

Development and Verification of Novel Black and White Area Preserving Salt and Pepper Noise Removal Image Processing Design

Abhishek Jain and Richa Gupta

Jaypee Institute of Information Technology, Department of ECE, 201309 Noida, India

Abstract: In semiconductor industry, image processing algorithms are developed and evaluated using software models ('C'/MATLAB) before actual implementation of RTL. After the evaluation of the algorithm, software models are used as a golden reference model for the RTL development of the image processing algorithms. In this study, we are describing the novel black and white area preserving salt and pepper noise removal algorithm and its RTL implementation using Verilog language. SystemVerilog UVM based verification environment of salt and pepper noise removal RTL design is developed. Quality of different image denoising algorithms is quantitatively measured by different parameters, namely Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Multi Scale Structural Similarity Index (MS-SSIM). Time complexity is measured by Big O notation and stopwatch timer. The main motivation behind this work is to propose best efficient black and white area preserving salt and pepper noise removal algorithm, its RTL implementation and development of efficient functional verification framework of salt and pepper noise removal RTL design. The proposed algorithm gives better MSE, PSNR, SSIM and MS-SSIM results. Time taken by proposed algorithm is not less than other existing algorithms but it is acceptable as Big O notation is same for other algorithms.

Key words: Salt and pepper noise, grayscale images, functional verification, SystemVerilog, Universal Verification Methodology (UVM), Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM)

INTRODUCTION

Salt and pepper noise is especially originated in images throughout signal addition stage. It's caused from presence of dirt particles in image acquisition source or because of hot faulty parts. It is also caused because of bit errors in transmission (Astola and Kuosmanen, 1997). Salt and pepper noise corrupted pixels takes either minimum or maximum gray level. As images, may be corrupted by different types of noises like impulse, additive or signal dependent noise, thus, noise removal is a tough task.

Universal Verification Methodology (UVM) is the first standard, interoperable, open and proven verification re-use methodology. It provides interoperability of simulator, verification IP and high-level language within and across companies. It is scalable from IP level to system-level verification. It gives automation feature which is important for reuse of verification environment. It is developed, documented and maintained by multiple vendors and it does not bind users to use single vendor solution like existing class libraries. UVM is a frequent technique for the functional verification of hardware

designs, mainly the usage of simulation. The hardware design which is to be tested may be described using VHDL, Verilog, SystemVerilog or SystemC at any suitable abstraction level. This can be register transfer level, behavioral or gate level. Assertion-based verification and hardware emulation or acceleration additionally can be used in conjunction with UVM (Rosenberg and Meade, 2013; Timisescu and Simm, 2015).

A SystemVerilog UVM test bench consists of reusable verification elements. A verification element is a configurable, ready to use, encapsulated, verification environment for a design module, an interface protocol or a full System on Chip (SoC) (Holloway, 2012; Iman, 2008). Each verification element follows a standard architecture for test data generation, data and protocol checking and obtaining functional coverage information for a specific DUT or protocol. The verification environment is applied to the designs to verify implementation of the protocol or design architecture (Bergeron, 2013; Kitchen and Kuehlmann, 2007).

In our case, development and evaluation of novel black and white area preserving salt and pepper removal algorithm is done using MATLAB Software Model. After

development of the software model of salt and pepper removal algorithm, RTL design of proposed algorithm is done using Verilog language. MATLAB Model is used as a golden reference model for the development of RTL. Then, MATLAB reference model is used in SystemVerilog UVM based functional verification framework of proposed algorithm RTL for its bit true verification. The proposed algorithm gives better MSE, PSNR, SSIM and MS-SSIM results.

This study is organized in different studies. It presents the salt and pepper noise removal algorithms and limitations of early algorithms for preserving the black and white area of the image. We are describing salt and pepper noise removal algorithm investigation, novel algorithm development, its RTL implementation and functional verification using UVM based verification environment.

Literature review: Linear filtering can be used to remove additive noise in an image but it blurs edges and thus, fails to reduce impulse noise. This disadvantage results in the employment of non-linear filtering in salt and pepper noise reduction (Jain and Gupta, 2017). Because of its smart noise removal power and effective computation options, the median filter is the foremost ordinarily used nonlinear filters for removing salt and pepper noise. However, when the noise density value is above 0.5, some details and therefore the edges of the original image are spread by the median filter. Different kinds of median filter have been used for removing the salt and pepper noise, e.g., the adaptive median filter (Hwang and Haddad, 1995), the Switching Median Filter (SMF) (Zhang and Karim, 2002; Ng and Ma, 2006), Decision Based Algorithm (DBA) (Srinivasan and Ebenezer, 2007), Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) (Esakkirajan *et al.*, 2011) and High Density Bilateral Filter (HDBF) (Veerakumar *et al.*, 2012).

Adaptive Median Filter (AMF) (Hwang and Haddad, 1995) established in year 1995 is very effective to reduce salt and pepper noise at low noise densities. But at high noise densities (where noise density value >0.5), the kernel window size need to increase which may result in blurring the image. To avoid this problem, Decision Based Algorithm (DBA) (Srinivasan and Ebenezer, 2007) was proposed where a 3×3 window is used to reduce salt and pepper noise. If the processing central pixel value is either 0 or 255 then, it is processed or else it is left unmodified. At high noise density values, the calculated median value can be 0 or 255 which are noisy pixel values and cannot be used for replacement. Thus, left neighboring pixel is used for replacement of the noisy central pixel. But repeated replacement of noisy central pixel by neighboring pixel

causes streaking effect in the output denoised image. To overcome the above problem, Decision Based Unsymmetric Trimmed Median Filter (DBUTMF) was 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 density values (0.8-0.9). To avoid this drawback, Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) (Esakkirajan *et al.*, 2011) algorithm was proposed in 2011. Edge-Preserving Algorithm (EPA) (Sree *et al.*, 2012) proposed in 2012 was using a directional correlation dependent noise filtering technique to remove salt and pepper noise.

A Bilateral filter is an effective filter for removing salt and pepper noise that uses weights based upon spatial and radiometric similarity (Veerakumar *et al.*, 2012). The bilateral filter has good results in removing salt and pepper noise while preserving image details. This method is local, simple and non-iterative. The bilateral filter is based upon the "detect and replace" methodology. Noise detection is based on the absolute difference between a current pixel value and the reference median. The reference median is obtained from Sorted Quadrant Median Vector (SQMV).

