IJCTA, 9(12), 2016, pp. 5609-5622

An Efficient VLSI Architecture to Remove Impulse Noise in Images Using Decision Tree Based Denoising Method

***Mrs. S.Murugappriya **R.Kanimozhi ***Dr. G.R.Suresh**

Abstract : Impulse noise often corrupts the images in the procedures of image acquisition and transmission. Denoising of image corrupted by impulse noise is a prominent research area in Image Processing.To carry out noise suppression many denoising schemes introduced which uses standard median filter or its modifications. However, these approaches might blur the image since both noisy and noise-free pixels are modified. An efficient denoising scheme called Decision Tree Based Denoising Method (DTBDM) and its VLSI architecture introduced to avoid the damage on noise-free pixels and also for the removal of high density impulse noise. Decision Tree Based Denoising method is performed as two phase process—a detection phase and a filtering phase. Noisy pixels will be detected by decision-tree-based impulse noise detector followed by a direction oriented edge-preserving median filter. Edge preserving filter mainly reconstructs the intensity values of noisy pixels in order to achieve optimal detail edge preservation. The performance of this method is analysed quantitatively and qualitatively using performance metrics like Peak Signal to Noise Ratio(PSNR),Image Enhancement factor(IEF) and Mean Square Error(MSE). Simulation results indicate that the direction oriented edge preserving filter impressively outperforms other techniques in terms of noise suppression and detail preservation across a wide range of impulse noise corruption.DTBDM algorithm is implemented in VLSI architecture to achieve the goal of low cost, a low complexity and low power. It is found that the given method can be best suited for denoising chip in consumer electronics.

Keywords : Impulse noise; Denoising method; DTBDM; Denoising chip.

1. INTRODUCTION

In the area of image processing ,image denoising is the primary predominant step in removal of impulse noise.Aim of denoising is to remove the impulse noise from the original image.The application of digital images spreads widely in Medical,Underwater,Robot vision,Consumer electronic products.Digital images are corrupted by impulse noise frequently during the process of acquisition,storage and transmission.As a result quality of image is degraded, hence it is very much essential to prevent the degradation of image quality which is superimposed by impulse noise.

Impulsive noise, which can be caused by malfunctioning camera photosensors, optic imperfections,electronic instability of the image signal, aging of the storage material, faulty memory locations in hardware or transmission errors due to natural or man-made processes [15-18].Noise removal researches have been performed in the area of image processing [1-14].

Traditional method of noise removal is median filter[1].This filter is effective in impulse noise removal only for low density impulse noise.For high noise density images thin lines,edges are not preserving methods effectively.Hence various filtering methods such as weighted median (WM) filter [19], center weighted median (CWM) filter [20]

* Department Of ECE, Easwari Engineering College
** Department Of ECE, Easwari Engineering College
*** Department Of ECE, Easwari Engineering College

and adaptive center weighted median (ACWM) filter [21], decision based adaptive filter [22], switching median filter [23], mean based rank order filter [24],etc used.These filters works well for high density impulse noise but preserving details is still an important issue.

In general denoising method involve detection phase and a filtering phase.Many denoising methods exist such as differential rank impulse detector (DRID) method(25), rank-ordered relative differences (RORDWMF) method (26), directional weighted median (DWM) method (27), these methods are of high complexity.One of the lower complexity decision tree based denoising method (DTBDM) is proposed to detect and remove the noisy pixels.

2. PROPOSED TECHNIQUE

Decision Tree Based Denoising Method(DTBDM) is a two stage process-a detector stage and filtering stage. A 3*3 window size is designed for noise removal.To achieve the goal of low cost normally lower complexity techniques are usually prefered .since it requires use of fixed size local window it can be easily implemented in VLSI architecture.

In order to enhance the robustness of the image ,the removal of impulse noise is necessary.One of the simple and powerful form of multivariable analysis is decision tree.The complex process is broken into simple form so that solution can be obtained easily. According to the input sequence of image denoising process, we can divide other eight pixel values into two sets: $W_{Top\ Half}$ and $W_{Bottom\ Half}$.

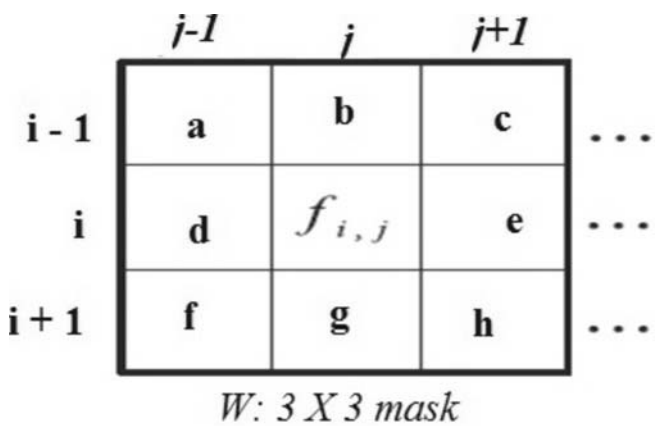


Fig. 1. A 3*3 Mask Centered On $P_{i,j}$

$$W_{Top\ Half} \ : \ \{a, b, c, d\} \tag{1}$$

$$W_{Bottom\ Half} \ : \ \{e, f, g, h\} \tag{2}$$

A. Algorithm

Step 1. The process starts with reading the original image .

