Advanced Impulse Detection & Reduction Based on Multimodal Filter

P. Thirumurugan#1, Dr. S. Sasi Kumar*2

#PSNA College of Engineering and Technology, Dindigul, Tamil Nadu 624622, India
*RMD Engineering College, Tamil Nadu 601206, India
1thiruece@psnacet.edu.in

Abstract-Impulse noises are occurred in the images during image signal acquisition and processing from one location to another location. In this paper, the optimal detector noise filtering algorithm and its efficient hardware architecture is presented. The proposed architecture comprises of orthogonal direction pattern generation, sorter, thresholder, local binary converter, multimodal filter and pixel converter units respectively. The local binary converter unit detects and corrects the noise pixel efficiently using a simple logic circuit. The design possesses only two line memory buffers with very low computational complexity, thereby reducing the hardware cost and appropriate for several real-time applications.

Keywords— Edges, Image quality, Impulse noise reduction, Optimal detector, Power consumption.

I. INTRODUCTION

Images are often corrupted by impulse noise due to a noisy sensor or channel transmission errors. The goal of impulse noise removal is to suppress the noise while preserving the integrity of edge and detail information. A fundamental problem in image processing is to effectively remove noise from an image while keeping its features intact, but nature of problem depends upon the type of noise in the image. Basically there are two types of Impulse Noise. They are Random Value Impulse Noise and Fixed Value Impulse Noise. In Random value impulse noise the noise pixels are distributed based on Random Distribution. In Fixed value the noise pixels takes a value of 0 and 255 i.e., the minimum and maximum value in the grey scale. Digital filters are widely used in many applications in signal processing, communications, control, electrical and biomedical systems.

II. PROPOSED IMPULSE NOISE FILTERING ALGORITHM

The proposed optimal detector noise filtering (ODNF) system employs an innovative technique in finding the suitable direction which is used in detecting if the currently processed pixel is noise-corrupted or noise-free. The edge pixels can be detected in high quantity if the optimal direction of the edge is found out accurately.

A. Algorithm—step-by-step operation

The algorithm design includes the following steps:

Step 1: At the first step, the image which is to be denoised is first split into m×n pixel blocks. These blocks should not overlap with each other. In our investigation, we split the noise-corrupted image of size 36×36 into 16 numbers of 16×16 sub-block. Each sub-block is of size 16×16.

Step 2: Next, the first sub-block (of size 9×9) is divided into four orthogonal directional patterns (ODPs). Then we remove the current pixel from the selected odp.

Step 3: The obtained ODPs are sorted in ascending order and then from these sorted values, the lowest and highest vector patterns are removed.

Step 4: The vectors are found out after removing the lowest and highest elements.

Step 5: The minimum value of the standard deviations calculated for each orthogonal sorted directional patterns and its corresponding direction is considered as the optimum direction.

Step 6: The similarity factor between the current pixel under process and the pixels in the optimum direction is estimated and compared with the threshold value to check if the current pixel is noisy or original pixel. If it is found to be a noisy pixel, it is immediately omitted after detection and it is not carried over in further steps. Lastly the mean filter is applied over the noisy pixels.

III.SIMULATION RESULTS

The principle aspects in terms of edge preserving concept for high density impulse noise are tested MATLAB environment and its results are evaluated with the conventional noise removal algorithms. To verify the characteristics and the quality of denoised images of the modified denoising algorithm, a variety of simulations are carried out on the two well-known test images: Cameraman and Lena. For the test image, the corrupted versions of it are generated in MATLAB environment with impulse noise at various high level noise densities 40%, 50% and 55%. Then we employ the proposed algorithm to detect impulse noise and restore the corrupted image. Fig. 1 shows the simulation results achieved through MATLAB. The metrics used for comparison are Mean square error (MSE), Peak signal to noise ratio (PSNR), Normalized Absolute Error (NAE), Normalized Cross Correlation (NCC) and are defined as follows:

$$\mathrm{MSE} = \frac{1}{MN}\sum_{j=1}^{M}\sum_{k=1}^{N}\left(x_{j,k} - x'_{j,k}\right)^2 \tag{1}$$

