IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 2, Issue 6 (Jul. – Aug. 2013), PP 47-51
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Hardware implementation of Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF)

Lakshmi Darsi[1], G. Vara Lakshmi[2]
[1](PGScholar, ECE, SRK Institute of Technology Enikepadu, Vijayawada, INDIA)
[2](Assistant Professor,ECE,SRK Institute Of Technology Enikepadu, Vijayawada, INDIA)

***Abstract :* For** *the restoration of images (gray scale or color images) contaminated by the high density salt and pepper noise, an efficient filter(MDBUTMF) scheme is used. In this noise pixel is replaced with mean/median of the element of window. Compared to existing system the MDBUTMF gives results better than the existing one. For different gray scale and color images processed using MDBUTMF. It gives reduced MSE and better PSNR than existing system*

Keywords: *Impulse noise, Median filter, Salt and Pepper noise, Unsymmetrical trimmed median filter*

I. Introduction

Images are corrupted by impulses stemming from decoding errors or noisy channels or bit errors in transmission or introduced during the signal acquisition stage [1][3]. Impulse noise can corrupt the images where the corrupted pixel takes either maximum or minimum gray level. Impulse noise also called salt and pepper noise because of its appearance as white and black dots super imposed on an image. Salt and pepper noise appear as a black and white dots in the image, so white dots represents the salt noise and black dots represents the pepper noise. Detection of impulse noise is easily by the eye. Median filters are particularly effective in the presence of impulse noise [8]. To perform median filter at a point in an image we first sort the values of the pixels in question and its neighborhood determine their median and assign this values to that pixel.

For example 3×3 neighborhood the median is the 5th largest value, in 5×5 neighborhood. The median is the 13th largest value and so on [9]. The performance of median filter is quite well but it falters when the occurrence of impulse noise becomes high to overcome this situation a new algorithm is used adaptive median filter with variable window size.

Adaptive median filter are two types 1.Ranked Order Based Adaptive median filter (RAMF), 2. Impulse Size based Adaptive median Filter (SAMF). RAMF preserving sharpness better than non linear mean filter, SAMF preserving details better than Lin's scheme [2]. At higher noise densities windows size increased which leads to blurring the images. The effective removal of impulse is often accomplished at the expense of blurred and distorted features. The use of switching scheme in impulse removal has attracted more attention because it can avoid the damage of good pixels by employing an impulse detector to determine which pixel should be filtered [3][7].

Median Based detectors fails to distinguish thin lines from impulses, thin lines are interpreted as impulses and removed; we use a simple impulse detector for the SM filter that exhibits better impulse detection than other detectors. it can successfully preserve thin line and other detail features the main drawback of SMF is that defining a robust decision measures is difficult[4] because the decision is based on a predefined threshold value an additional drawback is that the noisy pixels are replaced by some median values in their vicinity without taking into local features such as possible presence of edges so details and edges are not removed satisfactory especially when the noise level is high the processing time high because window size increased, Complex circuitry is needed for implementation and determining the smoothing factor. To overcome this drawback a new algorithm is used called DBA (decision Based Algorithm).DBA uses simple fixed length of window of size 3×3 requires lower processing time compared with AMF [5]. Replacement of neibouring pixels repeatedly produced a long thin mark on image this is considered as streaking effect, to overcome this streaking effect we use decision based unsymmetrical trimmed median filter (DBUTMF) [6].

To reject the noisy pixel form selected 3×3 window we sue two types of trimmed filters:

1. Symmertic filter (alpha TMF): 2. Unsymmetric filter (U TMF):

In ATMF uncorrupted pixels are trimmed ,because of this image details will be lost which leads blurring of image to overcome this drawback in ATMF ,UTMF is proposed .3×3 window arrange in increasing or decreasing order. Zero's and 255's pixel values in image are removed from the image ,than median value of the remaining pixels is taken this median value is used to replace the noisy pixel ,this filter is called trimmed median filter because zero's and 255's are removed from the window. This procedure removes noise in better way than the ATMF, it doesn't give better results. To remove the drawbacks in DBUTMF, we use a new algorithm MDBUTMF, which gives better signal to noise ratio and reduced mean square error [7].

MDBUTMF

The proposed modified based unsymmetrical trimmed median filter algorithm processes the corrupted images by first detecting the salt and pepper noise. Processing Pixel is checked whether it is noisy or noisy free pixel, if the processing pixel is lies between maximum and minimum gray level values then it is considered as noise free pixel, it is left unchanged. If the processing pixel takes the maximum or minimum gray level takes minimum or maximum gray level then it is considered as noisy pixel which is processed by MDBUTMF.

