Imperial Journal of Interdisciplinary Research (IJIR)
Vol-2, Issue-7, 2016
ISSN: 2454-1362, http://www.onlinejournal.in

Implementation of Multiple Threshold MDBUT Filter for Salt-Pepper Noise Removal in Low Resolution Images

N.Chidambaram[1], S.Monica[2], C.Nishantni[3], M.Ragavi[4] & J.Reena Celene[5]
[1]Associate Professor , [2,3,4,5] U.G. Final Year Students
Electronics and Communication department, Panimalar Engineering College, Chennai

***Abstract**: It is important to enhance the quality of blurred monochrome images captured using low resolution monitoring or Surveillance cameras. The images, thus captured suffer from noise and degradation leading to loss or misconception of information. The images are usually available in fixed and variable scale. In this paper, we propose a technique for digital image restoration for variable scale low resolution images. Median filtering is applied to multiple threshold value of images. The result of filtering is noisy and noiseless pixels. The noiseless pixels are preserved first and restoration is applied only to the noisy pixels. Hardware implementation of Multiple Threshold MDBUT filter demonstrates that this method performs well than many existing decision-based methods. The analysis of FPGA implementation of the filter clearly shows that this algorithm is effective for blurred images captured by low resolution monitoring cameras corrupted by salt-pepper noise.*

Keywords: Median Filter, Salt-Pepper noise, MDBUTMF, Denoising

1. Introduction

An image can be defined as a two-dimensional signal (digital), that contains grayscale or colour information arranged along an x and y spatial axis. Images may be two dimensional or three dimensional. Image is represented as a rectangular grid of pixels. And it has a definite height, size and colour. Every pixel is a square picture element and it is fixed or variable size based on types of display. Each pixel consists of numbers representing brightness and colour in terms of magnitude. Group of four 8 bits, represents information. These pixels values tend to change due to less intensity and low resolution camera images. Although several techniques available to suppress various noises [2][3], this work concentrates only on removal of salt-pepper noise in common JPEG file format.

2. Image File Formats

JPEG/JFIF

JPEG (Joint Photographic Experts Group) is a compression method. JPEG compressed images are usually stored in the JFIF (JPEG File Interchange Format) file format. JPEG compression is lossy compression. Nearly every digital camera can save images in the JPEG/JFIF format, which supports 8 bits per colour (red, green, blue) for a total of 24-bits, producing relatively small files. Photographic images may be better stored in a lossless non-JPEG format. [10]

EXIF

The EXIF (Exchangeable image file format) format is a file standard similar to the JFIF format with TIFF extensions. It is incorporated in the JPEG writing software used in most cameras. Its purpose is to record and to standardize the exchange of images with image metadata between digital cameras and editing & viewing software. The metadata are recorded for individual images and include such things as camera settings, time &date, shutter speed, exposure, image size, compression, name of camera, colour information, etc. [10]

TIFF

The TIFF (Tagged Image File Format) format is a flexible format that normally saves 8 bits or 16 bits per colour (red, green, blue) for total of 24-bits and 48-bit both respectively, usually using either the TIFF or TIF filename extension. TIFFs are of lossy and lossless type. Some offer relatively good lossless compression for bi-level (black & white) images. Some digital cameras can save in TIFF format, using the LZW compression algorithm for lossless storage. TIFF image format is not widely supported by web browsers. TIFF remains widely accepted as a photograph file standard in the printing business. TIFF can handle device-specific colour spaces, such as the CMYK defined by a particular set of printing press inks [10]

GIF

GIF (Graphics Interchange Format) is limited to an 8-bit palette, or 256 colours. This makes the GIF format suitable for storing graphics with relatively few colours such as simple diagrams, shapes, logos and cartoon style images. The GIF format supports animation and is still widely used to provide image animation effects. It also uses a lossless compression that is more effective when large areas have a single colour, and ineffective for detailed images or dithered images [10]

3. Salt-Pepper Noise

Common form of noise is data drop-out noise (commonly referred to as intensity spikes, speckle or salt and pepper noise). Here, the noise is caused by errors in the data transmission [3]. The corrupted pixels are either set to the maximum value (which looks like snow in the image) or have single bits flipped over. In some cases, single pixels are set alternatively to zero or to the maximum value, giving the image a ‘salt and pepper' like appearance. Unaffected pixels always remain unchanged. The noise is usually quantified by the percentage of pixels which are corrupted [9].

