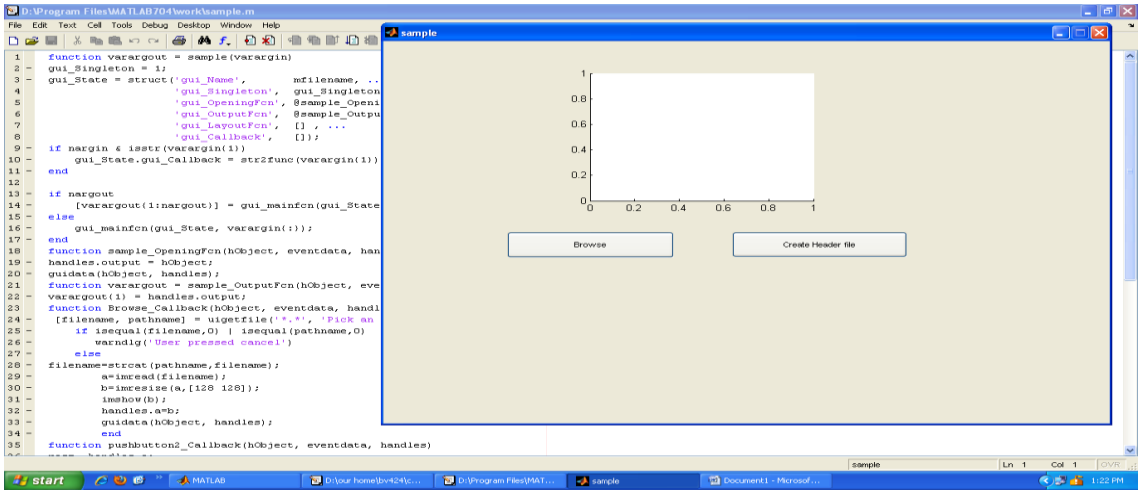


Fig 1: This snapshot shows the GUI based window is obtained when RUN option is selected

Step3: Click on Browse button to choose the appropriate noise image as the input.

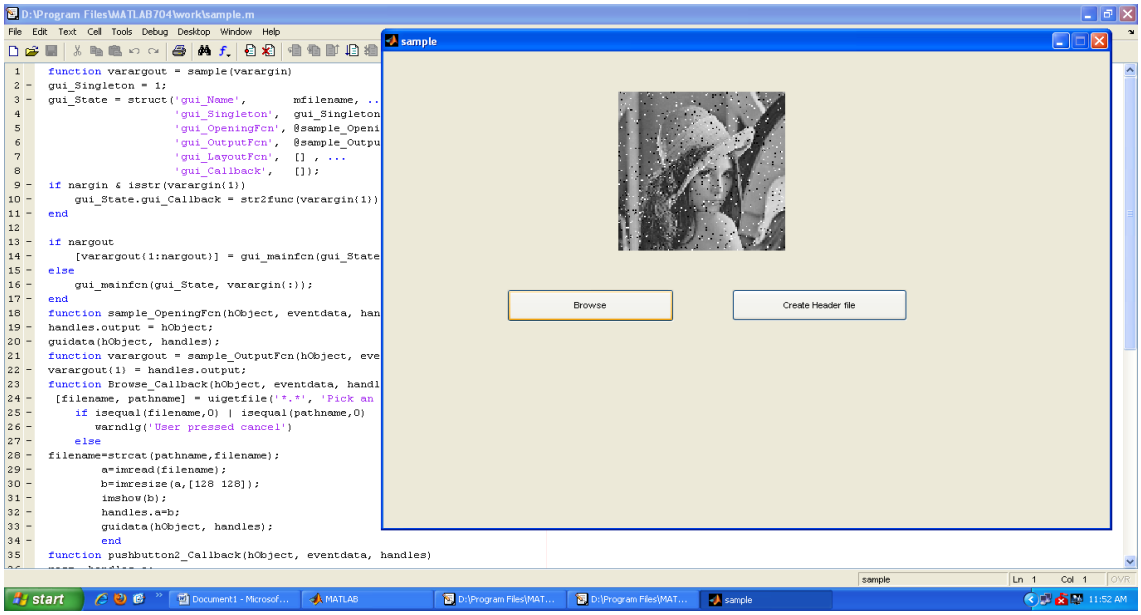


Fig 2: This snapshot shows the appropriate Noise input image is readed.

