Removal of High-Density Salt and Pepper Noise in Videos through MDBUTMF

V. Pranava Jyothy, K. Padmavathi

Abstract: A modified decision based unsymmetrical trimmed median filter (MDBUTM) algorithm for the restoration of gray scale, and color video's that are highly corrupted by salt and pepper noise is proposed in this paper. In the Transmission of Videos over channel, Video frames are corrupted by salt and pepper noise (Impulse Noise), due to faulty communication systems. The objective of this paper is to implement a better filtering technique that makes the noisy video frames to noise free video frames. The proposed algorithm replaces the noisy pixel by trimmed median value when 0's and 255's are present in the selected window and when all the pixel values are 0's and 255's then the noise pixel is replaced by mean value of all the elements present in the selected window. This proposed algorithm shows better results than the Standard Median Filter (MF), Decision Based Algorithm (DBA), Modified Decision Based Algorithm (MDBA), and Progressive Switched Median Filter (PSMF). The proposed algorithm is tested against different gray scale and color video frames and it gives better Peak Signal-to-Noise Ratio (PSNR) and Image Enhancement Factor (IEF).

Keywords: Median filter, salt and pepper noise, unsymmet- -rical trimmed median filter.

I. INTRODUCTION

Video frames are often corrupted by impulse noise. In general, the impulse noise in video frames is present due to bit errors in transmission or introduced during the signal acquisition stage. Based on the noise values, the impulse noise is classified in to two; they are salt and pepper noise and random valued noise. Salt and Pepper noise is easier to restore but, the random valued noise is more difficult to restore. Salt and Pepper noise can corrupt the frame where the corrupted pixel takes either maximum or minimum gray level. Many different non-linear filters have been proposed for restoration of salt and pepper noisy video frames. Among all the methods for removal of impulse noise, the median filter is used widely because of its effective noise suppression capability and high computational efficiency. Non-linear digital filters, based on order statistics are median filters (MF). Median filters are well known for their capability to remove impulse noise without damaging the edge information.

However, the major drawback of standard Median Filter (SMF) is that the filter is effective only at low noise densities. At high noise densities, SMF often exhibits blurring for large window sizes and insufficient noise suppression for small Window sizes. When the noise level is over 50% the edge details of the original frame will not be preserved by the Standard median filter (SMF). However, most of the median filters are operates uniformly across the video frames and it modifies both noise and noise-free pixels, causes information loss. Ideally, the filtering should be applied only to corrupted pixels but not to uncorrupted pixels. Adaptive median filter (AMF) works well at very low noise densities. Noise detection process to discriminate between uncorrupted pixels and the corrupted pixels prior to applying non-linear filtering is highly desirable Adaptive Median Filter is a decision based or switching filter that differentiates the noisy pixels and performs the filtering operation on them leaving all other pixels unchanged. These filters will not take into account the local features as a result of which details and edges may not be recovered satisfactorily, especially for high noise level cases. By providing enough large size windows, detection of impulse noise at high noise levels using AMF is reliable for simple background video frames.

Decision based algorithm (DBA) is proposed to overcome this problem. In this, selected video frame is de-noised by using a 3x3 window. The detection of noise and noise free pixels is decided by checking the value of a processed pixel element lies between the maximum and minimum values that occur inside the selected window. If the processing pixel value is 0 or 255 it is processed otherwise it is left unchanged. In addition, the DBA uses simple fixed length window of size 3x3, and hence it requires lower processing time while compared to AMF. But, at high noise density levels the median value may also be noisy. In this case, neighboring pixel is used for replacement. This repeated replacement of neighboring pixels produces streaking effect. Hence, details and edges are not recovered satisfactorily, especially when the noise level is high.

In order to avoid this drawback, Decision Based Un-symmetric Trimmed Median Filter (DBUTMF) is proposed At high noise densities, if the selected window contains all "0"s or "255"s or both then, trimmed median value cannot be obtained. So, this algorithm does not give better results at very high noise densities i.e. at >80%. Hence we proposed Modified Decision Based Un-symmetric Trimmed Median Filter (MDBUTMF) algorithm to overcome the problem with DBUTMF. We tested the algorithm on simple background, complex background video frames and it gives better Peak Signal-to-Noise Ratio (PSNR) and Image Enhancement Factor (IEF).