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

$$NAE = \frac{\sum_{j=1}^{M}\sum_{k=1}^{N}\left|x_{j,k} - x'_{j,k}\right|}{\sum_{j=1}^{M}\sum_{k=1}^{N}\left|x_{j,k}\right|} \tag{3}$$

$$NCC = \frac{\sum_{j=1}^{M}\sum_{k=1}^{N} x_{j,k} \bullet x'_{j,k}}{\sum_{j=1}^{M}\sum_{k=1}^{N} x^2_{j,k}} \tag{4}$$

(a) (b) (c) (d) (e) (f)

Fig. 1. (a)/(d) Cameraman/Lena image at different Noise Density i.e., 40%, 50%, and 55%. (b)/(e) Resultant image of Standard Median Filter. (c)/(f) Resultant image of proposed algorithm.

Quality evaluation for the test images were done for Switching Median Filter (SMF) [6], Progressive Switching Median Filter (PSMF) [7] and Proposed Optimal Detector Noise Filter (ODNF) at different noise density is tabulated in Table I.

TABLE I.
Quality evaluation for cameraman and Lena images.

	CAMERAMAN				LENA			
Noise Density	**Noise Image**	**SMF[6]**	**PSMF[7]**	**PA**	**Noise Image**	**SMF[6]**	**PSMF[7]**	**PA**
PSNR					**PSNR**			
20	12.08	23.98	25.10	30.47	12.45	26.52	32.37	34.26
50	8.06	14.18	18.50	24.34	8.47	14.96	30.06	27.35
70	6.64	9.48	9.39	20.70	7.01	10.00	9.88	23.00
80	6.05	7.76	7.71	18.71	6.41	8.09	7.98	20.36
MSE					**MSE**			
20	4004.64	261.58	200.88	55.45	3712.21	147.97	37.60	23.90
50	10144.9	2505.75	917.02	243.44	9384.84	2141.33	516.86	122.52
70	14171.0	7470.18	7471.20	531.34	12919.5	6472.77	6679.1	315.07
80	16286.2	11324.6	10996.00	833.01	14802	10046.6	10346	548.25
NAE					**NAE**			
20	0.214	0.0149	0.027	0.012	0.204	0.035	0.025	0.009
50	0.539	0.171	0.084	0.038	0.512	0.148	0.097	0.031
70	0.755	0.421	0.411	0.072	0.719	0.396	0.511	0.061
80	0.858	0.611	0.618	0.095	0.821	0.576	0.757	0.091
NCC					**NCC**			
20	0.965	0.987	0.988	0.997	0.974	0.993	0.991	0.998
50	0.924	0.968	0.975	0.991	0.943	0.974	0.977	0.995
70	0.886	0.921	0.921	0.986	0.926	0.950	0.996	0.989
80	0.872	0.895	0.907	0.975	0.920	0.935	0.999	0.990

IV. VLSI IMPLEMENTATION

Fig. 2 shows the block diagram of VLSI architecture for impulse noise detection algorithm. The architecture comprises of six main blocks: odd line buffers, even line buffers, register bank, sorter, threshold block and mean filter.

A. Line Buffer

The proposed algorithm implements a mask of size 3×3, thus requiring three scanning lines for the process. When current pixel is being processed, three pixels from each row will be desired to carry out the denoising process. By means of four cross-over multiplexers (Fig. 2) and two line buffers, three scanning lines are achieved. The pixels at odd and even rows are stored in Line Buffer-odd and Line Buffer-even, respectively. In the line buffer, a dual-port SRAM is used for read/write operation to lessen the cost and power consumed.