Step4**:** To generate the header file for the given input noisy image, Click on *create header file* button. Now a new dialog box is displayed which indicates that the file has been created successfully.

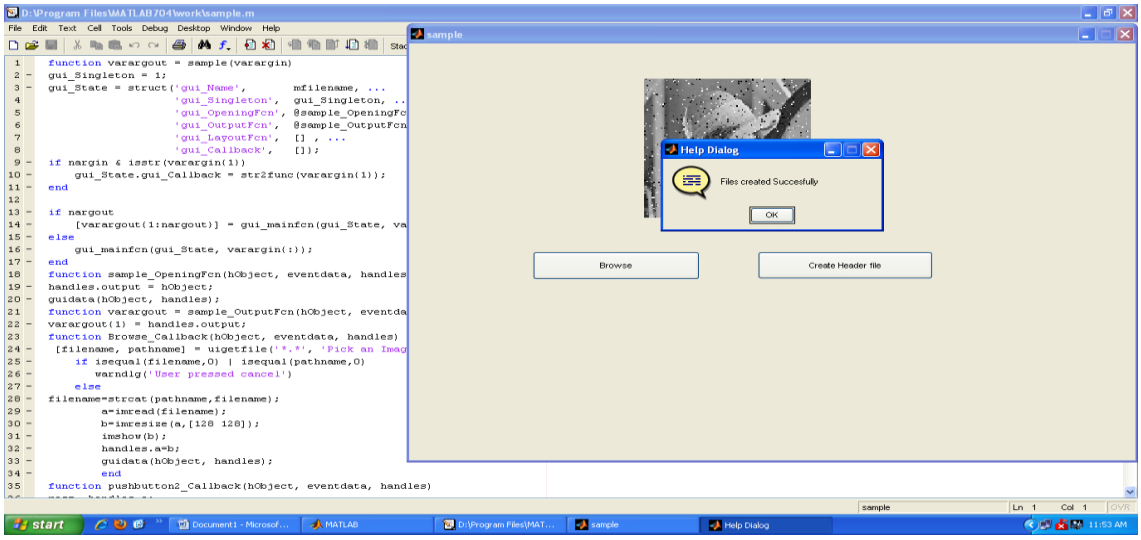


Fig 3: This snapshot shows the dialog box. The dialog box indicates that the header file is created successfully

Step5: If click on *OK* button, then the header file is created successfully and it is displayed in one separate window.

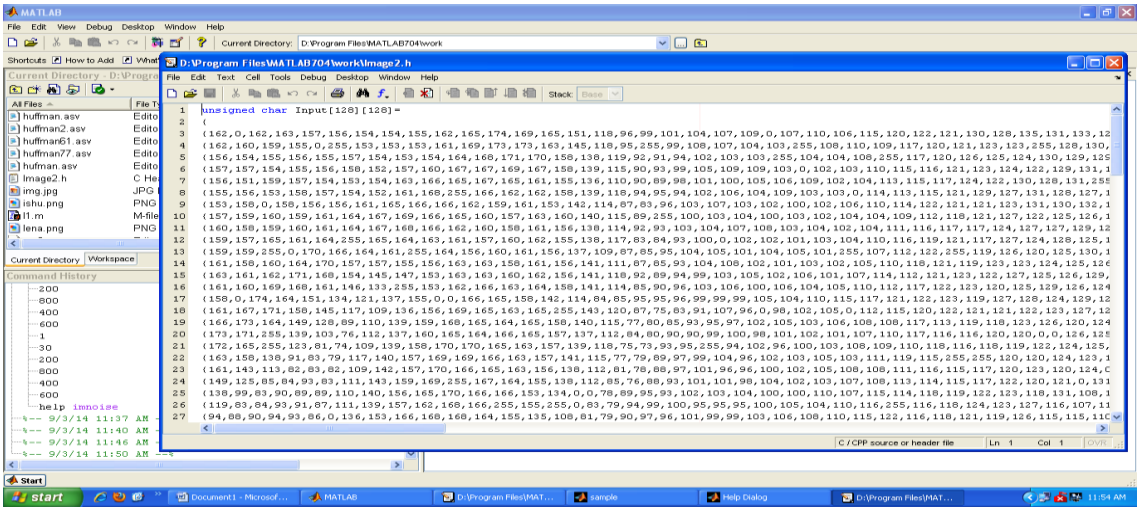


Fig 4: This snapshot shows the header file output

Module2: System c code & procedure for hardware implementation.