Revised Manuscript Received on 30 May 2013.
* Correspondence Author
V Pranava Jyothy, M.Tech in VLSI system Design from Indur Institute of Engineering and Technology, Siddipet Andhra Pradesh, India.
Assoc Prof. K Padmavathi, M.Tech with Indur Institute of Engineering and Technology, Siddipet Andhra Pradesh India

The rest of paper is structured as follows. A brief introduction of Salt and pepper noise is given in section II. Section III describes the concept of Unsymmetrical trimmed median filter. Section IV describes the overview of DBUTMF algorithm. Section V describes about the proposed algorithm and different cases of proposed algorithm and section VI describes detailed description of the proposed algorithm with an example is presented. Experimental setup is described in Section VII. Simulation results with different Video frames are presented in Section VIII. Finally conclusions are drawn in Section IX.

II.BRIEF REVIEW ON SALT AND 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. 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 a maximum (255) or minimum (0) 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.

In the following examples, Videos have been corrupted with various kinds and amounts of drop-out noise. In, pixels have been set to 0 or 255 with probability p=1%. In pixel bits were flipped with p=3%, and in 5% of the pixels (whose locations are chosen at random) are set to the maximum value, producing the snowy appearance.

For this kind of noise, conventional low pass filtering, *e.g.* mean filtering or Gaussian smoothing is relatively unsuccessful because the corrupted pixel value can vary significantly from the original and therefore the mean can be significantly different from the true value. Median filter removes drop-out noise more efficiently and at the same time preserves the edges and small details in the image better. Conservative smoothing can be used to obtain a result which preserves a great deal of high frequency detail, but is only effective at reducing low levels of noise.

III.UNSYMMETRIC TRIMMED MEDIAN FILTER

Median filters are widely used for smoothing operations in signal, speech and image processing. This filter operation is performed on an NxN image matrix ***I***[1...N, 1...N] using a *W*x*W* window where *W*=2w+1 is an odd number. The result of median filtering is an NxN matrix. The median for a 2-D frame is defined as:

To calculate the median

$$MEDIAN2D[i,j] = \text{median}\{\, I[a,b] \mid nbhd(a,i,w,N) \text{ and } nbhd(b,j,w,N) \,\}$$

Where,

$$nbhd\,(p,q,r,s) = \begin{cases} \text{True}; & (p-q) \bmod s \le r \text{ or } (q-p) \bmod s \le r \\ \text{False}; & \text{otherwise} \end{cases}$$

A trimmed filter rejects the noise pixel from the selected 3x3 window. Alpha Trimmed Median Filter (ATMF) is a symmetrical filter where the trimming is symmetric at either end. In this procedure, even the uncorrupted pixels are also trimmed. This leads to loss of video frame details and blurring of the video frames. In order to overcome this problem, an Un-symmetric Trimmed Median Filter (UTMF) is proposed. In UTMF, the selected 3x3 window elements are arranged in either increasing or decreasing order. Then the pixel values "0"s and "255"s which are causing salt and pepper noise, are removed from the video frame. The median value is then calculated for the remaining pixels. This median value is used to replace the noisy pixels. The pixel values 0‟s and 255‟s are removed from the selected window hence, we are calling it as trimmed median filter. This UTMF gives better noise removal than ATMF.

IV. DECISION BASED UNSYMMETRIC TRIMMED MEDIAN FILTER (DBUTMF)

Decision Based Unsymmetrical Trimmed median Algorithm is recently proposed algorithm to remove salt and pepper noise. In this algorithm each pixel is processed for de-noising using a 3x3 window. During processing if a pixel is "0" or "255" then it is processed else it is left unchanged treating the respective pixel as noise free. If the pixel is corrupted the corrupted pixel is replaced by the median of the window. At higher noise densities the median itself will be noisy, and the processing pixel will be replaced by the neighborhood processed pixel. This repeated replacement of neighborhood pixels produces streaking effect. The DBUTMF algorithm checks whether the left and right extreme values of the sorted array obtained from the 3x3 window are impulse values. The corrupted processing pixel is replaced by a mean value of the pixels in 3x3 windows after trimming impulse values. The corrupted pixel is replaced by the mean of the resulting array. 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.