ALGORITM

Step 1: Select 2-D window of size 3 × 3. Assume that the pixel being process is C_{ij} .

Step 2: If $0 < C_{ij} < 255$ then the processing pixel Cij is an uncorrupted pixel and its value is left unchanged.

Step 3: If Cij = 0 or Cij = 255 then the processing pixel Cij is corrupted pixel then two cases are possible as given in case i) and ii).

Case i): If the selected window contains all the elements as 0's and 255's. Then replace the processing pixel Cij with the mean of the element of selected window.

Case ii): If the selected window contains not all elements as 0's and 255's. Then eliminate 255's and 0's and find the median value of the remaining elements. Replace the processing pixel Cij with the median value.

Step 4: Repeat steps 1 to 3 until all the pixels in the entire image are processed

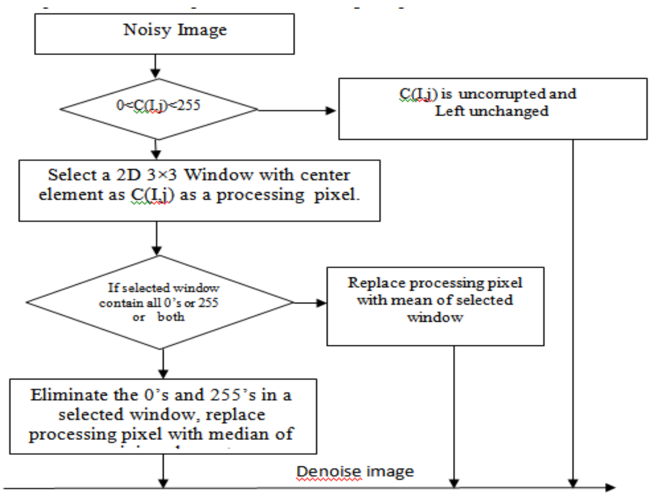
Noisy Image
0<C(I,j)<255
C(I,j) is uncorrupted and Left unchanged
Select a 2D 3×3 Window with center element as C(I,j) as a processing pixel.
If selected window contain all 0's or 255 or both
Replace processing pixel with mean of selected window
Eliminate the 0's and 255's in a selected window, replace processing pixel with median of
Denoise image

Figure 1: Flowchart of MDBUTMF

I. Illustration Of MDBUTMF Algorithm

Each and every pixel of the image is checked for the presence of salt and pepper noise. Different cases are illustrated in this Section. If the processing pixel is noisy and all other pixel values are either 0's or 255's is illustrated in Case i). If the processing pixel is noisy pixel that is 0 or 255 is illustrated in Case ii) If the processing pixel is not noisy pixel and its value lies between 0 and 255 is illustrated in Case iii

Case i): If the selected window contains salt/pepper noise as processing pixel (i.e., 255/0 pixel value) and neighboring pixel values contains all pixels that adds salt and pepper noise to the image:

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

Where "255" is processing pixel, i.e., (C_{ij})

Since all the elements surrounding (C_{ij}) are 0's and 255's. If one takes the median value it will be either 0 or 255 which is again noisy. To solve this problem, the mean of the selected window is found and the processing pixel is replaced by the mean value. Here the mean value is 170. Replace the processing pixel by 170.

Case ii): If the selected window contains salt or pepper noise as processing pixel (i.e., 255/0 pixel value) and neighboring pixel values contains some pixels that adds salt (i.e., 255 pixel value) and pepper noise to the image

$$\begin{bmatrix} 69 & 78 & 0 \\ 200 & < 255 > & 60 \\ 85 & 90 & 75 \end{bmatrix}$$

Where “255” is processing pixel, i.e., (C_{ij}).

Now eliminate the salt and pepper noise from the selected window. That is, elimination of 0’s and 255’s. The 1-D array of the above matrix is [69 78 0 200 255 60 85 90 75]. After elimination of 0’s and 255’s the pixel values in the selected window will be [60 69 75 78 85 90 200]. Here the median value is 78. Hence replace the processing pixel (C_{ij}) by 78.

Case iii): If the selected window contains a noise free pixel as a processing pixel, it does not require further processing. For example, if the processing pixel is 85 then it is noise free pixel

$$\begin{bmatrix} 70 & 66 & 69 \\ 90 & < 85 > & 55 \\ 45 & 85 & 65 \end{bmatrix}$$

Where “85” is processing pixel, i.e., (C_{ij}).Since “85” is a noise free pixel it does not require further processing.

II. Implementation

The total project plan is divided into four modules.

Module 1: Conversion of the image into a text file format.
Module 2: Custom logic for Algorithm.
Module 3: Tool’s Used.
Module 4: Implementation in FPGA.