- The system C code is developed for the proposed filtering technique.
- Create the new project in Xilinx Plat Form Studio.
- The application tab lists all software application option settings, header files, and source files associated with each application project.
- So add the header file code & system C code on Application Tab in XPS and load it to the block RAM.
- Software > Build All User Applications, Compiles the source files.
- Hardware > Generate Netlist, Bit stream, synthesis report is obtained.
- Device Configuration > Download Bit stream, XPS downloads the bit stream (download.bit file) onto the target board using iMPACT in batch mode. XPS uses the file etc/download.cmd for downloading the bit stream. Because XPS tools are make file based, the download button calls on the make file and executes the steps necessary to create the bit stream with the Executable and Linkable Format (ELF) file populated within the bit stream.

5. RESULT ANALYSIS

To check the functionality of various image filtering algorithms such as standard median filter algorithm, adaptive median filter algorithm, weighted median filter algorithm, modified weighted median filter algorithm, they are first verified in the Matlab environment. After the accomplishment of verification phase or simulation phase, equivalence c code is developed on the Xilinx plat form studio. The matlab results corresponding to the various algorithms and proposed work are furnished below.

5.1 Matlab Simulation Results

The figure5 indicates original input image, original input image with noise, noise-free image. Here, first image is readed and then salt & pepper noise is added to the original input image. This noisy image is next subjected to the proposed algorithm. The proposed algorithm extracts the noise and gives noise-free image.

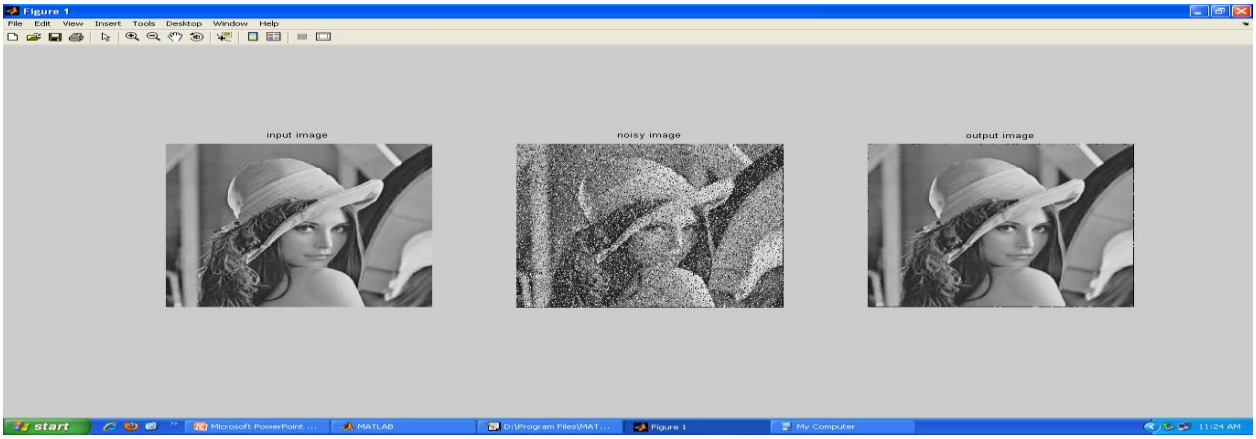


Fig 5: This snapshot shows the simulated output at 20 percentage of noise

The MSE & PSNR values corresponding to output noise-free image is (when 20 % of noise is added to the image)at 20 percentage of noise) shown in figure6.

