

Implementation of De-noising Algorithm for Removal of Impulse Noise in Image using FPGA

K. SHASHIDHAR

Associate Professor, Dept of ECE, Dhruva Institute of Engineering & Technology, JNTUH, Hyderabad, India,
Email: k_shashidhar@yahoo.com.

Abstract: A Proficient of a modified decision based unsymmetrical trimmed median filter algorithm for the reclamation of gray scale pictures and features that are profoundly tainted by salt and pepper noise is proposed in this paper. The noise pixels can take just the greatest or least values (i.e. 0 or 255 for 8-bit grayscale images). Noise evacuation can be attained, by utilizing various existing linear filtering techniques. Median channels are known for their ability to evacuate motivation noise without harming the edges. This filter first recognizes conceivable noise pixels and afterward replaces them utilizing the average channel or its variations, while leaving all different pixels unaltered. This paper recommends an advanced construction modeling for channel execution on FPGA. A 3×3 sliding window calculation is utilized as the base for channel operation. Partial execution is carried out through soft core processor. The plans are orchestrated for a Xilinx Spartan-3 EDK. The proposed calculation is tried against distinctive grayscale pictures at diverse noise levels and it gives better Peak Signal-to-Noise Ratio (PSNR) and mean square error (MSE).

Keywords: FPGA, MATLAB, Median Filter, Salt and Pepper Noise, SPARTAN3EDK. Xilinx Platform Studio.

I. INTRODUCTION

Digital images play a very important part both in applications such as television magnetic resonance imaging computer tomography as well as in field of science and technology such as geographical information system and astronomy. Sets of data collected by image sensors and other devices are generally contaminated by noise. Also noise can introduced due to transmission errors and compression. Hence denoising is often a necessary and first step to be performed before image data is analyzed and processed. An efficient denoising technique must be applied to compensate for such data corruption. Noise is generally modeled as Gaussian noise (Normal), Uniform noise and Impulse noise (salt and pepper noise). The impulse noise is of two types, Fixed valued and random valued. The fixed valued impulse noise is also known as salt and pepper noise which can have value either 0 or 255. Here 0 represent complete black and 255 represent complete white on gray scale image. The random valued impulse noise can have any value between 0 and 255; hence its removal is very important and difficult. Image de-noising is an important pre-processing step for image analysis. It refers to the task of recovering a good estimate of the true image from a degraded observation without altering and changing useful structure in the image such as discontinuities and edges.

Image denoising still remains an important challenge for researchers because denoising process removes noise but introduces artifacts and also causes blurring. Several nonlinear filters have been proposed for restoration of images

contaminated by salt and pepper noise. Among these standard median filter has been established as reliable method to remove the salt and pepper noise without damaging the edge details. However, the major drawback of standard Median Filter (MF) is that the filter is effective only at low noise densities. When the noise level is over 50% the edge details of the original image will not be preserved by standard median filter. Adaptive Median Filter (AMF)[3] performs well at low noise densities. But at high noise densities the window size has to be increased which may lead to blurring the image. In switching median filter, the decision is based on a pre-defined threshold value. The major drawback of this method is that defining a robust decision is difficult. Also these filters will not take into account the local features. As a result of which details and edges may not be recovered satisfactorily, especially when the noise level is high. To overcome the above drawback, Decision Based Algorithm (DBA) is proposed. In this, image is denoised by using a 3×3 window. If the processing pixel value is 0 or 255 it is processed or else it is left unchanged.

II. PROPOSED AGORITH

The proposed Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) Calculation forms the corrupted pictures by first distinguishing the impulse noise. The processing pixel is checked whether it is uproarious or noisy free. That is, if the transforming pixel lies in the middle of most extreme and minimum gray level values then it is noise free pixel, it is left unaltered. In the event that the

transforming pixel takes the greatest or least gray level then it is noisy pixel which is transformed by MDBUTMF.