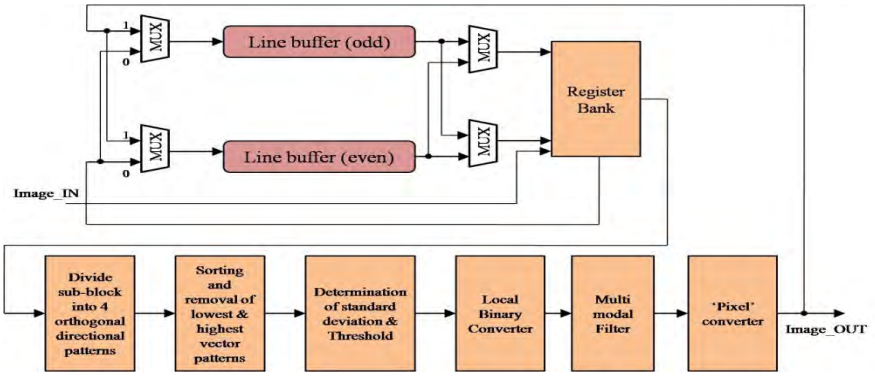


Fig. 2. VLSI architecture for impulse noise detection algorithm.

B. Register Bank

The Register Bank (RB) includes 12 registers—Reg0 to Reg11, which stores the 3×3 pixel values of the current mask. Fig. 3 illustrates the arrangement of RB in which each 3 registers are connected in series to

provide three pixel values of a row in mask and Reg4 keeps the luminance value of the current pixel to be denoised.

The luminance value from the input device enters the RB and immediately the denoising process starts. The twelve pixel values are stored in RB and then made use by consequent extreme data detectors and noise filters for denoising. After the denoising is complete, the reconstructed pixel values produced by the arbiter are fed to the line buffer. Suppose if we denoise row2 and all four selection signals are set to 0, the values of row1 and row2 will be stored in Line Buffer-odd and Line Buffer-even, respectively.

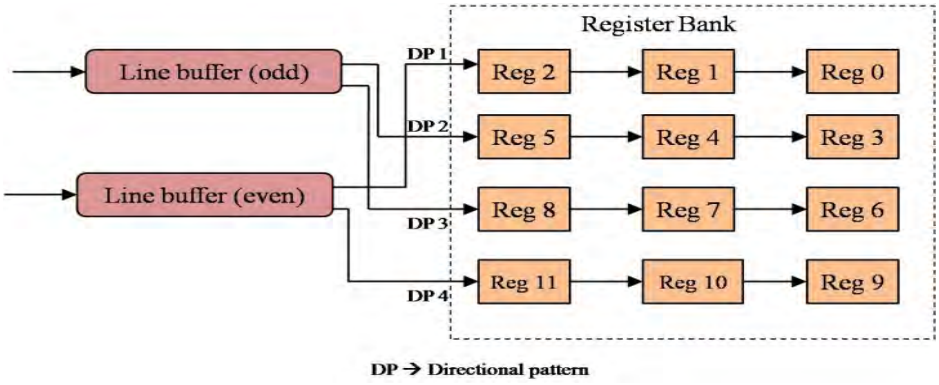


Fig. 3. Structural design of register bank.

C. Sorter block

The sorter block which arranges the orthogonal directional patterns is depicted in Fig. 4.

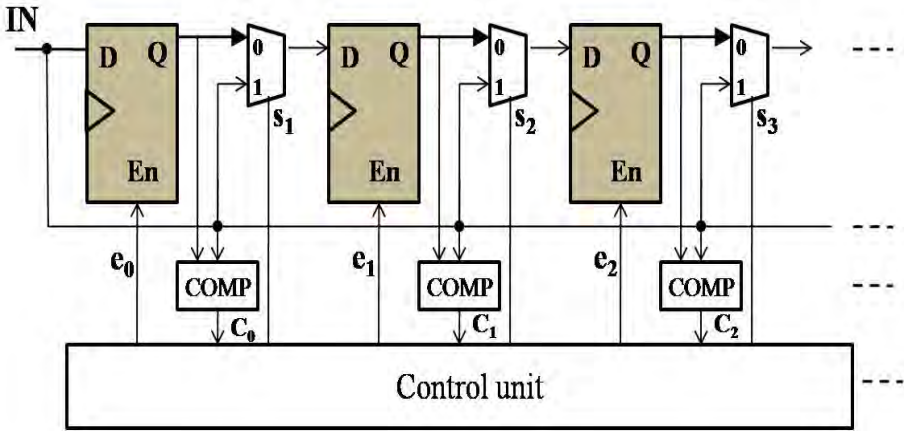


Fig. 4. Internal view of sorter block.