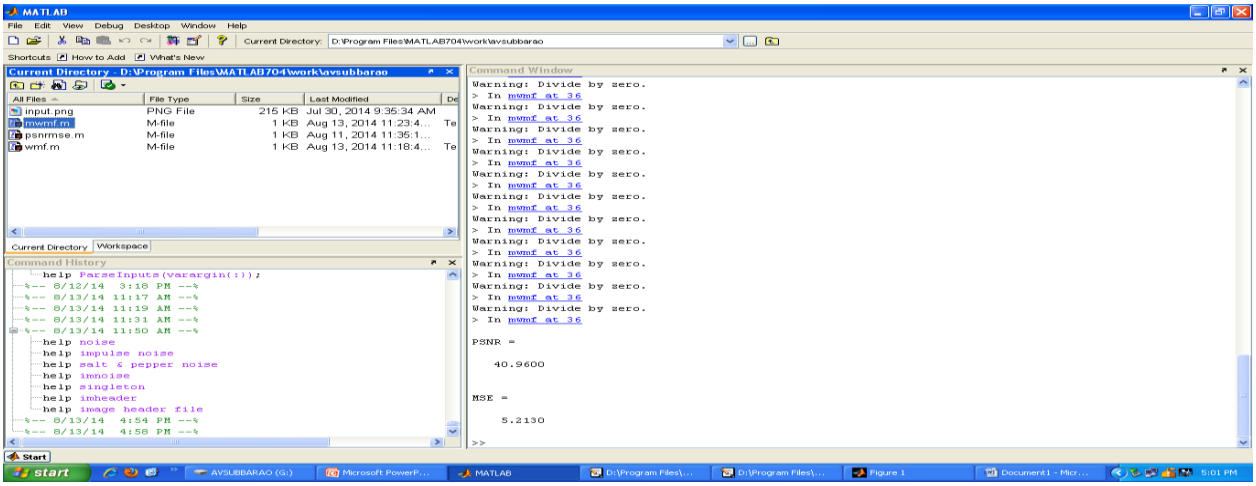


Fig 6: This snapshot shows the PSNR & MSE values at 20 percentage of noise

The performance of the proposed algorithm is evaluated based on visual quality, PSNR & MSE values. The definitions of various performance evaluation factors are stated below.

(i) Mean Squared Error [MSE]: Better reconstruction gives lower values and zero indicates perfect reconstruction.

(ii) Peak Signal to Noise Ratio [PSNR]: This is a measure related to the previous one. It is a logarithmic measure widely used in image processing. Higher values indicate better quality.

$$\text{MSE} = \frac{\Sigma_{i=1}^{x}\Sigma_{j=1}^{y}(g(i,j)-f(i,j))\text{^}2}{xy}$$

PSNR = 10log10 (A^2/MSE

Table 1. MSE values of existing techniques and proposed technique for the Lena image.

Percentage	5	10	20	30	40
SMF	11.17	17.58	62.13	260.10	795.55
AMF	3.82	6.42	11.85	24.22	61.21
WMF	2.88	4.69	8.80	14.15	23.68
MWMF	2.14	2.90	5.22	8.36	13.04

Table 2. PSNR values of existing techniques and proposed technique for the Lena image.

Percentage	5	10	20	30	40
SMF	37.64	35.67	30.19	23.96	19.12
AMF	40.75	38.78	36.77	35.47	30.26
WMF	43.53	41.41	38.68	36.62	34.38
MWMF	44.81	43.50	40.96	38.90	36.97