All the currently available algorithms (Hwang and Haddad, 1995; Zhang and Karim, 2002; Ng and Ma, 2006; Srinivasan and Ebenezer, 2007; Esakkirajan *et al.*, 2011; Sree *et al.*, 2012; Wang and Zhang, 1999; Veerakumar *et al.*, 2012; Jayaraj and Ebenezer, 2010; Aiswarya *et al.*, 2010; Chen and Lien, 2008; Nair *et al.*, 2008) for removal of salt and pepper noise are not efficiently preserving the black and white area of the image. In this study, we are proposing the novel algorithm which will efficiently remove the salt and pepper noise while preserving the black and white area of the image. Proposed algorithm gives better MSE, PSNR, SSIM and MS-SSIM results as compared to previous algorithms. RTL implementation of the proposed algorithm and its functional verification using SystemVerilog UVM based verification environment is also described.

MATERIALS AND METHODS

The proposed black and white area preserving algorithm for removal of salt and pepper noise processes the corrupted images by first detecting the salt and pepper noise. Proposed algorithm also considers the fact that black (0) and white (255) pixels can also be part of the uncorrupted image. Thus, proposed salt and pepper noise removal algorithm also detects the black and white area of

the image while processing the corrupted image. This algorithm is based on the logic that for different noise density values (up to 0.9), all the pixel's values in the selected window of image can't become 0's and 255's and if all the pixels in the selected window are 0's and 255's then central pixel in selected window can be actual image pixel not salt and pepper noise pixel. The processing pixel is checked whether it is noisy or noise free. That is, if the processing pixel lies between minimum and maximum gray level values then it is noise free pixel, it is left unmodified. If the processing pixel takes the minimum or maximum gray level then it can be noisy pixel or actual image pixel which is processed by the proposed algorithm. The steps of the proposed algorithm are described as:

Algorithm 1; Proposed algorithm:

For iteration 1:N

Step 1: Select 2-D window of size 3x3 (if Noise Density <0.5) or 5x5 window (ND >= 0.5) with center element P_{ij} as a processing pixel

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

Step 3: If $P_{ij} = 0$ or $P_{ij} = 255$ then P_{ij} can be corrupted pixel or actual image pixel and 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 center processing pixel can be actual image pixel and replace center pixel value by either 0 or 255 depending upon higher density of pixel value (0 or 255) present in the selected window

Case ii): If the selected window does not contain all elements as 0's and 255's, then, delete elements having 255's and 0's values and find the mean value of the remaining elements. Then finally replace center pixel value with the mean value

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

Step 5: Calculate PSNR value of current output denoised image if iteration >1

i) If PSNR value of current denoised output image is greater than PSNR value of previous denoised output image. Then, use current output denoised image for further processing in next iteration

ii) If PSNR value of current denoised image is less than or equal to PSNR value of previous denoised image. Then, use previous output denoised image as final output image

else
Go to next iteration
end
end

The pictorial representation of each case of the proposed algorithm is shown in Fig. 1.

RTL implementation and verification framework of proposed algorithm: RTL Model of proposed salt and pepper noise removal algorithm (SNPNR) is implemented using Verilog language. SystemVerilog Universal Verification Methodology (UVM) based environment is developed for verification of proposed salt and pepper noise removal RTL design. SystemVerilog UVM verification environment is integrated with software model of proposed salt and pepper noise removal algorithm (written in MATLAB Version 2013a) using DPI

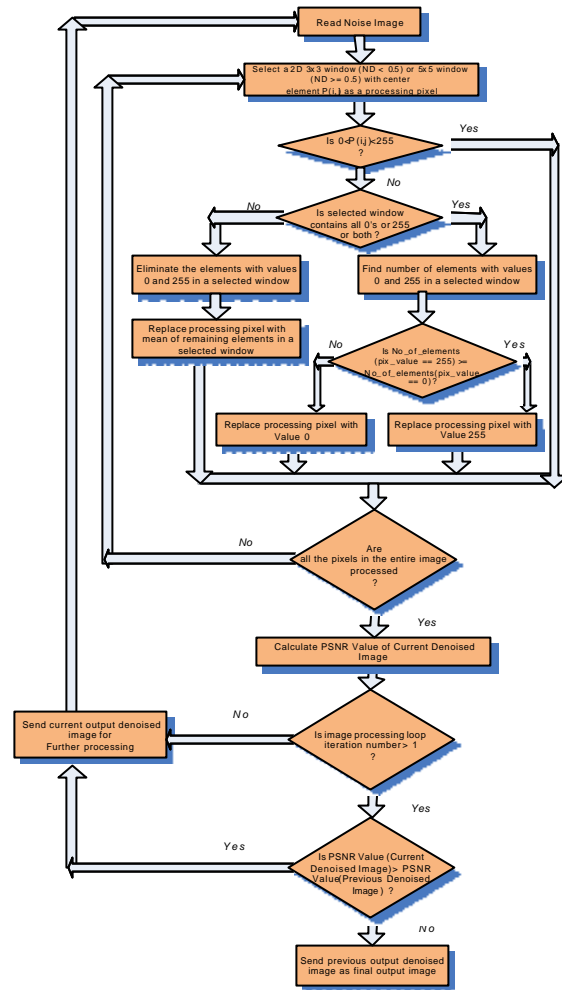


Fig. 1: Flow chart of proposed algorithm

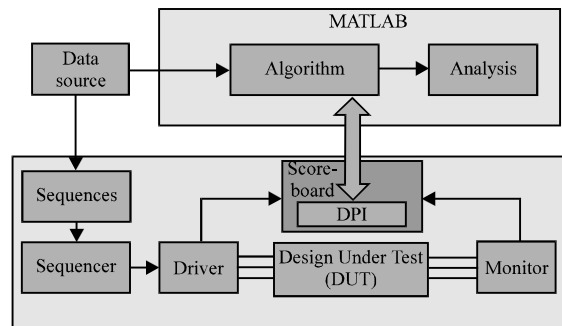


Fig. 2: UVM verification environment used for proposed algorithm

communications. Figure 2 shows the integration of SV UVM based verification environment with software model of proposed algorithm. Test data used in the verification environment is both random data as well as directed data (Jain *et al.*, 2014; Marconi *et al.*, 2015; Liang *et al.*,

Table 1: Testing features and their description

Testing feature (s)	Feature (s) description
Register tests	For all registers, it is verified that: register values can be read; default values are correct; attempted write does change read back register contents. Write/Read registers is done in terms of bytes, words and dwords and different endiannesses
IP enable/disable	SNPNR is disabled to verify that no salt and pepper noise removal is done by SNPNR RTL design. SNPNR is enabled to verify that Salt and Pepper Noise Removal is done by SNPNR IP in conformity with that of the reference software model
Soft reset	SNPNR is verified under soft reset condition; 0: normal-normal algorithmic function; Reset-Resets algorithm block
Inter line gap	SNPNR supports minimum interline gap of 5 clock periods. This value is randomized in the base test sequence. In this range, there are some selected values which is constrained for testing purpose. The range for inter_line_gap in SNPNR is 5-500
Multiple frame	SNPNR supports multiple frame processing. Multiple frames are given as the stimulus to the IP for functional checking. The number of frames which are provided as stimulus is randomized, for SNPNR the required values are achieved by constraining it
Image resolution	The stimulus image is generated through randomization in the base sequence. These values are constrained in the test case for generating the specific scenarios. Also, Random testing is performed to generate complex/special cases as well as to enhance coverage