4. MDBUT Median Filter

The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether or not it is representative of its surroundings. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used.) [5]. Fig. 4.1 illustrates an example calculation. As can be seen the central pixel value of 150 is rather not representative of the surrounding pixels and is replaced with the median value: 124.
Neighborhood values:
115, 119, 120, 123, 124, 125, 126, 127, 150
Median value: 124

123	125	126	130	140
122	124	126	127	135
118	120	150	125	134
119	115	119	123	133
111	116	110	120	130

Figure 4.1

A 3×3 square neighborhood is used here larger neighborhoods will produce more severe smoothing. A Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) is proposed for the restoration of colour images that are highly corrupted by salt and pepper noise [2][4]. The proposed filter (MDBUTMF) replaces the noisy pixel by trimmed median value when some of the elements with values 0’s and 255’s are present in the selected window. If all the pixel values in the selected window are 0’s and 255’s means then the noisy pixel is replaced by mean value of all the elements present in that selected window. In the proposed method first the noisy image is read then based on some decision salt and pepper noise detection takes place. At the end of the detection stage the noisy and noise-free pixels get separated. The noise-free pixel is left unchanged and the noisy pixel is given to the Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) [1]. The data flow of the proposed method is shown in Fig. 4.2.

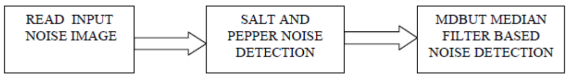


Figure 4.2

The flow goes in the way that, first the noisy image is given to a noisy image reader. Followed by this is the salt and pepper noise detection. After this, based on the state of the elements the corrupted pixel is either replaced by Type-I or replaced by Type-II. Type I: If the selected window contains all the elements as 0’s and 255’s means, then replace the processing pixel by the mean value of the elements present in that window. Type-II: If the selected window contains not all elements as 0’s and 255’s. Then eliminate 0’s and 255’s and find the median value of the remaining elements. Replace the processing pixel with the median value. The clear explanation of Type-I and Type-II with examples is given in this Section. The output images produced by the combination of Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) contain excellent Result than the existing methods. The proposed Modified Decision Based Unsymmetrical Trimmed Median Filter (MDBUTMF) algorithm processes the corrupted images by first detecting the impulse noise. The processing pixel is checked whether it is noisy or noisy free. That is, if the processing pixel lies between maximum and minimum gray level values then it is noise free pixel, it is left unchanged. If the processing pixel takes the maximum or minimum gray level then it is noisy pixel which is processed by MDBUTMF.

5. 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. Comparison of PSNR values is shown in Table 5.1.

Table 5.1

% of Noise	PSNR		
	SMF	AMF	MDBUTMF
10	41.32	46.87	52.7
30	41.02	45.38	49.4
50	40.61	43.98	46.65
70	40.05	42.67	42.21
90	39.49	42.94	43.96

Comparison of PSNR values for SMF and AMF and the proposed filter algorithms are given in table 5.1. From the above table, it clearly shows that the proposed system has high value of Peak Signal to Noise Ratio (PSNR) [4],[6]. Comparison of MSE values for SMF and AMF and the proposed filter algorithms are given in Table 5.2.

Table 5.2

% 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

6. Proposed System Block Diagram

As with the reference of Fig 6.1, Adaptive Median is a decision-based filter that first identifies possible noisy pixels and then replaces them using the median filter or its variants, while leaving all other pixels unchanged. This filter is good at detecting noise even at a high noise level. The difference lies in on the type of filter we use after binary masking.

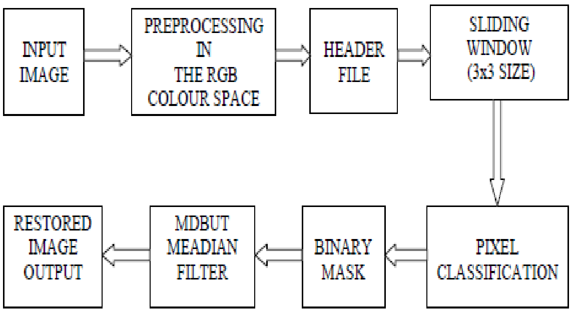


Figure 6.1

7. Sliding Window

A sliding window (block) system incorporating a methodology for providing a processor access to image data is described [7]. An image is received for processing that has a size that is too large for the processor to access directly. As a result, the sliding window system creates first, second, and third swappable windows (blocks) as shown in the Fig 7.1

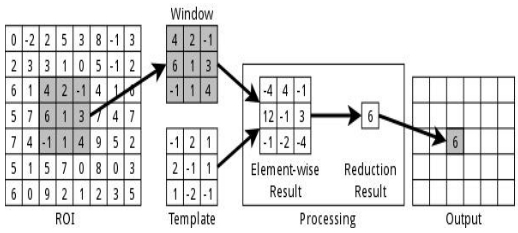


Figure 7.1

for accessing image data from the image; each windows is swappable so that any two are available within the memory space of the processor while a third is being loaded in a background memory. The system cycles through the three windows such that, at any given point in time, two of the three windows are affixed in the memory space of the processor as left and right adjacent windows, while the remaining or third window is being loaded in the background (e.g.,, in a DRAM) as a temporary shadow or background window. After the shadow window is loaded with appropriate image data, it is brought into the foreground (i.e., within the memory space of the processor) as the new right window. The prior (old) right window now becomes the new left window; the prior (old) left window now becomes a new shadow or background window. The process repeats as necessary, until all image data of the target image has been accessed/ processed.