D. Threshold block

The threshold block as shown in Fig. 5 sets a threshold value Ts. For the pixel input values greater than the threshold Ts, the comparator output will be logic 1 and vice-versa. In our design, to obtain a value from one line buffer and load it into RB, a single clock cycle is enough and for the denoising process, the mean filter requires two clock cycles. The upper multiplexer outputs the pixel generated by comparator when it is noise-free and the other multiplexer forwards the noisy pixel to the mean filter for further processing.

E. The Mean filter

Fig. 6 shows the design of the mean filter in which the |ADD| unit finds the absolute sum of two inputs. The mapping module helps in locating the four optimal directions entirely consisting of noise-free pixels. The directional differences for the four directions are computed and the least value is decided by the DIV/9 unit.

The last block gives the filtered output, i.e. the mean of two pixel values.

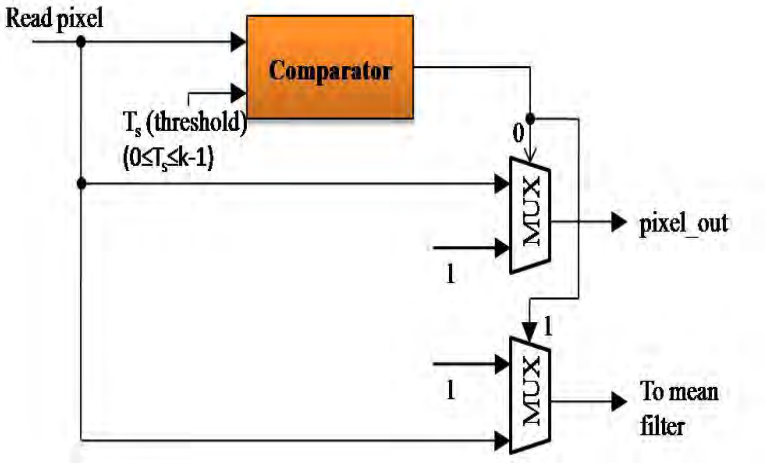


Fig. 5. Architecture of threshold block.

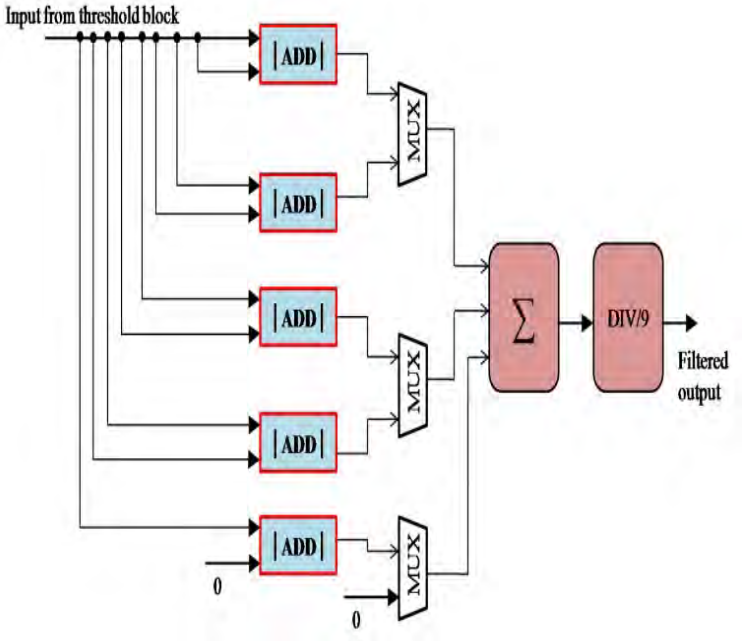


Fig. 6. Architecture of mean filter.