V.PROPOSED ALGORITHM FOR VIDEOS

The Noisy video sequence is first converted into frames and again frames are converted into images. Then DBUTM Filtering algorithm is applied to the images which are separated from frames. This algorithm processes the corrupted images by first detecting the impulse noise. If the processing pixel is noisy then filtering is applied and noisy pixel is replaced by resulting pixel. If the processing pixel is noise free it is kept unchanged. After the filtering process, the frames are converted back to the original Video.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

ALGORITHM IMPLEMENTATION STEPS

1) Video to Frames: The noisy video sequence containing Impulse noise (Salt and Pepper noise) is converted into frames at 24 f/s rate.
2) Frames to Images: Frames are then converted in to .JPEG images to pass through the Filtering process to remove the Impulse noise presented.
3) Filtering: The impulse noise from the noisy images is removed using DBUTMF algorithm.
4) Frames to Video: After removal of impulse noise from all noisy images, the frames are converted back in to original video. The following figure 1 shows the flow chart of DBUTMF.

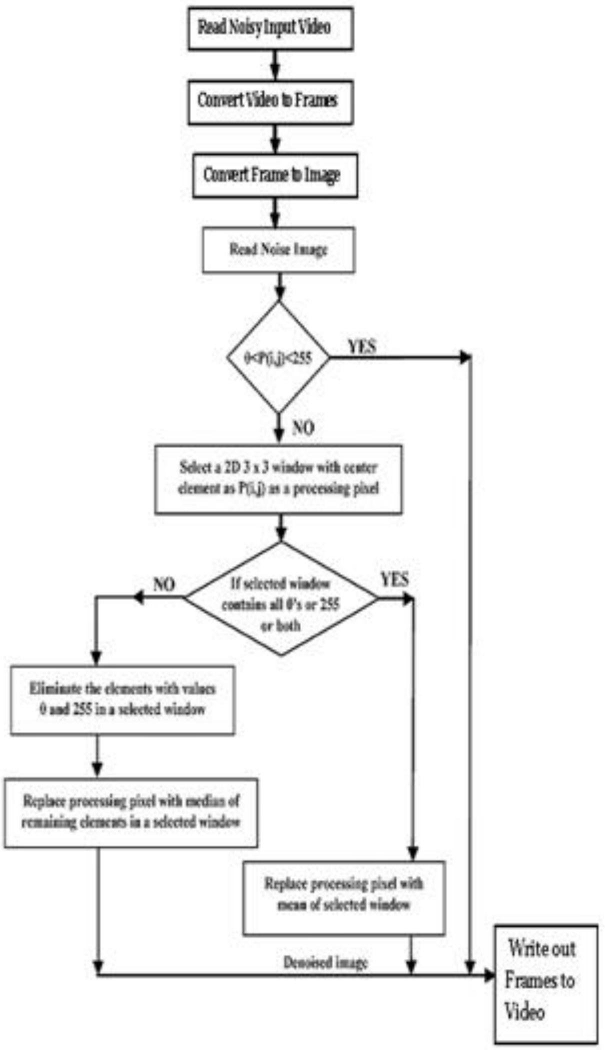

Fig.1. Flow Chart of MDBUTMF for videos

VI.ILLUSTRATION OF MDBUTMF ALGORITHM

Each and every pixel of the frame 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 and its value lies between 0 and 255 is illustrated in case (iii).

Case (i): If the selected window contains salt and pepper noise as processing pixel and neighboring pixel values contains all pixels that adds salt and pepper noise to the video frame.

$$\left\{ \begin{matrix} 0 & 255 & 0 \\ 0 & \langle 255 \rangle & 255 \\ 255 & 0 & 255 \end{matrix} \right\}$$

Where "255" is processing pixel.

Since all the elements surrounding **þ(x,y)** 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 and pepper noise as processing pixel and neighboring pixel values contains some pixels that adds salt and pepper noise to the video frame.

$$\left\{ \begin{matrix} 78 & 90 & 0 \\ 120 & \langle 0 \rangle & 255 \\ 97 & 255 & 73 \end{matrix} \right\}$$

Where "0" is the processing pixel.

Salt and pepper noise is now eliminated 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 **þ(x,y)** 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.

$$\left\{ \begin{matrix} 43 & 67 & 70 \\ 55 & \langle 90 \rangle & 79 \\ 85 & 81 & 66 \end{matrix} \right\}$$

Where "90" is the processing pixel.