Step 1: The MDBUT Median Filter selects a 2D-window of size 3×3. The center pixel in the selected window is the processing pixel and it is denoted as P_{ij} as shown in Figure 1. The neighboring pixels of the processing pixel P_{ij} are present in the directions NW, N, NE, W, E, SW, S, and SE. The positions of these directions are $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$ and $(i+1, j+1)$ respectively. The directions are clearly mentioned in the following Fig1.

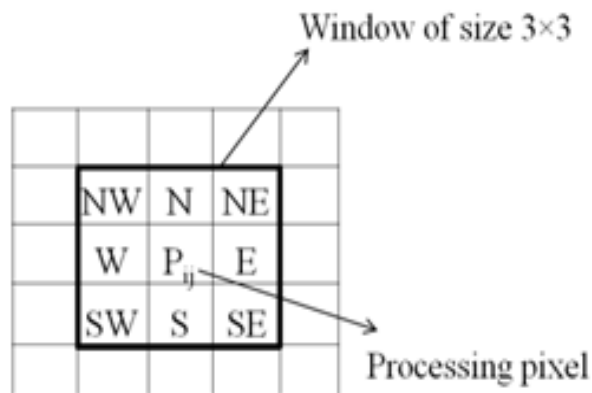


Fig.1-D-window of size 3×3.

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

Step 3: If $P_{ij} = 0$ or $P_{ij} = 255$ then P_{ij} is a 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 P_{ij} with the mean of the element of 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 P_{ij} with the median value.

The pictorial representation of each case of the proposed algorithm is shown in flow chart shown in fig.2.

Step 4: Repeat steps 1 to 3 until all the pixels in the entire image is processed. Every last pixel of the picture is checked for the vicinity of salt and pepper noise. Diverse cases are delineated beneath. On the off chance that the preparing pixel is noisy and all other pixel qualities are either 0's or 255's is represented in Case i). On the off chance that the processing pixel is noisy pixel that is 0 or 255 is outlined in Case ii). If the processing pixel is not noisy pixel and its esteem lies somewhere around 0 and 255 is outlined 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:

$$\begin{bmatrix} 0 & 255 & 0 \\ 0 & 255 & 255 \\ 255 & 0 & 255 \end{bmatrix} \quad (1)$$

Where "255" is processing pixel, i.e., P_{ij} .

Since all the elements surrounding P_{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.

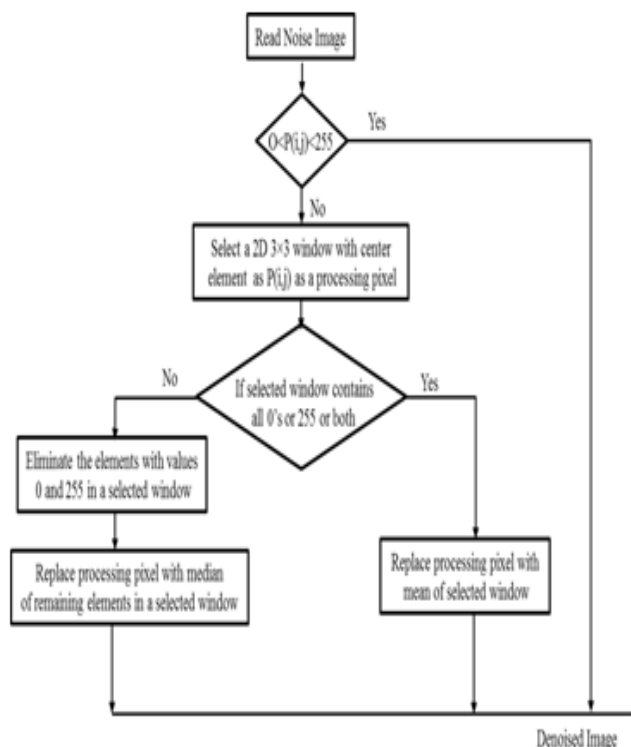


Fig.2. Flow Chart of MDBUTMF.

- 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:

$$\begin{bmatrix} 78 & 90 & 0 \\ 120 & 0 & 255 \\ 97 & 255 & 73 \end{bmatrix} \quad (2)$$

Where "0" is processing pixel, i.e., P_{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 [78 90 0 120 0 255 97 255 73]. After elimination of 0's and 255's the pixel values in the selected window will be [78 90 120 97 73]. Here the median value is 90. Hence, replace the processing pixel P_{ij} by 90.