F. Local binary converter

The noisy image is divided into several regions from which the LBP feature distributions are extracted and concatenated into an enhanced feature vector to be used as an image feature descriptor. The source noisy image and its LBP feature extraction are shown in Fig. 7.

The Local binary conversion process includes various steps. First, a sub-image is formed from the source image. Then, the pixel values of the sub-image are extracted as shown in Fig. 7. A center pixel is chosen and each surrounding pixel is compared with the centre pixel value. If it is greater than the center value, it is replaced with '1', if found to be less, then it is replaced with '0'. The values of the surrounding pixels, thus found, are arranged in clockwise direction and its decimal value is found and replaced with the centre pixel value and so on.

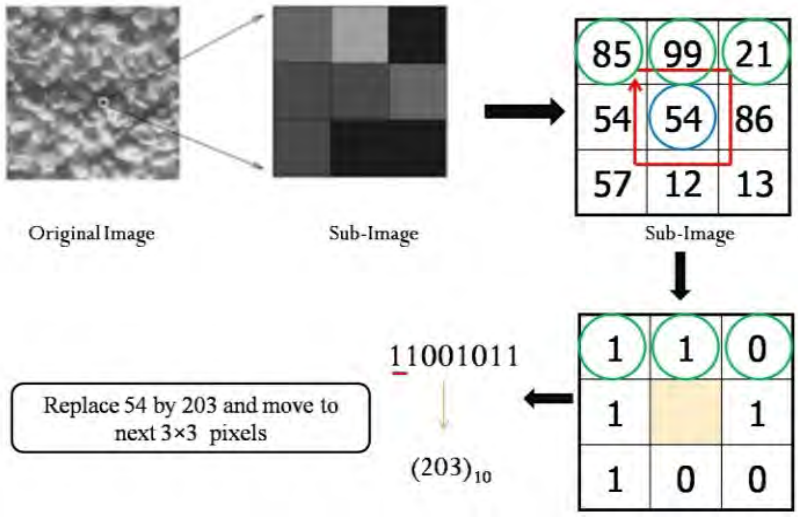


Fig. 7. Local binary conversion process.

G. Multimodal Filter

The multimodal filter has been employed for the process of removing the errors present in the denoised image. The filter operates in two modes depending on the intensity level of the boundary pixel. The modes are: DC offset mode and Default mode. If the boundary is in very smooth region with blocking artifact, then DC offset mode is selected, else default mode will be selected.

// THD1=2

// THD2=6

$$eq\_cnt = \varphi(v_0 - v_1) + \varphi(v_1 - v_2) + \varphi(v_2 - v_3) + \varphi(v_3 - v_4) + \varphi(v_4 - v_5) + \varphi(v_5 - v_6) + \varphi(v_6 - v_7) + \varphi(v_7 - v_8) + \varphi(v_8 - v_9) \quad (5)$$

where, $\varphi(\delta) = 1$ if $|\delta| \le THD1$ and 0, otherwise.

If (eq_cnt $\geq$ THD2), DC offset mode is selected and applied, else Default mode is selected and applied.

Default Mode:

$$V_4^{'} = V_4 - d \quad (6)$$

$$V_5^{'} = V_5 + d \quad (7)$$

where, $d = CLIP\left(5, \left(S^{'}_{3,0} - S_{3,0}\right) + 8, 0, \left(V_4 - V_5\right) \div 2\right) \partial\left(|a_{3,0}| < QNT\right)$;

$S^{'}_{3,0} = SIGN(S_{3,0}).MIN\left(|S_{3,0}|, |S_{3,1}|, |S_{3,2}|\right)$; $CLIP(x, p, q)$ clips x to a value between p and q; QNT represents the Quantization Parameter; and $\partial(condition) = 1$ if the condition is true and '0' otherwise.