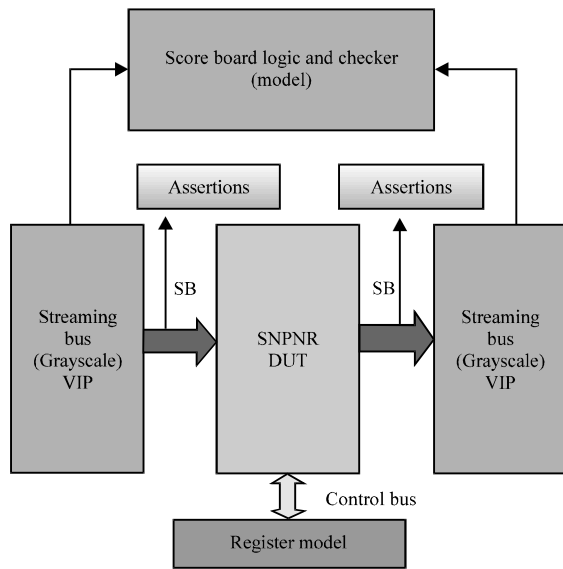


Fig. 3: Verification environment architecture of salt and pepper noise removal design

2014; Kim *et al.*, 2009). Around 100 test images corrupted with salt and pepper noise are used as input stimulus to verify the efficiency of the proposed algorithm. To regressively verify the proposed algorithm for all possible values of the input pixel values, random test images are generated in the verification environment and driven to proposed design (Kannavara, 2013; Glasser, 2009). It is also verified in the verification environment that proposed algorithm is not corrupting the input images where salt and pepper noise is not present.

Verification environment architecture of salt and pepper noise removal RTL design is described in Fig. 3. To verify the functionality of salt and pepper noise removal algorithmic IP, various kinds of images (achieved through randomization) are given as stimulus with different configuration, specific to the IP to verify the functionality.

Integrated testing of the block is done for functional coverage, through deterministic simulation, directed random simulation and protocol checking (Jain and Gupta, 2015; Swan, 2001; Iman, 2008). Table 1 lists out the testing features and their description.

Checking mechanism: The following checking mechanisms is used within the verification environment: bit-true checking methodology is used for data integrity checking (Jain *et al.*, 2013; Jain and Gupta, 2016). For each output image, corresponding expected data is found by driving the same input image to the MATLAB Model, compare the output image from MATLAB Model with RTL. This suite of tests is using self-checking mechanism.

Monitor checking methodology is used for temporal and access checking (Anonymous, 2014; Swan, 2001). At respective clock edges, items such as register reset value, control bus access and overflow signals are checked.

Functional checklist: The following list summarizes the main functions of the SNPNR block:

- Protocols (Signal assertion/de-assertion timings)
 - Control bus (Communicate with the UVM_REG Register Model (Rosenberg, 2012))
 - Data transfer protocol (Streaming bus)
- Registers access
- IP enable/Disable
- Soft reset-SNPNR is verified under soft reset condition
 - 0: Normal-Normal algorithmic function
 - 1: Reset-Resets algorithm block
- Data integrity under different scenarios say image sizes, inter line gaps, inter frame gap, multiple frame

Randomization: All the parameters with respect to the image, e.g., img_h_size, img_v_size, inter_line_gap, inter_frame_gap are randomized in the base sequence. The specific scenarios are achieved by constraining these values specific to required feature.

Directed test cases are created for achieving corner cases and negative scenarios. As 100% functional coverage is not achieved through randomization, directed test cases are also written to increase the coverage count.

Functional coverage: Table 2 shows the functional coverage bins and their possible values.

Table 2: Functional coverage bins and their possible values

Cover bins	Values
snprn_enable	0, 1
snprn_soft_reset	0, 1
user_image	100 test images (Lena, Dog, Taj, Baboon, Medical Images etc.)
inter_line_gap	5-500 clocks
random_data	0,1
inter_frame_gap	4-20 lines
multiple_frame	Number of frames
Image resolution	160×120, QVGA (320×240), VGA (640×480), 512×512, 1 MP etc.

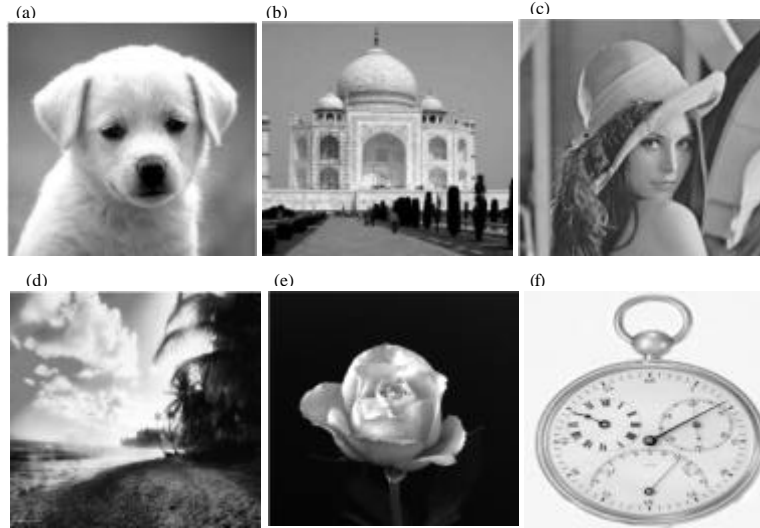


Fig. 4: Test images: a) Grayscale dog; b) Taj Mahal; c) Lena; d) Palms tree; e) Rose and f) Chronometer

RESULTS AND DISCUSSION

The performance of the proposed algorithm is tested with different grayscale test images. Some of the grayscale test images are shown in Fig. 4. Image database for test images is obtained from following links:

- http://www.imageprocessingplace.com/root_files_V3/image_databases.htm
- <http://marathon.csee.usf.edu/Mammography/Database.html>
- <http://www.fit.vutbr.cz/~vasicek/imagedb/>

The noise density value is varied from 0.1-0.9. Denoising performances of different algorithms are quantitatively measured by the Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Structural Similarity (SSIM) Index and Multi Scale Structural Similarity Index (MS-SSIM) as defined in Eq. 1 and 2:

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

$$MSE = \frac{\sum_i \sum_j (Y(i, j) - D(i, j))^2}{M \times N} \tag{2}$$

Where:

- MSE = Mean Square Error
- M×N = Size of the image
- Y = The original image
- D = The Denoised image

The Structural Similarity (SSIM) Index quality assessment index is based on the computation of three terms, namely the luminance term, the contrast term and the structural term. The overall index is a multiplicative combination of the three terms.