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

VII. EXPERIMENTAL SETUP

A. Xilinx Platform Studio

The Xilinx Platform Studio (XPS) is the development environment or GUI used for designing the hardware portion of your embedded processor system.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

B. Embedded Development Kit

Xilinx Embedded Development Kit (EDK) is an integrated software tool suite for developing embedded systems with Xilinx MicroBlaze and PowerPC CPUs.

EDK includes a variety of tools and applications to assist the designer to develop an embedded system right from the hardware creation to final implementation of the system on an FPGA. System design consists of the creation of the hardware and software components of the embedded processor system and the creation of a verification component is optional. A typical embedded system design project involves: hardware platform creation, hardware platform verification (simulation), software platform creation and software verification. Base System Builder is the wizard that is used to automatically generate a hardware platform according to the user specifications that is defined by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system architecture, peripherals and embedded processors]. The Platform Generation tool creates the hardware platform using the MHS file as input. The software platform is defined by MSS (Microprocessor Software Specification) file which defines driver and library customization parameters for peripherals, processor customization parameters, standard 110 devices, interrupt handler routines, and other software related routines. The MSS file is an input to the Library Generator tool for customization of drivers, libraries and interrupts handlers.

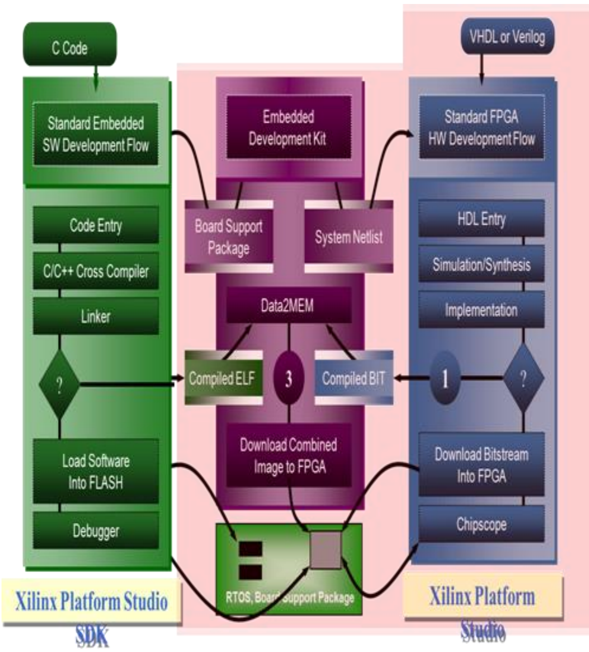

Fig.2. Embedded Development Kit Design Flow

The creation of the verification platform is optional and is based on the hardware platform. The MHS file is taken as an input by the Simgen tool to create simulation files for a specific emulator. Three types of simulation models can be generated by the Simgen tool: behavioral, structural and timing models. Some other useful tools available in EDK are Platform Studio which provides the GUI for creating the MHS and MSS files. Create / Import IP Wizard which allows the creation of the designer's own peripheral and import them into EDK projects.

Platform Generator customizes and generates the processor system in the form of hardware netlists. Library Generator tool configures libraries, device drivers, file systems and interrupt handlers for embedded processor system. Bitstream Initializer tool initializes the instruction memory of processors on the FPGA shown in figure2. GNU Compiler tools are used for compiling and linking application executables for each processor in the system. There are two options available for debugging the application created using EDK namely: Xilinx Microprocessor Debug (XMD) for debugging the application software using a Microprocessor Debug Module (MDM) in the embedded processor system, and Software Debugger that invokes the software debugger corresponding to the compiler being used for the processor. C. Software Development Kit Xilinx Platform Studio SDK is an integrated development environment, complimentary to XPS, that is used for C/C++ embedded software application creation and verification. SDK is built on the Eclipse open source framework. Soft Development Kit (SDK) is a suite of tools that enables you to design a software application for selected Soft IP Cores in the Xilinx EDK. The software application can be written in a "C or C++" then the complete embedded processor system for user application will be completed, else debug & download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device.

VIII. SIMULATION REPORTS AND RESULTS

The developed algorithm is tested on various simple and complex impulse noisy gray/color videos and compared with standard filters namely standard median filter (SMF), Adaptive median filter (AMF), decision based algorithm and decision based un-symmetric median filter (DBUTMF). Each time the test frame is corrupted by salt and pepper noise of different density ranging from 10% to 90% with an increment of 10% and it will be applied to various filters. In addition to image quality, the performance of the developed algorithm and other standard algorithms are quantitatively measured by the parameters such as Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE) and Image Enhancement Factor (IEF).