Module 1: The Conversion of the image into a text file format:

Steps in Module 1 to create text/Header File:

1. Open MATLAB -> New -> GUI 2.Select Blank GUI and Click OK
2. Menu bar appears in the figure it consists of all the required buttons.

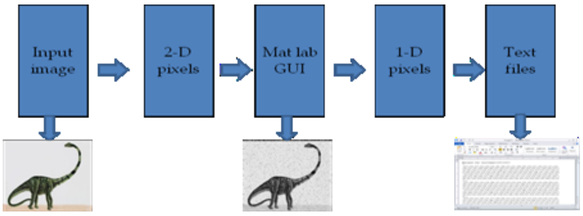

Figure: 2 conversion of image into text file.

In this module three buttons are used

1. PUSH button is used to browse the required image selection. Right click on Push button & go for property inspector and change the string name to “Browse” Right click on M-File editor and code should be written to browse the image, if the selected image is original image code should be written to add the noise and noisy image is selected on the specific path.
2. Axes button is used to show the selected image on the GUI.

PUSH button is used to create the header file. Right click on Push button & go for property inspector and change the string name to “Browse”. After that right click and select M-File and write the code to create the header file. After creation of the Header file “Header File created successfully” pop-up shown.

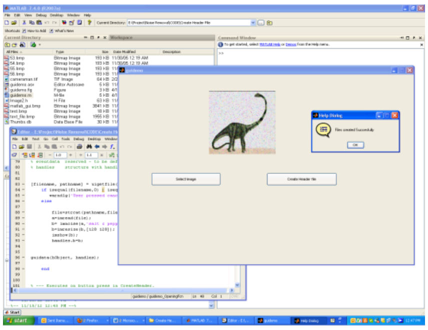

Figure 3: Conversion of image to header file

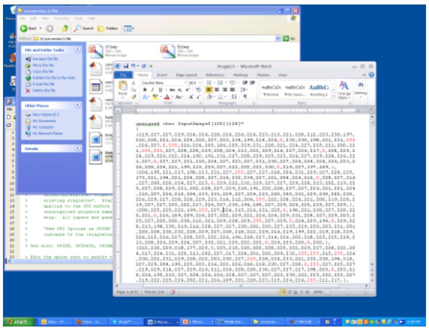

Figure 4: Module one output.

Module 2: Custom logic

Logic to remove the salt and pepper noise should be written in "System C" language.

Main Features are as follows:

- Writing software to control the MicroBlaze processor must be done in C/C++ or System C language.
- Using System C is the preferred method by most people and is the format that the Xilinx Embedded Development Kit (EDK) software tools expect.
- The EDK tools have built in System C compilers to generate the necessary machine code for the MicroBlaze processor. The FPGA / FPGA chip is supported with acomplete set of software and hardware development tools, including Xilinx Embedded Development Kit (EDK) and Xilinx Platform Studio (XPS) tools development software.
- This tool is used to create a simple processor system and the process of adding a custom OPB peripheral (an 32-bit adder circuit) to that processor system by using the Import Peripheral Wizard.

Module 3: Tool's Used

Device

FPGA Spartan3 XC3S200

Software

Xilinx ISE 8.1i
Xilinx Platform Studio
Visual Basic
Matlab

Language used

System C

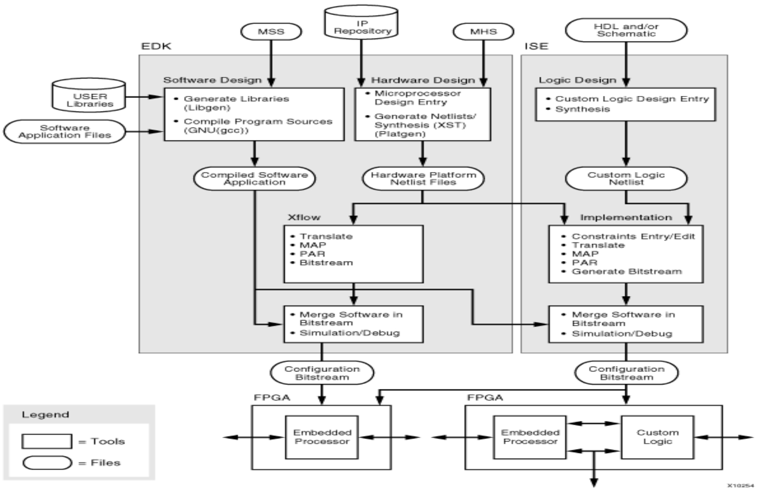

Figure 5: Implementation in SPARTAN 3 FPGA Architecture

Configuration

- MHS- microprocessor hardware specifications.
- MSS- microprocessor Software specifications.
- UCF – user constraints file.
- Linker script.