In Multi Scale Structural Similarity Index (MS-SSIM), Multi Scale Method is used to include details of image at various resolutions.

Time complexity of the algorithms for removing salt and pepper noise is measured using 2 methods: running the same program 1000 times in different machines of same configuration and calculate the minimum, maximum, average and mode of run time taken by each algorithm.

Big O notation is employed to classify algorithms consistent with how their running time or area requirements grow because the input size grows.

The qualitative inspection of the proposed algorithm against the existing algorithms at various noise densities for grayscale dog image is shown in Fig. 5. In Fig. 5,

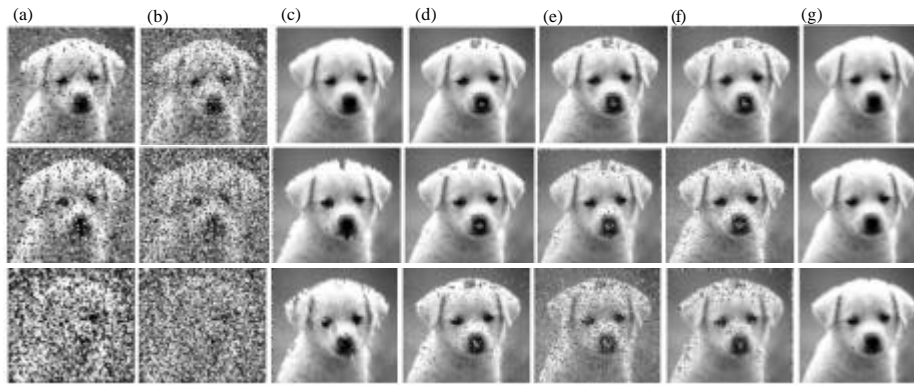


Fig. 5: Results of different algorithms for grayscale dog image: a) Output of median filter; b) Output of PSMF; c) Output of DBA; d) Output of MDBUTMF; e) Output of bilateral filter; f) Output of edge preserving filter and g) Output of proposed filter. Row 1-3 show processed results of various algorithms for grayscale dog image corrupted by 70, 80 and 90% noise densities, respectively

Table 3: Comparison of mean square error values of different algorithms for dog grayscale image at different noise densities

MSE	Noise density (%)								
	10	20	30	40	50	60	70	80	90
MF	09.86	25.78	051.90	092.48	225.61	0698.00	2239.53	5569.9	11510.3
DBA	07.52	14.12	024.79	039.29	060.00	0093.86	0162.67	00298.34	00697.33
EP	32.00	65.03	128.62	186.64	250.34	0364.04	0614.84	0922.45	01480.74
PSMF	20.79	47.37	101.90	202.67	503.98	4272.1	7128.26	11015	15349.2
UNTRIMMED	17.70	37.51	072.02	103.93	153.80	0265.37	0513.71	1123	02558.16
MDBUTMFG	10.47	20.92	036.27	054.12	075.90	0116.38	0161.49	247.08	00658.46
Bilateral	12.67	26.28	050.09	081.73	135.11	0250.71	0507.52	1141.5	02574.52
Proposed filter	06.50	12.47	018.68	025.43	037.91	0048.33	0061.22	84.065	00127.29

Table 4: Comparison of peak signal to noise ratio values of different algorithms for dog grayscale image at different noise densities

PSNR in dB	Noise density (%)								
	10	20	30	40	50	60	70	80	90
MF	38.19	34.01	30.97	28.47	24.59	19.69	14.62	10.67	07.51
DBA	39.36	36.63	34.18	32.18	30.34	28.40	26.01	23.38	19.69
EP	33.07	29.99	27.03	25.42	24.14	22.51	20.24	18.48	16.42
PSMF	34.95	31.37	28.04	25.06	21.10	11.82	09.60	07.71	06.26
UNTRIMMED	35.65	32.38	29.55	27.96	26.26	23.89	21.02	17.62	14.05
MDBUTMFG	37.92	34.92	32.53	30.79	29.32	27.47	26.04	24.20	19.94
Bilateral	37.10	33.93	31.13	29.00	26.82	24.13	21.07	17.55	14.02
Proposed filter	39.99	37.16	35.41	34.07	32.34	31.28	30.26	28.88	27.08

Table 5: Comparison of structural similarity index values of different algorithms for dog grayscale image at different noise densities

Structural similarity index	Noise density (%)								
	10	20	30	40	50	60	70	80	90
MF	0.9821	0.9646	0.9432	0.9126	0.8198	0.5938	0.2867	0.0966	0.0264
DBA	0.9834	0.9666	0.9447	0.9186	0.8861	0.8421	0.7863	0.7039	0.5870
EP	0.9659	0.9369	0.9014	0.8689	0.826	0.7814	0.7249	0.6530	0.5708
PSMF	0.9669	0.9367	0.8827	0.7855	0.6034	0.1015	0.0471	0.0221	0.0111
UNTRIMMED	0.9691	0.9497	0.926	0.9002	0.8562	0.7627	0.5926	0.3385	0.1351
MDBUTMFG	0.973	0.9535	0.9321	0.9112	0.8874	0.8585	0.8252	0.7603	0.5201
Bilateral	0.971	0.9517	0.9259	0.8958	0.844	0.7418	0.5703	0.3232	0.1302
Proposed filter	0.9841	0.9697	0.9542	0.9363	0.9062	0.8813	0.8529	0.8129	0.7537

the first column represents the processed image using median filter at 70, 80 and 90% noise densities. Subsequent columns represent the processed images for PSMF, decision based algorithm, MDBUTMF, bilateral filter, edge preserving filter and proposed filter. From visual analysis, proposed algorithm gives better results than other existing algorithms.

The MSE, PSNR, SSIM and MS-SSIM values of the proposed algorithm are compared against the existing algorithms by varying the noise density from 10-90% and is shown in Table 3-6. Plots of MSE, PSNR, SSIM and MS-SSIM against noise densities for grayscale dog image are shown in Fig. 6.