DC offset mode:

Four pixels each are taken (right and left) from the boundary or edge region. Let the pixels be$V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8$. Next, the maximum value among these 8 pixels is noted as MAX and the minimum

value is noted as MIN. The absolute difference between MAX and MIN is calculated and compared with the Quantization Parameter (QNT)

Based on the comparison result, the new artifact removed pixels are found as,V_{11},V_{21},V_{31},V_{41},V_{51},V_{61},V_{71},V_{81}.Finally, the old pixels V_1,V_2,V_3,V_4,V_5,V_6,V_7,V_8. are replaced with the new pixels V_{11},V_{21},V_{31},V_{41},V_{51},V_{61},V_{71},V_{81}.

$$MAX = \max(V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8) \tag{8}$$

$$MIN = \min(V_1, V_2, V_3, V_4, V_5, V_6, V_7, V_8) \tag{9}$$

$if\ (|MAX - MIN| < 2 * QNT)$, then

$$V_n^{'} = \sum_{k=-4}^{4} b_k . p_{n+k,} 1 \le n \le 8 \tag{10}$$

$$p_m = \left\{ v_{m\ ,} \begin{matrix} (|v_1 - v_0| < QNT\ \ ) ? v_0 : v_{1\ ,} if m < 1, if\ 1 \le m \le 8 \\ (|v_8 - v_9| < QNT\ \ ) ? v_9 : v_{8\ ,} if m > 8 \end{matrix} \right\} \tag{11}$$

and $\{b_k : -4 \le k \le 4\} = \{1,1,2,2,3,2,2,1,1\} \div 16$ (12)

V. RESULTS AND DISCUSSION

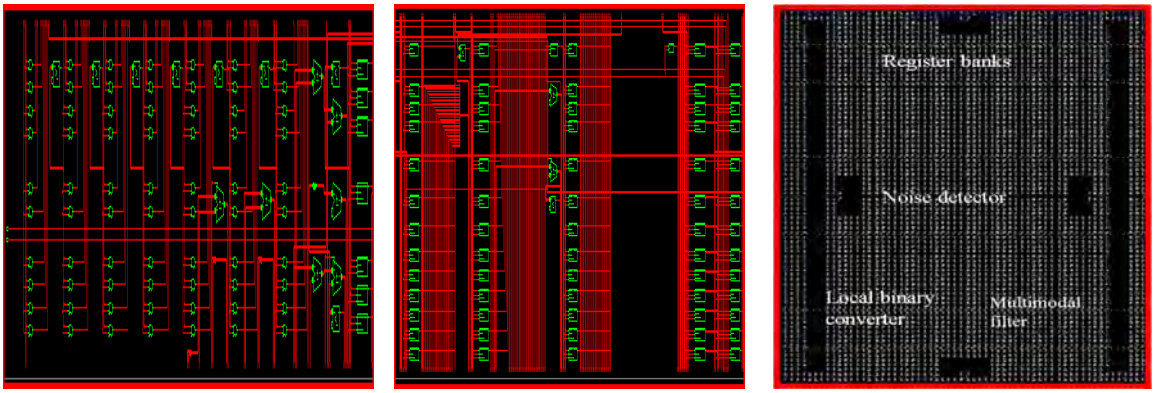
A. Evaluation details of hardware architecture

The proposed Optimal detector noise filtering algorithm and its hardware architecture system is designed and tested on various version of FPGA family device against their hardware utilization and latency, tabulated in Tables I, II and III, and also graphically plotted in Fig. 9 and Fig. 10. The proposed work results shows that the system incorporated with its hardware architecture leads to lower hardware consumption. The proposed denoising architecture is implemented in 90nm CMOS technology. The post layout results of the proposed de-noising architecture are summarized in Table IV, and the chip layout is shown in Fig. 8. The slice utilization is about 345 in virtex 5 FPGA family.

TABLE II.

Performance estimation of hardware utilization in XC5VLX30 VIRTEX 5 device.

Hardware Utilization Parameters	Utilizations
Adders/Subtractors	68
Latches	21
Comparators	60
IO Buffers	954
Logic Cells	345
Latency	12.265ns



(a) (b) (c)

Fig. 8. Simulation results of proposed technique. (a) RTL schematic view, (b) Technology schematic view, and (c) Chip layout of proposed

TABLE III

Comparison of latency on FPGA family.