A. Performance Characteristics

Peak Signal to Noise Ratio (PSNR)

$$PSNR\ in\ dB = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad \text{---(1)}$$

Where *MSE* is the mean square error, and is defined as follows

$$MSE = \frac{\sum_x \sum_y \left(Y(x,y) - \hat{Y}(x,y) \right)^2}{MN} \quad \text{---- (2)}$$

Where *Y*(x,y) is the original video frame of size M x N.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The Image Enhancement Factor is a measure of Image quality, and is defined as

$$IEF = \frac{\sum_{x}\sum_{y}\left(\xi(x,y) - Y(x,y)\right)^2}{\sum_{x}\sum_{y}\left(\hat{Y}(x,y) - Y(x,y)\right)^2} \quad \text{----(3)}$$

The PSNR and IEF values of the proposed algorithm are compared against the existing algorithms by varying the noise density from 10% to 90% and are shown in Table I and Table II. From the Tables I and II, it is observed that the performance of the proposed algorithm (MDBUTMF) is better than the existing algorithms at both low and high noise densities. A plot of PSNR and IEF against noise densities for Baboon video frame is shown in Fig. 3 and 4.

The qualitative analysis of the proposed algorithm against the existing algorithms at different noise densities for Baboon video frame is shown in Fig 5 and 6. In this the video frame represents the processed frame using MF at 80% and 90% noise densities. Subsequent columns represent the processed Video frame for AMF, PSMF, DBA, MDBA and MDBUTMF. From the tables it is clear that MDBUTMF gives better PSNR values irrespective of the nature of the input Video frame.

B. Comparison of PSNR and IEF values

TABLE 1

Comparison of PSNR values of various algorithms for Video Frame one shown in Fig.6.

Noise %	PSNR in dB					
	MF	AMF	PSMF	DBA	DBUTMF	MDBUTMF
10	28.33	30.5	32.42	38.24	38.95	39.93
20	27.86	29.4	30.39	34.69	34.64	36.98
30	23.87	28.26	27.56	32.16	32.44	33.66
40	20.22	26.5	24.59	30.59	30.56	32.42
50	17.06	25.39	21.23	28.54	28.66	30.25
60	13.21	22.5	14.22	26.93	26.83	28.43
70	11.93	17.25	11.84	24.64	24.48	26.3
80	10.68	12.31	10.02	22.32	22.45	23.14
90	8.85	9.96	8.57	19.15	19.65	20.5

TABLE 2

Comparison of IEF values of various algorithms for Video Frame one shown in Fig.6.

Noise %	Image Enhancement Factor (IEF)					
	MF	AMF	PSMF	DBA	DBUTMF	MDBUTMF
10	12.58	25.25	173.52	395.69	425.29	650.99
20	30.27	39.79	209.35	360.98	379.56	570.56
30	32.09	45.68	195.96	325.79	345.63	594.26
40	25.37	43.69	153.33	268.49	281.22	436.25
50	15.79	39.41	65.89	212.87	222.22	349.24
60	9.31	27.31	8.31	193.86	179.29	283.46
70	5.21	9.79	5.38	130.44	131.55	173.56
80	5.12	5.62	4.26	70.46	76.52	104.24
90	2.32	2.31	2.35	35.89	35.35	36.67