- 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 90 then it is noise free pixel:

Implementation of De-noising Algorithm for Removal of Impulse Noise in Image using FPGA

$$\begin{bmatrix} 43 & 67 & 70 \\ 55 & 90 & 79 \\ 85 & 81 & 66 \end{bmatrix} \quad (3)$$

Where "90" is processing pixel, i.e., P_{ij} .

Since "90" is a noise free pixel it does not require further processing.

III. BLOCKDIAGRAM AND ARCHITECTURE OF DENOISING ALGORITHM

The median is characterized as the middle of a group of numbers when the numbers are sorted. The gathering ought to contain odd number of components. For the 2D picture, a standard median operation is actualized by sliding a window of odd size (e.g. 3x3 windows) over a picture.

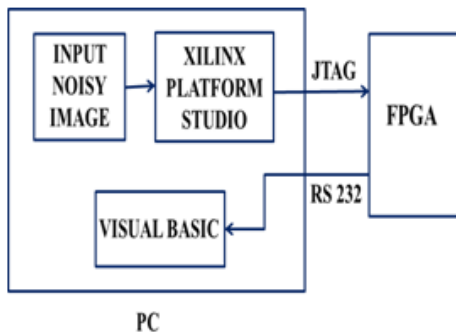


Fig.3. Block Diagram.

A 3x3 window size is picked which is viewed as powerful for most regularly utilized picture sizes. At each one position of the window, the nine pixels values inside that window are replicated and sorted. The estimation of the focal pixel of the window is supplanted with the average estimation of the nine pixels in the window as shown in Fig.3.

A. Architecture

To build an embedded system on Xilinx FPGAs, the embedded development kit (EDK) is used to complete the reconfigurable design. Dissimilar to the outline stream in the conventional programming configuration utilizing C/C++ language or hardware description languages, the EDK empowers the mix of both hardware and programming parts of an installed framework. For the hardware side, the configuration entrance from VHDL/Verilog is initially synthesized into a gate-level net list, and after that interpreted into the primitives, mapped on the particular gadget resources, for example, Look-up tables, flip-flops, and block memories. The area and interconnections of these gadget assets are then set and steered to meet with the timing Constraints. A downloadable .bit file is made for the entire equipment stage. The software side takes after the standard installed programming stream to gather the source codes into an executable and linkable file (ELF) format. In the meantime, a microprocessor software specification (MSS) document and a microprocessor hardware specification (MHS) record are utilized to characterize programming structure and equipment association of the framework. The EDK utilizes these records to control the outline stream and inevitably consolidate the framework into a solitary

downloadable record. The entire configuration runs on a real-time operating system (RTOS).