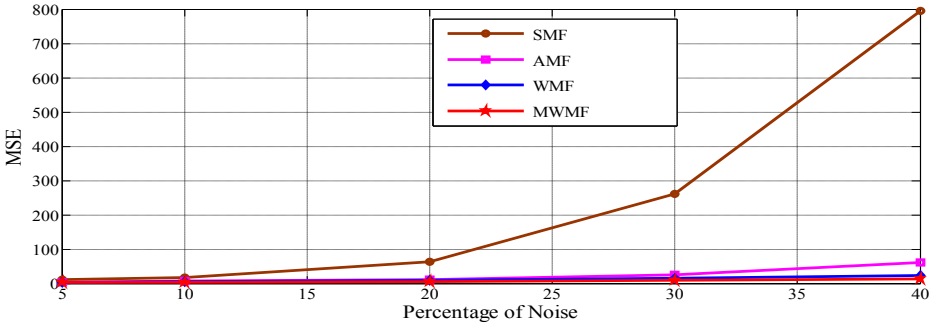


Fig 7: Variation of MSE values of table1 with respect to the percentage of noise

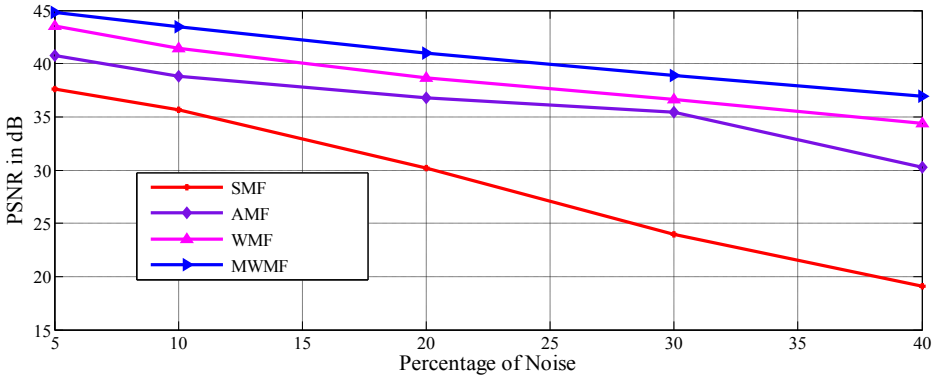


Fig 8: Variations of PSNR values of table2 with respect to the percentage of noise

Comparing with existing filtering algorithms, the proposed algorithm gives better PSNR & MSE values

5.2 Hardware Results

The proposed algorithm is implemented on Spartan3 XC3S200 FPGA by using Xilinx Plat Form Studio. The output results are shown on desktop with the help of JTAG, RS-232 & Visual Basic. The below snapshots shows the output is obtained by using FPGA device.Noise input image is readed into FPGA.

The noisy image output is appeared on desktop.

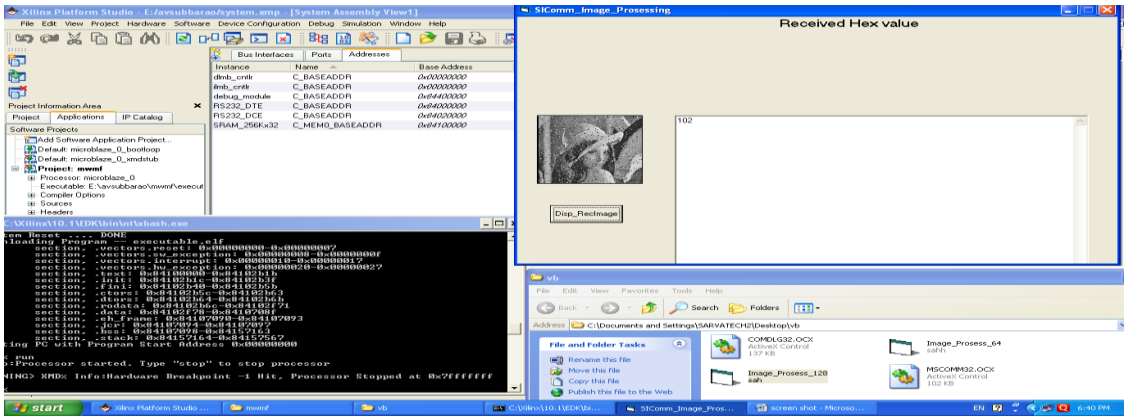


Fig 9: This snapshot shows the Noise input image

Denoised image is appeared after processing though the filtering technique by using FPGA.