8. Denoising Algorithm

In this work, the denoising process is carried out by capturing the image directly or taking snap shot of the video especially captured by monitoring/spy cameras and giving it to the FPGA module using a JTAG cable. The image processed is then sent to PC with front end supported by visual basic. We used serial port interface with RS 232 Male-Female connector on either end of the communication between PC and FPGA after processing the image. This is shown in Fig. 8.1.

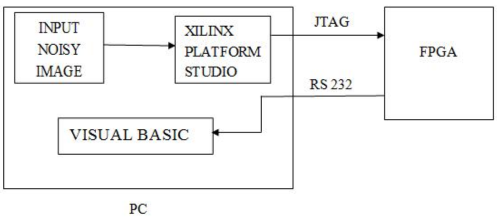


Figure 8.1

9. Hardware-Software Interface

To build an embedded system on Xilinx FPGAs, the embedded development kit (EDK) is used to complete their configurable design [4]. 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).

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.

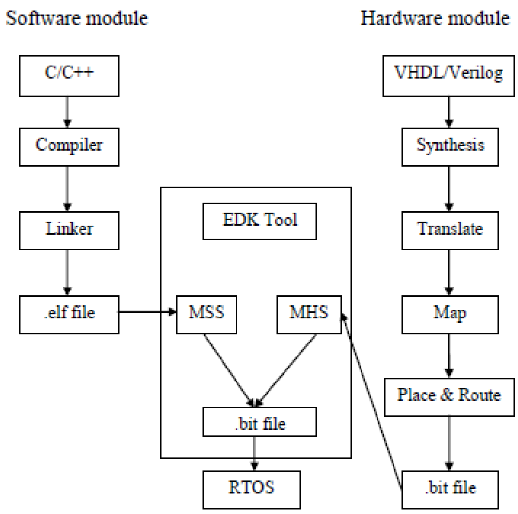


Figure 9.1

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. 9.1.

10. Design of Proposed Median Filter

The denoising algorithm is used in FPGA hardware where the input noise image is observed. Intentionally added salt-pepper noise can be introduced for the testing purpose using MATLAB. The proposed filter uses multiple thresholds to classify the signal as either noise-free or noise-corrupted so that only noisy signals are filtered while good signals are preserved.

In the detection process, if $x(k)$ is judged to be the maximum or minimum in the filter window, then the decision rules (threshold functions) are used on the neighboring pixels of $x(k)$ to decide whether it is a noise corrupted pixel. To identify corrupted pixels, input pixels can be separated into two classes, A and B. Pixels in class A are supposedly much more likely to be impulses than those in class B. To make sure that this happens, first, we check $x(k)$ to see whether it is a maximum or minimum in the filter window. If $x(k)$ is a maximum or minimum, it will be classified into class A, otherwise it will be classified into class B. When $x(k)$ is classified into class A, the pixels of the 3×3 filter window (excluding $x(k)$) are sorted in ascending order.

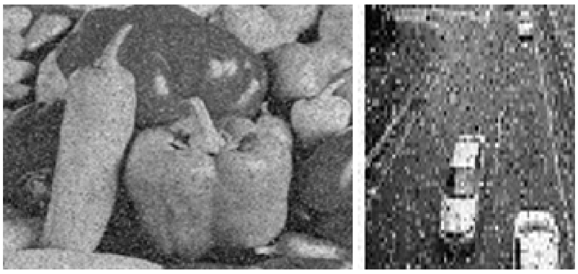
The sorted vector can be defined as

$$s(k) = (s_1(k), s_2(k), \ldots, s_8(k))$$

Where $s_1(k)$, $s_2(k)$, …, $s_8(k)$ are the elements of $w(k)$ arranged in ascending order. The differences between the input pixel $x(k)$ and each of the elements of $s(k)$ provide an efficient measurement to identify noisy pixels. The Center Weighted Median Filter (CWM) designed outperforms many filters for the same noise removal.