Table 6: Comparison of multi scale structural similarity index values of different algorithms for dog grayscale image at different noise densities

Multi scale structural similarity index	Noise density (%)								
	10	20	30	40	50	60	70	80	90
MF	0.9976	0.9949	0.9912	0.9839	0.9384	0.7919	0.53	0.3118	0.1333
DBA	0.9978	0.9951	0.9912	0.9858	0.9782	0.964	0.9422	0.8938	0.7817
EP	0.9882	0.9802	0.9646	0.9539	0.9385	0.9183	0.8855	0.8437	0.7534
PSMF	0.9953	0.99	0.9761	0.9461	0.8518	0.3955	0.2725	0.1767	0.0913
UNTRIMMED	0.99	0.9836	0.9746	0.966	0.951	0.917	0.8422	0.6867	0.4353
MDBUTMFG	0.993	0.9875	0.9804	0.9745	0.9666	0.9549	0.9437	0.9188	0.771
Bilateral	0.9903	0.9838	0.9741	0.964	0.9465	0.91	0.8347	0.6814	0.4386
Proposed filter	0.9979	0.9957	0.9932	0.9903	0.9852	0.9798	0.9736	0.9625	0.9395

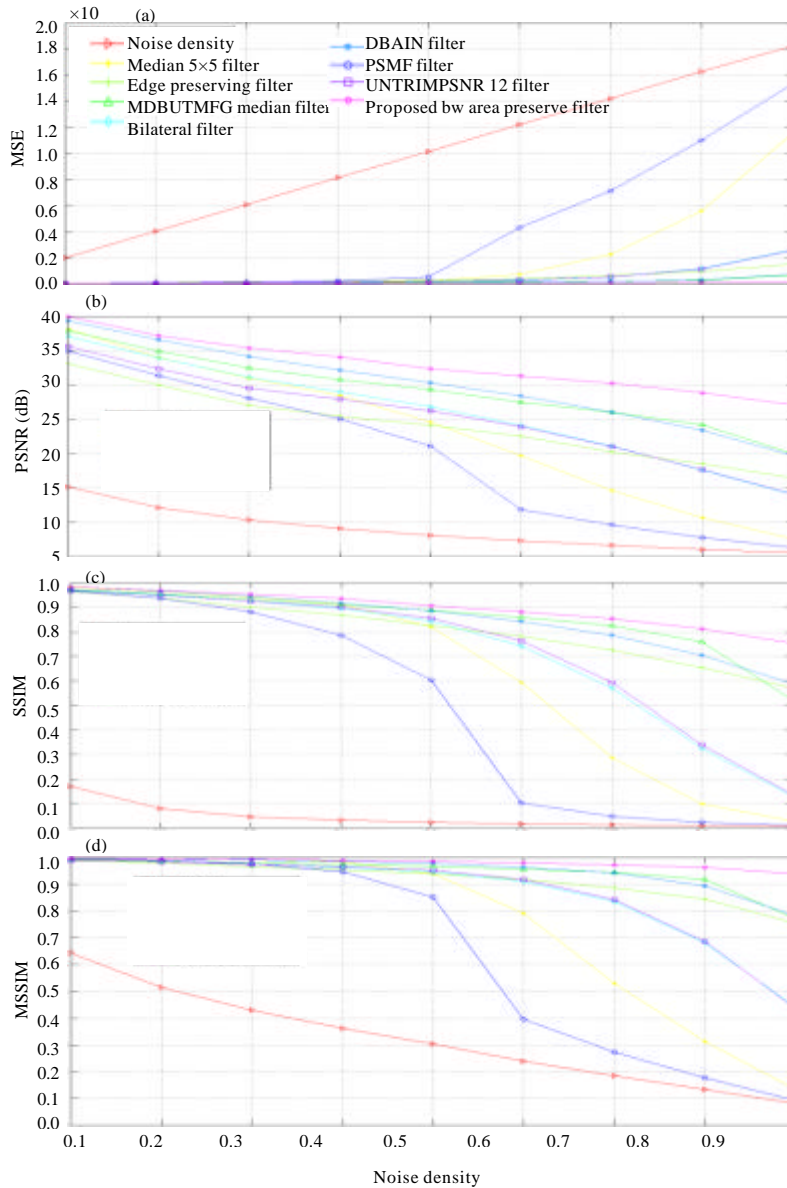


Fig. 6: a-d) Comparison of restoration results in terms of MSE, PSNR (dB), SSIM and MSSIM for grayscale dog image corrupted by various densities of salt and pepper noise

Cumulative ranks for 100 test images at various noise densities is obtained for Mean Square Error (MSE), Peak

Signal to Noise Ratio (PSNR), Structural Similarity (SSIM) Index and Multi Scale Structural Similarity Index

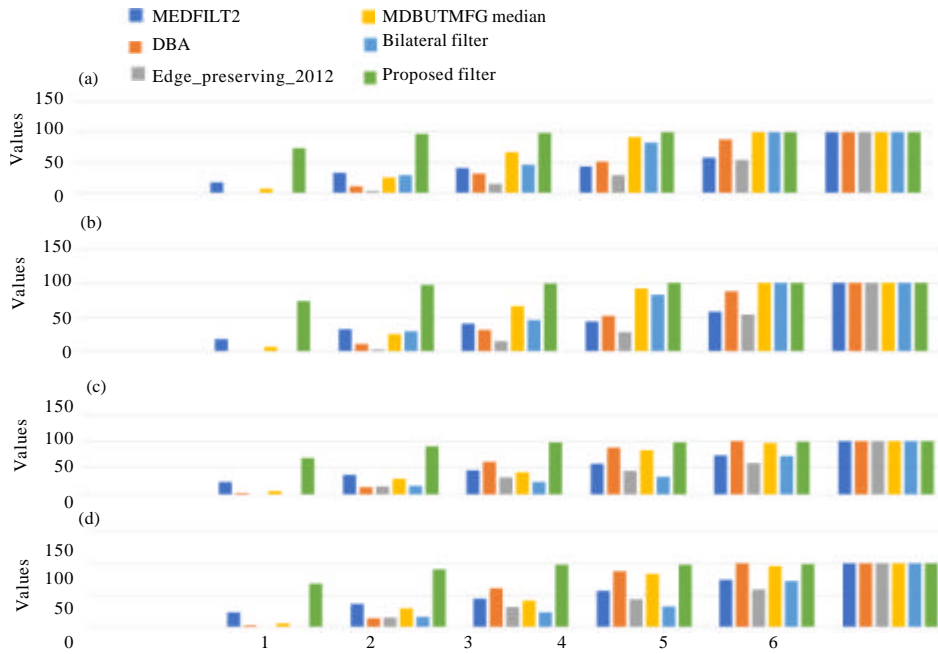


Fig. 7: Cumulative ranks for 100 test images (noise density = 0.4): a) Mean square error cumulative rank; b) PSNR cumulative rank; c) Structural similarity (SSIM) index cumulative rank and d) Multi Scale Structural Similarity Index (MS-SSIM) cumulative rank

(MS-SSIM) parameters. It can be easily seen that proposed algorithm outperforms and provides the best fit. For Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) parameters, proposed algorithm ranks first in 74% of the cases (for 40% noise density) while median filter explains about 18% cases and modified decision based unsymmetric trimmed median filter explains about 7% cases. Remaining 1% of the cases are explained by decision based algorithm. It is observed that Edge-Preserving Algorithm (EPA) is not ranking 1 for any of test images. Lower cumulative rank 2 is also given in Fig. 7 where proposed algorithm is again found to outperform with 97% success. It is noted that the success of edge-preserving algorithm for cumulative rank 2 is only 3% while median filter is able to achieve 41% success.