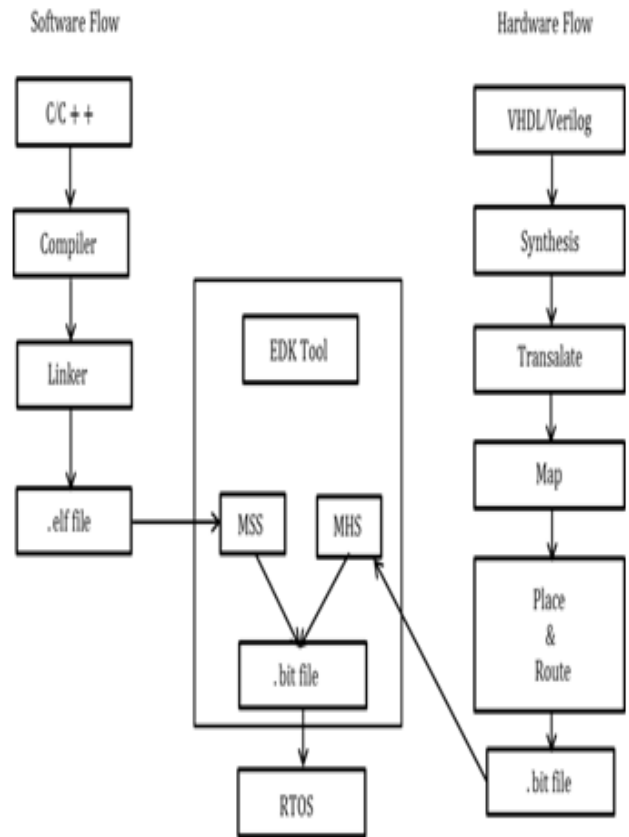
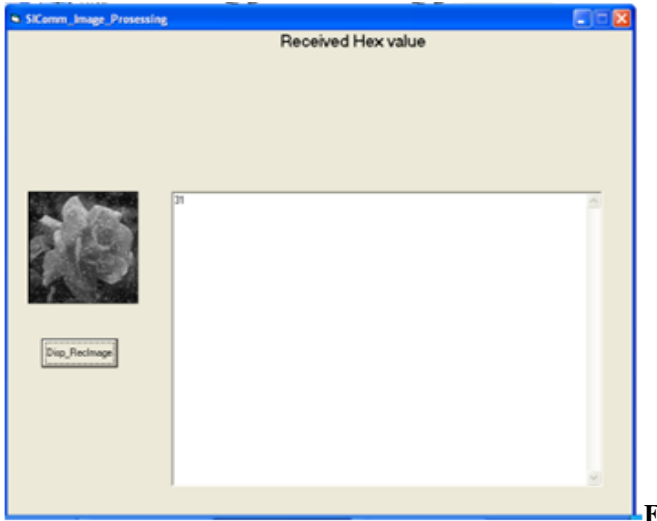


Fig.4. Embedded Development Kit Design Flow.

There are diverse approaches to incorporate processors inside Xilinx FPGA for System-on-a-Chip (SOC) Power PC hard processor center, or Xilinx Micro blaze soft processor core, or user-defined soft processor core in VHDL/Verilog. In this work, The 32-bit Micro blaze processor is picked on account of the adaptability as shown in figure 3.6. The client can tailor the processor with or without development peculiarities, in light of the financial backing of equipment. The advance features include memory management unit, floating processing unit, hardware multiplier, hardware divider, instruction and data cache links etc. The building design diagram of the framework is demonstrated in Fig.4. It can be seen that there are two separate buses (i.e., processor local bus (PLB) and fast simplex link bus (FSL Bus) utilized as a part of the framework. PLB takes after IBM Core join transport structural engineering, which backs high bandwidth master and slave devices, gives up to 128-bit data bus, up to 64-bit address bus and centralized bus intervention. It is a sort of shared transport. Other than the right to gain entrance overhead, PLB conceivably has the danger of incongruous hardware/software because of transport discretion. On the other hand, FSL supports point-to-point unidirectional communication.

IV. SIMULATION RESULTS

Experiments are performed on gray level images to verify the proposed method. These images are represented by 8 bits/pixel and size is 128 x 128. Image used for the experiment is shown in following fig.5.



ig.5. Input Image.

The entropy (E) is defined as the average amount of information contained in each message received where s is the set of processed coefficients and p (e) is the probability of processed coefficients. By using entropy, number of bits required for compressed image is calculated. The Quantitative performance of the proposed algorithm is evaluated based on Peak signal to noise ratio (PSNR) ,Mean Square Error (MSE) and Image Enhancement Factor (IEF) which is given in equations 1 and 2 respectively.

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \quad (4)$$

Where $MSE = \sum_j \sum_j (r_{ij} - x_{ij})^2$

$$IEF = \frac{(\sum_j \sum_j (n_{ij} - r_{ij})^2)}{(\sum_j \sum_j (x_{ij} - r_{ij})^2)} \quad (5)$$

Where 'r' refers to Original image, 'n' gives the corrupted image 'x' is denotes restored image, M x N is the size of processed image as shown in Fig.6.