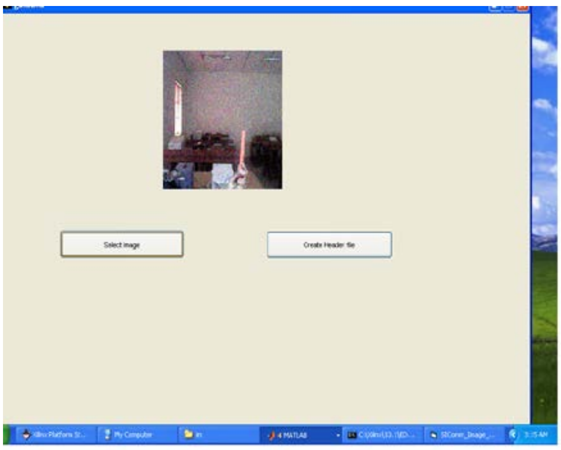
11. Results and Screen Shots

Sample Noise Input Monochrome Images

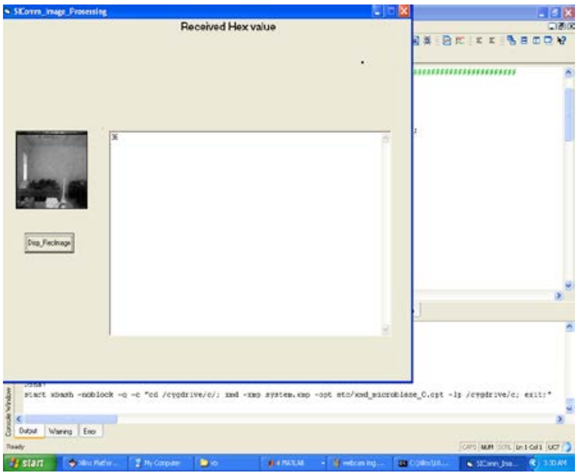


Denoised Output Images

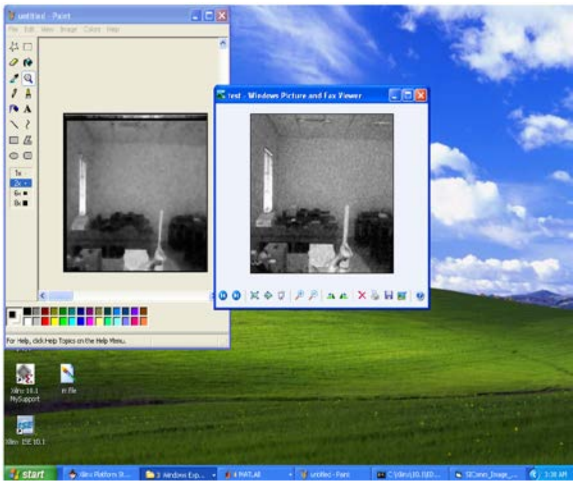
Sample Colour Image Captured by Web Cam



Received Hex Value



Noise Input and Output Image on VB Front end



12. Conclusion

In this paper, we have presented a new efficient decision-based filter, the multiple threshold MDBUT filter, for image restoration. 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. The experimental results included in this project have demonstrated that the proposed filter significantly outperforms a number of well accepted decision-based filters. Furthermore, this filter has the potential to be applied to real-time video signals captured by monitoring cameras. The filter has potential applications in toll-booth, recognition of tumor in medical diagnosis and restoration images [6] corrupted by transmission noise. The work is carried out by taking monochrome images and design can be further extended to colour images.

13. References

[1] G. Pok, J. C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," *IEEE Transactions on Image Processing*, Vol. 12, 2003, pp. 85-92.

[2] Y.Lilly Margaret ,T.Vijaya Nirmala, "FPGA Implementation of removal of Impulse noise in images using MDBUT technique", IJIRCCE, Vol 2, Special issue 4, Sep 2014.

[3] Harish, M.R.Gautham, "The component median filter for noise removal in digital images, IJETT, Vol 4, issue 5, 2013.

[4] K.Shahidhar, "Implementation of denoising-algorithm for removal of impulse noise in image using FPGA," IJSETR, Vol 4, issue 11, May 2015

[5] T. C. Lin and P. T. Yu, "Adaptive two-pass median filter based on support vector machine for image restoration," *Neural Computation*, Vol. 16, 2004, pp. 333-354.

[6] T. Chen and R. W. Hong, "Application of partition-based median type filters for suppressing noise in images," *IEEE Transactions on Image Processing*, Vol.10,2001, pp. 829-836.

[7] E. Abreu, M. Lightstone, S. K. Mitra, and K. Arakawa, "A new efficient approach for the removal of impulse noise from highly corrupted images," *IEEE Transactions on Image Processing*, Vol. 5, 1996, pp. 1012-1025.

[8] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," *IEEE Transactions on Circuits Systems*, Vol. 38, 1991, pp.984-993.

[9] T. Chen, K. K. Ma and L. H. Chen, "Tri-state median filter for image denoising," *IEEE Transactions on Image Processing*, Vol. 8, 1999, pp. 1834-1838.

[10] https://en.wikipedia.org/wiki/Image_file_formats