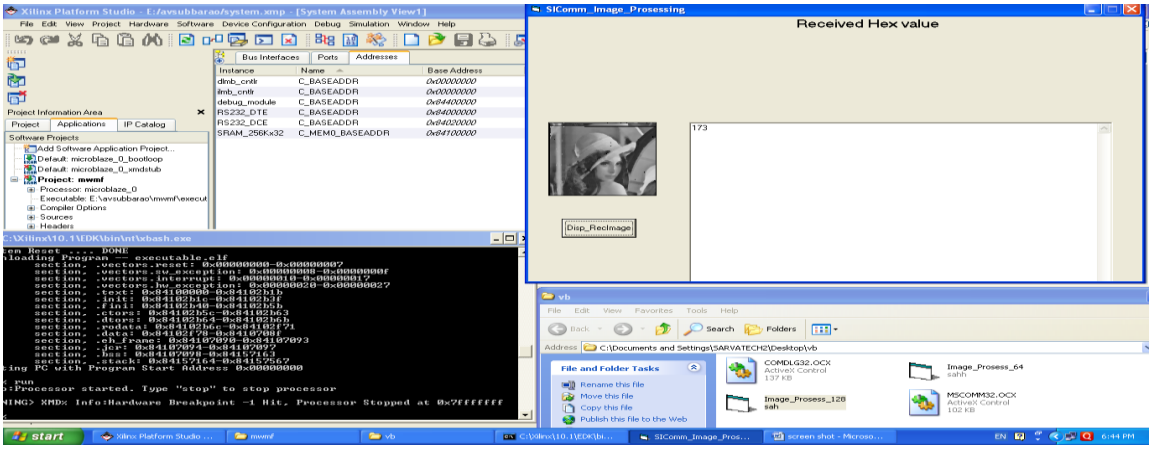


Fig 10: This snap shot shows the Noise Free image

6. CONCLUSION & FUTURE SCOPE

It is concludes that the proposed filtering technique gives the improved results in terms of Visual Quality, PSNR (Peak Signal Noise Ratio] & MSE (Mean Square Error) as compared to the previous techniques. It also concluded that it effectively removes the salt & pepper noise present in images. Both visual quality and quantitative results are demonstrated. And these results are verified on FPGA Spartan3 device for reducing the processing time. In FPGA the filtered output is acquired within nanoseconds compared to the Matlab implementation. FPGA implementation takes less time compared to the matlab implementation

During the process of transmission of Videos over channel, Video frames are corrupted by salt and pepper noise due to the faulty communication systems. In this paper, the proposed algorithm is applied only to the independent images. It is also possible to implement the same algorithm to video frames. By applying the proposed algorithm- *Modified Weighted Median Filter* to noisy video frames, can make the noisy video frames to noise free videos.

7. REFERENCES

[1] Jasdeep kaur, Mamta Garg, "an improved weighted median filter for the image processing application", International Journal of science and Research (IJSR), India Online ISSN: 2319-7064. Volume 2 Issue 4, April 2013.

[2] B Murali Krishna, K Gana Deepika, "Image Processing using IP Core Generator through FPGA", International Journal of Computer Applications (0975 – 8887) Volume 46– No.23, May 2012.

[3] N. Vani, N. Usha rani "FPGA implementation of image filtering applications using XGS", International Conference on Navigational Systems & Signal Processing Applications, December 13th &14th, 2013.

[4] Sparsh Mittal, Saket Gupta and S. Dasgupta, "FPGA: An Efficient and Promising Platform For Real-Time Image Processing Applications", Proceedings of the National Conference on Research and Development in Hardware & Systems (CSI-RDHS 2008) June 20-21, 2008, Kolkata, India.

[5] S. Klupsch, et al. "Real Time Image Processing based on Reconfigurable Hardware Acceleration", Available www.mpiinf.mpg.de/~strzodka/papers/public/KlErHu_02fpga.pdf.

[6] S. Zhang and M. A. Karim, "A new impulse detector for switching median filters", IEEE Signal Process. Lett., vol. 9, no. 11, pp. 360–363, Nov. 2002.

[7] K Nareshkumar, CH Rajarao, "Performance Analysis of Weighted Median Filter Technique for High Density Salt and Pepper Noise Removal", IJRRECS/October 2013/Volume-1/Issue-6/1170-1175.

[8] H. Hwang and R. A. Hadded, "Adaptive median filter: New algorithms and results," IEEE Trans. Image Process., vol. 4, no. 4, pp. 499–502, Apr. 1995.

[9] Jain, P. and et al. 2012," Text Fusion in Medical Images Using Fuzzy Logic Based Matrix Scanning Algorithm", International Journal of Scientific and Research Publications, Volume 2, Issue 6, ISSN 2250-3153.

[10] Mahdi, J, and mehdi "Impulse noise Detection and Reduction using Fuzzy logic and Median Heuristic Filter'', International Conference on Networking and Information technology, 2011.

[11] Alon Hayim, Michael Knieser, Maher Rizkalla, "DSPs/FPGAs Comparative Study for Power Consumption, Noise Cancellation, and Real Time High Speed Applications" J. Software Engineering & Applications, 2010, 3: 391-403.

[12]Castleman, K.R. "Digital Image Processing", Upper Saddle River, New Jersey: Prentice-Hall, 1996.