For Structural Similarity (SSIM) Index parameter, proposed algorithm ranks first in 68% of the cases (for 40% noise density) while median filter explains about 23% cases and Modified Decision Based Unsymmetric Trimmed Median Filter (MDBUTMF) explains about 6% cases. Remaining 3% of the cases are explained by decision based algorithm.

For Multi Scale Structural Similarity (SSIM) Index parameter, proposed algorithm ranks first in 73% of the cases (for 40% noise density) while median filter explains

about 20% cases and Decision Based Algorithm (DBA) explains about 6% cases. Remaining 3% of the cases are explained by modified decision based unsymmetric trimmed median filter.

For Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) parameters (for 70% noise density) as shown in Fig. 8, proposed algorithm ranks first in 99% of the cases. Modified decision based unsymmetric trimmed median filter explains remaining 1% cases. Lower cumulative rank 2 is also given in Fig. 8 where modified decision based unsymmetric trimmed median filter algorithm explains about 68% cases.

For Structural Similarity (SSIM) index parameter, proposed algorithm ranks first in 75% of the cases (for 70% noise density) while modified decision based unsymmetric trimmed median filter explains about 12% cases. Both decision based algorithm and Edge-Preserving Algorithm (EPA) explains 6% cases each. Remaining 1% of the cases are explained by bilateral filter.

For Multi Scale Structural Similarity (SSIM) index parameter, proposed algorithm ranks first in 88% of the cases while modified decision based unsymmetric trimmed median filter explains about 6% cases. Decision based algorithm explains 4% cases. Remaining 2% of the cases are explained by bilateral filter.

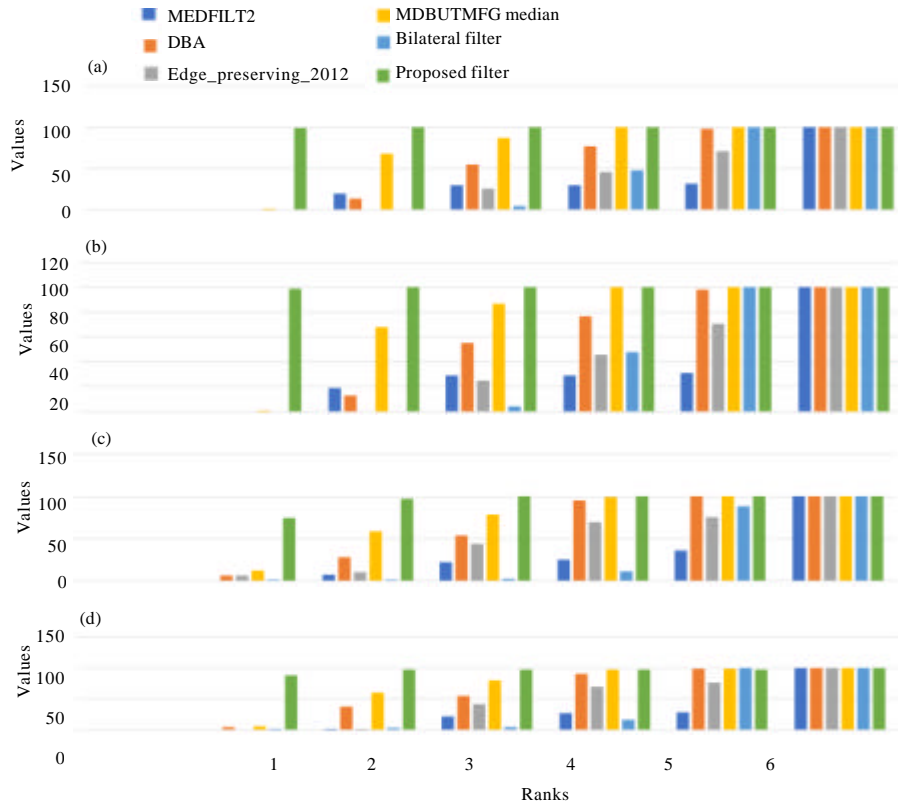


Fig. 8: Cumulative ranks for 100 test images (noise density = 0.7): a) Mean square error cumulative rank; b) PSNR cumulative rank; c) Structural similarity (SSIM) index cumulative rank and d) Multi Scale Structural Similarity Index (MS-SSIM) cumulative rank

Table 7: Computation time of the algorithm for 332×100 grayscale dog image (Noise density = 0.3)

Noise removal technique	Minimum time (sec) (Average of 4 machines)	Average time taken (sec) (Average of 4 machines)	Maximum time taken (sec) (Average of 4 machines)	Mode (sec)
MEDFILT2	0.028468	0.028892	0.037659	0.028796
DBA	0.582714	0.595845	0.621195	0.589606
Edge_preserving_2012	1.453055	1.479274	1.619176	1.483474
MDBUTMFG median	1.270105	1.285168	1.359285	1.278155
Bilateral filter	1.917159	1.946156	2.070852	1.962166
Proposed white and black area preserve filter	1.375545	1.409753	1.568335	1.385959

Table 8: Computation time of the algorithm for 332×300 grayscale dog image (Noise density = 0.7)

Noise removal technique	Minimum time (sec) (Average of 4 machines)	Average time taken (sec) (Average of 4 machines)	Maximum time taken (sec) (Average of 4 machines)	Mode (sec)
MEDFILT2	0.02844	0.028867	0.039841	0.028788
DBA	0.58618	0.59632	0.635929	0.599642
Edge_preserving_2012	1.75170	1.775987	1.866814	1.761829
MDBUTMFG median	2.13261	2.156193	2.283046	2.1327361
Bilateral filter	3.54403	3.615309	3.772333	3.6386627
Proposed white and black area preserve filter	3.10790	3.152506	3.343715	3.1277058

From Table 3-6, 9-16 and MSE, PSNR, SSIM and MS-SSIM plots in Fig. 6-8, it is observed that the performance of the proposed algorithm is better than the existing algorithms at both low and high noise densities.

From Table 7, 8, 17 and 18, it is observed that time taken by proposed algorithm is not less than other existing algorithms but it is acceptable as Big O notation is same for other algorithms.