Fig.6. Output Image.

```

Selected Device : 3a500efg320-4

Number of Slices:          2649 out of 4656  56%
Number of Slice Flip Flops: 3343 out of 9312  35%
Number of 4 input LUTs:   3794 out of 9312  40%
  Number used as logic:    3118
  Number used as Shift registers: 356
  Number used as RAMs:     320
Number of IOs:            83
Number of bonded IOBs:    40 out of 232  17%
  IOB Flip Flops:         55
Number of BRAMs:          7 out of 20  35%
Number of MULT18X18SIOs:  3 out of 20  15%
Number of GCLKs:          7 out of 24  29%
Number of DCMs:           2 out of 4  50%

Timing Summary:
-----
Speed Grade: -4

Minimum period: 12.384ns (Maximum Frequency: 80.749MHz)
Minimum input arrival time before clock: 41.553ns
Maximum output required time after clock: 13.840ns
Maximum combinational path delay: 3.344ns
    
```

Fig.7.Synthesis report.

A. Comparison of Filters

Now, compare the proposed algorithm i.e., Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) with the existing algorithm like SMF (Standard Median Filter) and AMF (Adaptive Median Filter) using MATLAB with the parameters like Peak signal-to-noise ratio (PSNR) Mean Square Error (MSE) in dB.

TABLE I: Comparison of PSNR values for Different Algorithms

% of Noise	PSNR		
	SMF	AMF	MDBUTMF
10	41.32	46.87	52.74
30	41.02	45.38	49.4
50	40.61	43.98	46.65
70	40.05	42.67	45.21
90	39.49	42.94	43.96

Comparison of PSNR values for SMF and AMF and the proposed filter algorithms are given in table1. From the above table, it clearly shows that the proposed system has high value of Peak Signal to Noise Ratio (PSNR). Therefore, we can conclude that the proposed system is better than existing filters.

TABLE II: Comparison of MSE values for Different Algorithms

% of Noise	MSE		
	SMF	AMF	MDBUTMF
10	4.79	1.33	0.58
30	5.13	1.88	1.23
50	5.65	2.6	2.1
70	6.42	3.59	3.39
90	7.33	4.25	4.02

Comparison of MSE values for SMF and AMF and the proposed filter algorithms are given in table2. From the above table, it clearly shows that the proposed system has

Implementation of De-noising Algorithm for Removal of Impulse Noise in Image using FPGA

low value of Mean Square Error (MSE). Therefore, we can conclude that the proposed system is better than existing filters.

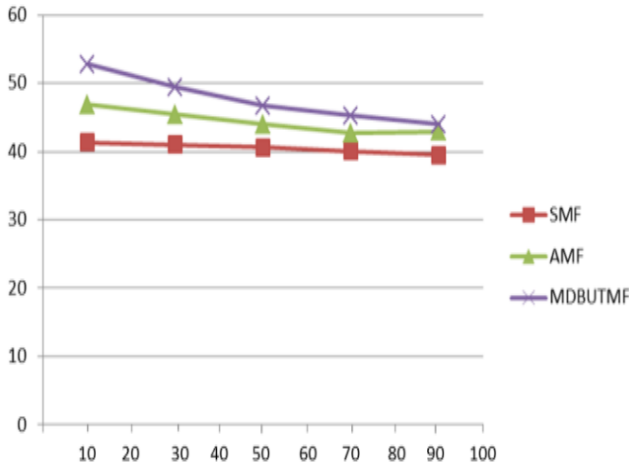


Fig.8. Comparison of PSNR values for Different Algorithms.

From the above Fig.8, it clearly shows that the proposed system has low value of Mean Square Error (MSE) than the existing filters like SMF and AMF. Therefore, we can conclude that the proposed system is better than existing filters.