FPGA Family	Device Specifications	Latency(ns)
Spartan 3E	XC3S1600E	9.905
Virtex 2	XC2V40	10.371
Virtex 4	XC4VLX15	6.32
Virtex 5	XC5VLX30	4.518
Virtex E	XCV50E	12.265

TABLE IV

Performance comparison in terms of hardware utilizations

Methodology	Logic cells Utilizations
Proposed Method	345
RSEPD[1]	709
NAVF[2]	2670
DTBMD[3]	1709

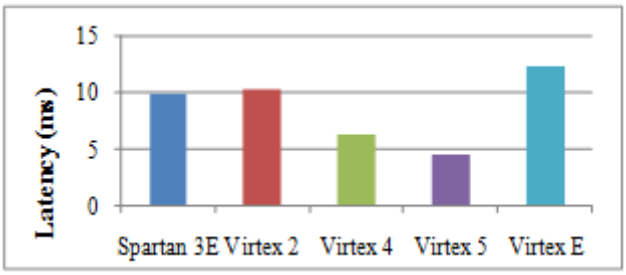


Fig. 9. Graphical plot for comparison of Latency of various FPGA devices.

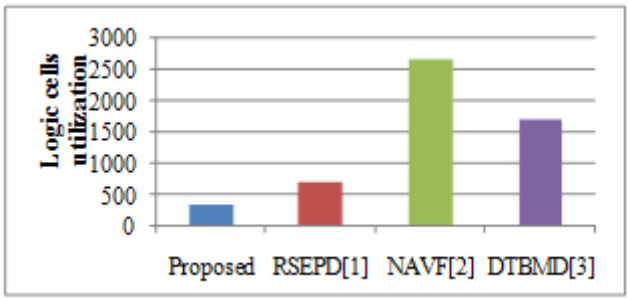


Fig. 10. Graphical plot for comparison of hardware utilizations.

VI. Conclusions

In this paper, an optimal detector noise filtering algorithm and its efficient hardware architecture is proposed which detects and removes the high density impulse noise from the noise affected images. The quality analysis of the algorithm is employed with SMF, PSMF and ODNF in terms of PSNR, MSE, NAE, and NCC. The experimental results show that the proposed algorithm outperforms the conventional denoising algorithms. At the same time, proposed hardware architecture is simple and consumes minimum hardware utilities. The design is implemented with low hardware cost and is suitable for real-time applications. The proposed hardware architecture is tested on various FPGA hardware devices and examined to have less hardware utilization.

References

[1] I. Andreadis, G. Louverdis, “Real-time adaptive image impulse noise suppression,” IEEE Trans. Instrum. Meas., vol. 53, no. 3, pp.798–806, 2004.

[2] V. Fischer, R. Lukac, K. Martin, “Cost-effective video filtering solution for real-time vision systems,” EURASIP J. Appl. Signal Process., vol. 2005, no. 13, pp. 2026–2042, 2005.

[3] P. Y. Chen, C. Y. Lien, Hsu-Ming Chuang, “Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 18, no. 3, pp. 473–481, 2010.

[4] Shih-Chang Hsia, “Parallel VLSI Design for a Real-Time Video-Impulse Noise-Reduction Processor,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 4, pp. 651-658, 2003.

[5] S. P. Philip, S. P. Prakash, S. Valarmathy, “An Efficient Decision Tree Based Architecture for Random Impulse Noise Removal in images,” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 2, Issue 4, 2013.

[6] S. Zhang, M. A. Karim, “A new impulse detector for switching median filters,” IEEE Signal Processing Letters, vol.9, no. 11, pp. 360-363, 2002.

[7] Z. Wang, D. Zhang, “Progressive switching median filter for the removal of impulse noise from highly corrupted images,” IEEE Trans. Of Circuits and Systems II, vol. 46, no.1, pp. 78-80, 1999.

[8] Chih-Yuan Lien, Chien-Chuan Huang, Pei-Yin Chen, Yi-Fan Lin, “An Efficient Denoising Architecture for Removal of Impulse Noise in Images,” IEEE Trans. Computers, vol. 62, no. 4, 2013.