Table 9: Cumulative rank for 100 test images Mean Square Error (MSE) (Noise density = 0.4)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	18	33	41	44	58	100
DBA	1	11	32	52	88	100
Edge_preserving_2012	0	3	15	29	54	100
MDBUTMFG median	7	26	67	92	100	100
Bilateral filter	0	30	46	83	100	100
Proposed filter	74	97	99	100	100	100

Computation time of all the algorithms for 332×300 grayscale dog image with following machine configuration is described in Table 7 and 8:

- Processor: Intel (R) Core (TM) i5-6500 CPU @ 3.20 and 3.19 GHz
- Installed RAM -8 GB
- System type: 64 bit Windows 10

Table 10: Cumulative rank for 100 test images (Peak Signal to Noise Ratio (PSNR), noise density = 0.4)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	18	33	41	44	58	100
DBA	1	11	32	52	88	100
Edge_preserving_2012	0	3	15	29	54	100
MDBUTMFG median	7	26	67	92	100	100
Bilateral filter	0	30	46	83	100	100
Proposed filter	74	97	99	100	100	100

Table 11: Cumulative rank for 100 test images (Structural Similarity (SSIM) Index, noise density = 0.4)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	23	37	45	57	74	100
DBA	3	14	61	88	100	100
Edge_preserving_2012	0	15	32	44	59	100
MDBUTMFG median	6	29	42	83	96	100
Bilateral filter	0	16	23	33	72	100
Proposed filter	68	91	98	98	99	100

Table 15: Cumulative rank for 100 test images (Structural Similarity (SSIM) Index, noise density = 0.7)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	0	7	22	25	36	100
DBA	6	28	54	95	100	100
Edge_preserving_2012	6	10	44	70	76	100
MDBUTMFG median	12	59	78	99	100	100
Bilateral filter	1	1	2	11	88	100
Proposed filter	75	97	100	100	100	100

Table 16: Cumulative rank for 100 test images (Multi Scale Structural Similarity Index (MS-SSIM), noise density = 0.7)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	0	1	22	27	28	100
DBA	4	37	55	91	99	100
Edge_preserving_2012	0	1	41	69	76	100
MDBUTMFG median	6	60	80	98	99	100
Bilateral filter	2	3	4	17	100	100
Proposed filter	88	98	98	98	98	100

Table 17: Complexity of the algorithm in Big-O notation (Noise density = 0.2)

Image resolution (N×M)	Total pixels	MEDFILT2 runtime (sec)	DBA filter runtime (sec)	Edge preserving filter runtime (sec)	MDBUTMFG median filter runtime (sec)	Bilateral filter runtime (sec)	Proposed filter runtime (sec)
160×120	19200	0.0135947	0.1476551	0.5538744	0.3790032	0.759508	0.3041478
320×240	76800	0.0552374	0.6838336	1.9069532	1.5438976	2.189812	1.3582229
640×480	307200	0.1888228	2.4646702	6.9420062	5.4674461	7.416471	6.0623926
1280×960	1228800	0.710327	10.135813	26.473837	21.174948	26.46712	19.708841
Big-O Notation		O(N ²)	O(N ²)	O(N ²)	O(N ²)	O(N ²)	O(N ²)

- Number of machines-4
- Number of iterations in each machine-1000

Table 12: Cumulative rank for 100 test images (Multi Scale Structural Similarity Index (MS-SSIM), noise density = 0.4)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	20	35	47	56	67	100
DBA	4	20	56	73	99	100
Edge_preserving_2012	0	14	28	40	50	100
MDBUTMFG median	3	19	45	90	93	100
Bilateral filter	0	16	24	38	87	100
Proposed filter	73	95	98	100	100	100

Table 13: Cumulative rank for 100 test images (Mean Square Error (MSE), noise density = 0.7)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	0	19	29	29	31	100
DBA	0	13	55	77	98	100
Edge_preserving_2012	0	0	25	46	71	100
MDBUTMFG median	1	68	87	100	100	100
Bilateral filter	0	0	4	48	100	100
Proposed filter	99	100	100	100	100	100

Table 14: Cumulative rank for 100 test images (Peak Signal to Noise Ratio (PSNR), noise density = 0.7)

Noise removal technique	Rank 1	Rank 2	Rank 3	Rank 4	Rank 5	Rank 6
MEDFILT2	0	19	29	29	31	100
DBA	0	13	55	77	98	100
Edge_preserving_2012	0	0	25	46	71	100
MDBUTMFG median	1	68	87	100	100	100
Bilateral filter	0	0	4	48	100	100
Proposed filter	99	100	100	100	100	100

Table 18: Complexity of the algorithm in Big-O notation (Noise density = 0.7)

Image resolution (N×M)	Total pixels	MEDFLT2 runtime (sec)	DBA filter runtime (sec)	Edge preserving filter runtime (sec)	MDBUTMFG median filter runtime (sec)	Bilateral filter runtime (sec)	Proposed filter Runtime (Sec)
160×120	19200	0.014974	0.178637	0.672022	0.894731	1.774209	1.181291
320×240	76800	0.053674	0.660299	2.268147	3.203027	5.176097	3.627699
640×480	307200	0.200348	2.357679	8.802961	12.44949	18.27525	14.00019
1280×960	1228800	0.786424	10.16646	40.89332	55.71473	84.53971	66.49908
Big-O Notation		O(N ²)	O(N ²)	O(N ²)	O(N ²)	O(N ²)	O(N ²)

CONCLUSION

This study presented the work done in developing salt and pepper noise removal proposed algorithm which gives better MSE, PSNR, SSIM and MS-SSIM results as compared to existing algorithms for both low and high noise densities. SystemVerilog Universal Verification Methodology (UVM) based environment is developed and integrated with proposed salt and pepper noise removal algorithm (written in MATLAB) using DPI communications. RTL design of proposed salt and pepper noise removal algorithm is developed and verified using SystemVerilog UVM based verification environment. Time taken by proposed algorithm is comparable to other existing algorithms. Bit-true checking methodology is used for data integrity checking. All the test cases are passing in the regression and 100% functional coverage is achieved. This study is very good reference for modeling efficient salt and pepper noise removal algorithm and for applying the advanced techniques of verification and automation for development of verification environment and functional verification of salt and pepper noise removal RTL designs.

ACKNOWLEDGEMENTS

The researchers would like to thank Dr. Vineet Khandelwal (JIIT, Noida) and Prof. Alka Tripathi (JIIT, Noida) for their guidance and support. We would also like to thank Faculty members and peer scholars of ECE Department, Jaypee Institute of Information Technology for their support and guidance.

REFERENCES

Aiswarya, K., V. Jayaraj and D. Ebenezer, 2010. A new and efficient algorithm for the removal of high density salt and pepper noise in images and videos. Proceedings of the 2nd International Conference on Computer Modeling and Simulation, Volume 4, January 22-24, 2010, Sanya, Hainan, pp: 409-413.

Anonymous, 2014. Universal Verification Methodology (UVM) 1.2 class reference. Accellera Standards Organization, Napa, California, USA. http://www.accellera.org/images/downloads/standards/uvm/UVM_Class_Reference_Manual_1.2.pdf

Astola, J. and P. Kuosmanen, 1997. Fundamentals of Nonlinear Digital Filtering. CRC Press, Boca Raton, New York.