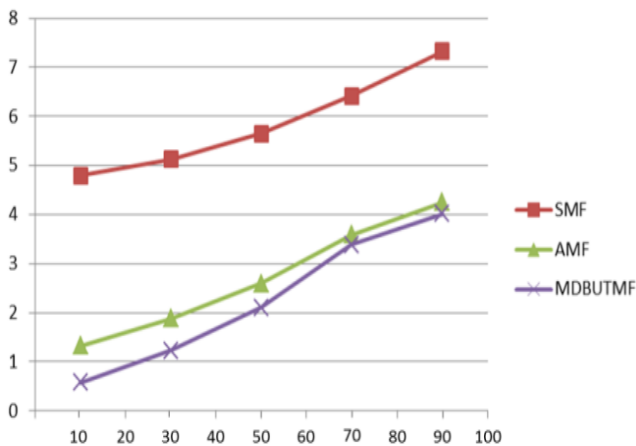


Fig.9. Comparison of MSE values for Different Algorithms.

From the above Fig.9, it clearly shows that the proposed system has high value of Peak Signal to Noise Ratio (PSNR) than the existing filters like SMF and AMF. Therefore, we can conclude that the proposed system is better than existing filters.

V. CONCLUSION

In this paper, a new algorithm (MDBUTMF) is proposed which gives better performance in comparison with MF, AMF and other existing noise removal algorithms in terms of Peak signal to noise ratio (PSNR) and Image Enhancement Factor (IEF). The performance of the algorithm has been tested at low, medium and high noise densities on both gray-scale and color images. Even at high noise density levels the MDBUTMF gives better results in comparison with other existing algorithms. Both visual and quantitative results are

demonstrated. The proposed algorithm is effective for salt and pepper noise removal in images at high noise densities. We have also presented an alternative implementation of median filtering for arbitrarily large windows. The architecture is immune to changes in window size, the area being determined solely by the bit width. This allows for a flexible window-size that can change from one calculation to another and we finally presented the results which are implemented on the Spartan-3 EDK evolution board.

VI. REFERENCES

- [1] S. Esakkirajan, T. Veerakumar, Adabala N. Subramanyam, and C. H. Prem Chand, "Removal of High Density Salt and Pepper Noise Through Modified Decision Based Unsymmetrical Trimmed Median Filter", IEEE Signal Processing letters, vol. 18, no. 5, may 2011.
- [2] J. Astola and P. Kuosmanen, Fundamentals of Nonlinear Digital Filtering. Boca Raton, FL: CRC, 1997.
- [3] H. Hwang and R. A. Haddad, "Adaptive median filter: New algorithms and results," IEEE Trans. Image Process., vol. 4, no. 4, pp. 499–502, Apr. 1995.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] V. Jayaraj and D. Ebenezer, "A new switching-based median filtering scheme and algorithm for removal of high-density salt and pepper noise in image," EURASIP J. Adv. Signal Process, 2010.
- [8] 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.
- [9] V. Satyanarayana, S. Srividya, U. Yedukondalu, "High Throughput Two-Dimensional Median Filters On FPGA for Image Processing Applications", International Journal of Engineering Research and Applications, Vol. 3, Issue 4, Jul-Aug 2013, pp. 1946- 1949.
- [10] Kati ke Abdulla, Syed Ahmed Pasha, K. Sudhakar, "Hardware Software Cosimulation Of Image Noise Cancellation By Adaptive Median Filter And Image Enhancement Using Sobel Edge Detection", Global Journal of Advanced Engineering Technologies, Vol2, Issue4.
- [11] Priya Kapoor, Samandeep Singh, "An Improved Modified Decision Based Filter to Remove High Density Impulse Noise", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 7, July 2014.
- [12] K. Santhosh Kumar, M. Gopi, "Removal of High Density Salt and Pepper Noise through Modified Decision based Unsymmetrical Trimmed Median Filter", International Journal of Science and Research, Volume 2 Issue 3, March 2013.

K. SHASHIDHAR

[13] R. Babul Reddy, N. Dinesh Kumar, "An Efficient Denoising Architecture for the Removal of Impulse Noise in Image Using FPGA" in Global Journal of Advanced Engineering Technologies, Vol3, Issue4- 2014, pp 380-385.

Author Profile:



K. Shashidhar completed M.Tech in VLSI System Design, working as an Associate professor for the department of Electronics & Communication Engineering in "Dhruva Institute of Engineering and Technology", Near to Ramoji film city, affiliated to JNTU, Hyderabad, Email-id:

k_shashidhar@yahoo.com.