III. Results

Output of Restored Image is shown as below

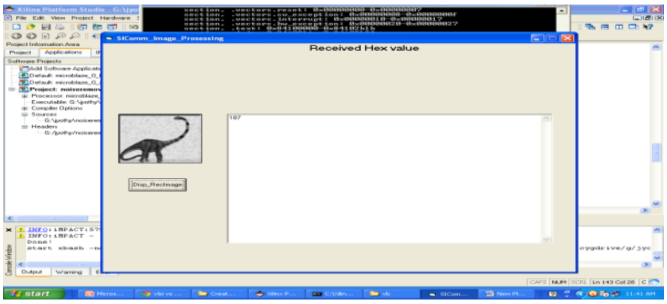
```
section, .vectors.reset: 0x00000000-0x00000007
section, .vectors.sw_exception: 0x00000008-0x0000000f
section, .vectors.interrupt: 0x00000010-0x00000017
section, .vectors.hw_exception: 0x00000020-0x00000027
section, .text: 0x84100000-0x84102b1b
```
```
INFO:iMPACT:5?
INFO:iMPACT -
Done!
start xbash -n
```

Figure 6: Output of Restored Image

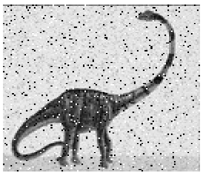

Figure 7:Noisy Image

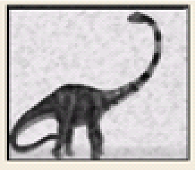

Figure 8:Restored Image

IV. Conclusion

In this project, we have proposed a new algorithm (MDBUTMF) which gives better performance in comparison with MF, AMF and other existing noise removal algorithms. FPGA implementation is verified for the MDBUTMF. The performance of the algorithm has been tested on both gray-scale and color images. Because the new impulse detection mechanism can accurately tell where noise is, only the noise-corrupted pixels are replaced with the estimated central noise-free ordered mean value. As a result, the restored images can preserve perceptual details and edges in the image while effectively suppressing impulse noise. Even at high noise density levels the MDBUTMF gives better results in comparison with other existing algorithms. The proposed algorithm is effective for salt and pepper noise removal in images at high noise densities. The experimental results included in this project have demonstrated that the proposed filter significantly outperforms a number of well-accepted decision- based filters.

V. Future Scope

In the Transmission of **Videos** over channel, Video frames are corrupted by salt and pepper noise (Impulse Noise), due to faulty communication systems. With this project we can implement a better filtering technique that makes the noisy video frames to noise free video frames. Median filters are the best known non-linear digital filters based on order statistics to solve the present problem in videos. Median filters are known for their capability to remove salt and pepper noise and preserves the shape. The noise detection process to discriminate between uncorrupted pixels and the corrupted pixels prior to applying non-linear filtering is highly desirable to protect the signal details of uncorrupted pixels. We proposed A Modified Decision Based Unsymmetrical Trimmed Median filter **(MDBUTM)** algorithm for the restoration of gray scale, and color video frames that are highly corrupted by salt and pepper noise.

References

Journal Papers:

[1] J. Astola and P. Kuosmaneen, Fundamentals of Nonlinear Digital Filtering. Boca Raton, FL: CRC, 1997.

[2] 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.

[3] 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.

[4] P. E. Ng and K. K. Ma, “A switching median filter with boundary discriminative noise detection for extremely corrupted images,” IEEE Trans. Image Process., vol. 15, no. 6, pp. 1506–1516, Jun. 2006.

[5] K. S. Srinivasan and D. Ebenezer, “A new fast and efficient decision based algorithm for removal of high density impulse noise,” IEEE Signal Process. Lett., vol. 14, no. 3, pp. 189–192, Mar. 2007.

[6] V. Jayaraj and D. Ebenezer, “A new switching-based median filtering scheme and algorithmfor removal of high-density salt and pepper noise in image,” EURASIP J. Adv. Signal Process., 2010.

[7] K. Aiswarya, V. Jayaraj, and D. Ebenezer, “A new and efficient algorithm for the removal of high density salt and pepper noise in images and videos,” in Second Int. Conf. Computer Modeling and Simulation,2010, pp. 409–413

Books:

[8] RafelC.Gonzalez and Richard E.Woods, “Digital Image Processing” Pearson Education, 2011

[9] S.Jayaraman, S.Esakkirajan, T.Veerakumar, “Digital Image Processing’’Mc Graw Hill Publishers, 2009