Bergeron, J., 2012. Writing Testbenches: Functional Verification of HDL Models. 2nd Edn., Springer Science Business Media, Berlin, Germany, ISBN:978-1-4613-5012-5, Pages: 477.

Chen, P.Y. and C.Y. Lien, 2008. An efficient edge-preserving algorithm for removal of salt-and-pepper noise. IEEE Signal Process. Lett., 15: 833 -836.

Esakkirajan, S., T. Veerakumar, A.N. Subramanyam and C.H. PremChand, 2011. Removal of high density salt and pepper noise through modified decision based unsymmetric trimmed median filter. IEEE Signal Proc. Lett., 18: 287-290.

Glasser, M., 2009. Open Verification Methodology Cookbook. Springer, New York, USA., ISBN:978-1-4419-0967-1, Pages: 235.

Holloway, S., 2012. The UVM Register Layer-introduction, experiences and recipes. Dialog Semiconductor, Reading, UK.

Hwang, H. and R.A. Haddad, 1995. Adaptive median filters: New algorithms and results. IEEE Trans. Image Proc., 4: 499-502.

Iman, S., 2008. Step-by-Step Functional Verification with system verilog and OVM. Hansen Brown Publishing, San Francisco, California, USA., ISBN-13:978-0981656212, Pages: 520.

Jain, A. and R. Gupta, 2015. Scaling the UVM_REG model towards automation and simplicity of use. Proceedings of the 28th International Conference on VLSI Design (VLSID), January 3-7, 2015, IEEE, Bangalore, India, ISBN:978-1-4799-6658-5, pp: 164-169.

Jain, A. and R. Gupta, 2016. Unified and modular modeling and functional verification framework of real-time image signal processors. VLSI Design, 2016: 1-14.

Jain, A. and R. Gupta, 2017. A novel and efficient black and white area preserving algorithm for removal of salt and pepper noise. Intl. J. Electron. Eng. Res., 9: 1055-1070.

Jain, A., D. Gupta, S. Jana and K. Kumar, 2014. Early development of UVM based verification environment of image signal processing designs using TLM reference model of RTL. Intl. J. Adv. Comput. Sci. Appl., 5: 77-82.

- Jain, A., G. Bonanno, H. Gupta and A. Goyal, 2013. Generic system verilog universal verification methodology based reusable verification environment for efficient verification of image signal processing IPs/SoCs. *Intl. J. VLSI Design Commun. Syst.*, 3: 13-25.
- Jayaraj, V. and D. Ebenezer, 2010. A new switching-based median filtering scheme and algorithm for removal of high-density salt and pepper noise in images. *EURASIP. J. Adv. Signal Process.*, 2010: 1-11.
- Kannavara, R., 2013. Towards a unified framework for pre-silicon validation. *Proceedings of the 4th International Conference on Information, Intelligence, Systems and Applications (IISA)*, July 10-12, 2013, IEEE, Piraeus, Greece, ISBN:978-1-4799-0771-7, pp: 1-7.
- Kim, Y., H.S. Kim, R. Lee and S. Kang, 2009. FPGA-based verification methodology of SoC-type CMOS image signal processor. *Proceedings of the International SOC Conference on SOCC*, September 9-11, 2009, IEEE, Belfast, UK, ISBN:978-1-4244-4940-8, pp: 231-234.
- Kitchen, N. and A. Kuehlmann, 2007. Stimulus generation for constrained random simulation. *Proceedings of the 2007 IEEE/ACM International Conference on Computer-Aided Design*, November 05-08, 2007, IEEE Press, San Jose, California, USA., ISBN:1-4244-1382-6, pp: 258-265.
- Liang, C., G. Zhong, S. Huang and B. Xia, 2014. UVM-AMS based sub-system verification of wireless power receiver SoC. *Proceedings of the 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, October 28-31, 2014, IEEE, Guilin, China, ISBN:978-1-4799-3296-2, pp: 1-3.
- Marconi, S., E. Conti, J. Christiansen and P. Placidi, 2015. Reusable System Verilog-UVM design framework with constrained stimuli modeling for high energy physics applications. *Proceedings of the International Conference Symposium on Systems Engineering (ISSE)*, September 28-30, 2015, IEEE, Rome, Italy, pp: 391-397.
- Nair, M.S., K. Revathy and R. Tatavarti, 2008. Removal of salt-and pepper noise in images: A new decision-based algorithm. *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, March 19-21, 2008, IAENG, Hong Kong, ISBN:978-988-98671-8-8, pp: 1-6.
- Ng, P.E. and K.K. Ma, 2006. A switching median filter with boundary discriminative noise detection for extremely corrupted images. *IEEE Trans. Image Process.*, 15: 1506-1516.
- Rosenberg, S. and K. Meade, 2013. *A Practical Guide to Adopting the Universal Verification Methodology (UVM)*. 2nd Edn., Cadence Design Systems, San Jose, California, USA., ISBN:978-1-300-53593-5, Pages: 319.
- Rosenberg, S., 2012. Register This! Experiences applying UVM registers. *Cadence Design System*, San Jose, California.
- Sree, S.J., S. Ashwin and S.A. Kumar, 2012. Edge preserving algorithm for impulse noise removal using FPGA. *Proceedings of the International Conference on Machine Vision and Image Processing (MVIP)*, December 14-15, 2012, IEEE, Taipei, Taiwan, ISBN:978-1-4673-2319-2, pp: 69-72.
- Srinivasan, K.S. and D. Ebenezer, 2007. A new fast and efficient decision-based algorithm for removal of high-density impulse noises. *IEEE. Signal Process. Lett.*, 14: 189-192.
- Swan, S., 2001. *An introduction to system level modeling in system C 2.0*. Cadence Design Systems, Inc., San Jose, California, USA.
- Timisescu, T. and U. Simm, 2015. Leveraging the UVM register abstraction layer for memory sub-system verification. *Proceedings of the European Conference on Design and Verification Conference and Exhibition DVCon*, October 24-25, 2015, Holiday Inn Munich City Center, Munich, Germany, pp: 1-7.
- Veerakumar, T., S. Esakkirajan and I. Vennila, 2012. High density impulse noise removal using modified switching bilateral filter. *Intl. J. Circuits Syst. Signal Process.*, 6: 189-196.
- Wang, Z. and D. Zhang, 1999. Progressive switching median filter for the removal of impulse noise from highly corrupted images. *IEEE. Trans. Circuits Syst. II Analog Digital Signal Process.*, 46: 78-80.
- Zhang, S. and M.A. Karim, 2002. A new impulse detector for switching median filters. *J. IEEE Signal Process. Lett.*, 9: 360-363.