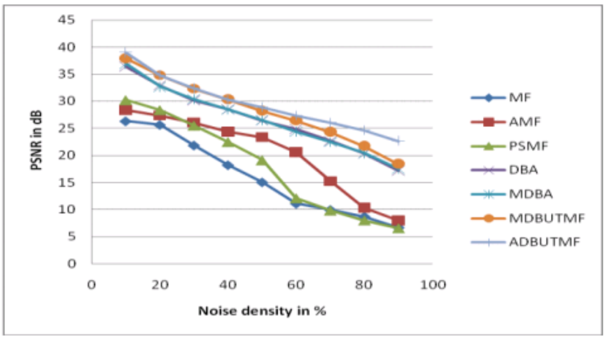

Fig.3 PSNR of various algorithms Vs Noise Density

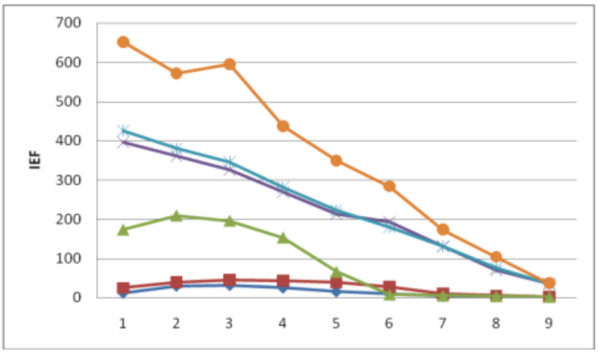

Fig.4 IEF of various algorithms Vs Noise Density

From the figures 3 and 4, it is possible to observe that the quality of the restored Video frame using proposed algorithm is better than the quality of the restored Video frame using existing algorithms.

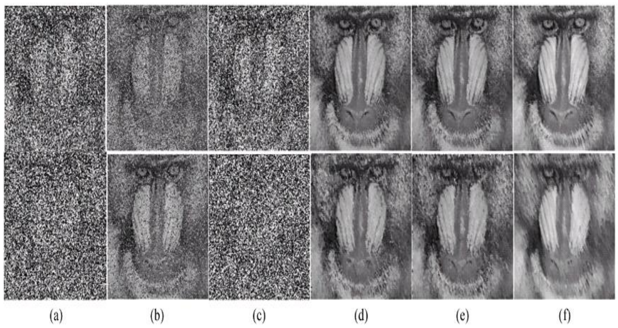

Fig.5.Results of different algorithms for Baboon Video frame. (a) Output of MF. (b) Output of AMF. (c) Output of PSMF. (d) Output of DBA. (e) Output of MDBA.(f) Output of MDBUTMF. Row 1 and Row 2 show processed results of various algorithms for video frame corrupted by 80% and 90% noise densities, respectively.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

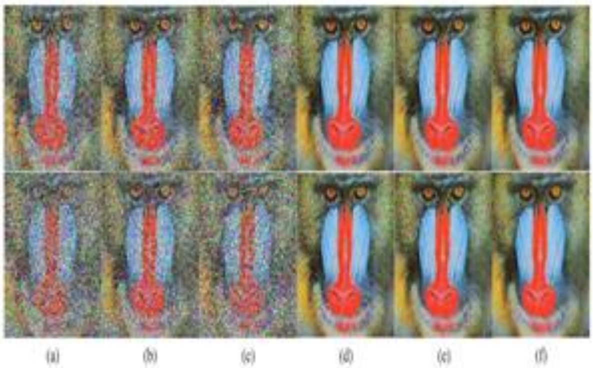

Fig.6. Results of different algorithms for color Baboon Video frame. (a) Output of MF. (b) Output of AMF. (c) Output of PSMF. (d) Output of DBA. (e) Output of MDBA. (f) Output of MDBUTMF. Rows 1 and 2 show processed results of various algorithms for color video frame corrupted by 70% and 80% noise densities, respectively.

IX. Conclusion

In this paper, a new algorithm MDBUTMF is proposed to remove noise in gray and color videos which gives better performance in comparison with MF, AMF and other existing noise removal algorithms in terms of PSNR and IEF. These filters are known for their capability to remove salt and pepper noise and preserve 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. The performance of the algorithm has been tested at low, medium and high noise densities on both gray-scale and color Videos. 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 Videos at high noise densities.

REFERENCES

1. 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.
2. 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.
3. P. E. Ng and K. K. Ma, “A switching median filter with boundary discriminative noise detection for extremely corrupted images,” IEEETrans. Image Process., vol. 15, no. 6, pp. 1506-1516, Jun. 2006.
4. K. S. Srinivasan and D. Ebenezer, “A new fast and efficient decisionbased algorithm for removal of high density impulse noise,” IEEE Signal Process. Lett., vol. 14, no. 3, pp. 189-192, Mar. 2007.
5. V. Jayaraj and D. Ebenezer, “A new switching-based median filteringscheme and algorithm for removal ofhigh-density salt and pepper noise in image,” EURASIP J. Adv. Signal Process., 2010.
6. 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.
7. E.Abreu, M.Lightstone, S.K.Mitra and K.Arakawa, 1996. “A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Videos,” IEEE Trans. Image Processing,
8. D.R.K.Brownrigg, 1984. “The Weighted Median Filter”, Commun. Assoc. Comput. Machin
9. John W.Leis “Digital signal processing using MATLAB”

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)