Step 2. Convert it to a noisy image by adding a noise density.

Step 3. The pixels are converted to the window size of 3 × 3.

Step 4. Detect whether each pixel is noisy or not using Decision tree based impulse detector.

Step 5. If the pixel is noisy,it is given to the Direction oriented edge preserving median filter.

Step 6. Compute the median value for that noisy pixel.

Step 7. Compute the restoration term.

Step 8. Finally replace the noisy pixel with the restoration term value.

Step 9. The above steps are to be followed iteratively for each and every pixel in the noisy image with a window size of 3 × 3.

Step 10. A noise free image is obtained. Calculate the PSNR , MSE, IEF value now.

B. Block Diagram

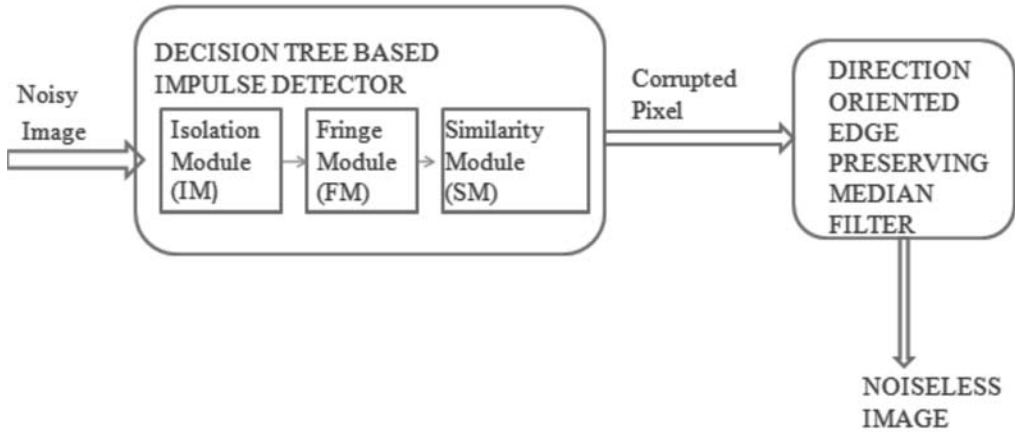


Fig. 2.

C. Decision Tree Based Impulse Detector

Isolation module (IM), Fringe module (FM), and Similarity module (SM)are the three modules that builds a decision tree in detector stage.Concatenating decisions of these three modules is taken into account to detect a noisy pixel.By using different equations in different modules status of $P_{i,j}$ is determined.

First,Isolation Module(IM) will detect whether the pixel is in smooth region.If result is positive it is given to the fringe module to confirm the pixel is noisy pixel or situated on the edge .If Fringe module(FM)result is also positive ,Similarity module(SM) will confirm the result by checking the similarity between current pixel and neighbouring pixel.Only when similarity module result is also positive we can confirm the detected pixel is a noisy pixel.

1. Isolation module and its Architecture

Usually in a region the pixel values should be close or slightly varying.However the distribution of pixel values are slightly different in a region then it may be noisy pixel or edge pixel.By observing the smoothness of the region we can determine whether the pixel value is isolated from its neighouring pixel values. Top half and bottom half differences are calculated and compared with a predefined threshold value Th_IMa.

$$\text{Top Half\_diff} = \text{Top Half\_max} - \text{Top Half\_min}$$

$$\text{Bottom Half\_diff} = \text{Bottom Half\_max} - \text{Bottom Half\_min}$$

$$\text{Decision I} = \begin{cases} \text{true if} \begin{pmatrix} \text{Top Half}_{\text{diff}} \geq \text{Th\_IMa} \\ \text{or} \\ \text{Bottom Half}_{\text{diff}} \geq \text{Th\_IMa} \end{pmatrix} \\ \text{false otherwide} \end{cases} \quad (3)$$

$$\text{IM\_Top Half} = \begin{cases} \text{true if}(|f_{ij} - \text{Top Half\_Max}|\text{Th}_1 \geq M_b \\ \qquad \text{or} \\ (|f_{ij} - \text{Top Half\_Min}|\text{Th}_1 \geq M_b \\ \text{False} \qquad \text{otherwise} \end{cases} \quad (4)$$

$$\text{IM\_Bottom Half} = \begin{cases} \text{true if } (|f_{i,j} - \text{Bottom Half\_Max}| \geq Th_1M_b \\ \quad \text{or} \\ (|f_{i,j} - \text{Bottom Half\_Min}| \geq Th_1M_b \\ \text{False} \quad \text{otherwise} \end{cases} \tag{5}$$

$$\text{Decision II} = \begin{cases} \text{true if} \begin{pmatrix} \text{IM\_Top Half} = \text{true} \\ \text{or} \\ \text{IM\_Bottom Half} = \text{true} \end{pmatrix} \\ \text{false} \quad \text{otherwise} \end{cases} \tag{6}$$

IM tophalf and IM bottom half is calculated using predefined threshold value Th_IMb.

The comparator CMP_L and CMP_S are used to compare two input values and give the output larger and smaller value respectively.The absolute difference of two input value is given by |SUB| unit.If the upper input value is larger than smaller one then GC Greater Comparator will output logic1.If the result is true it is a noisy or edge pixel.

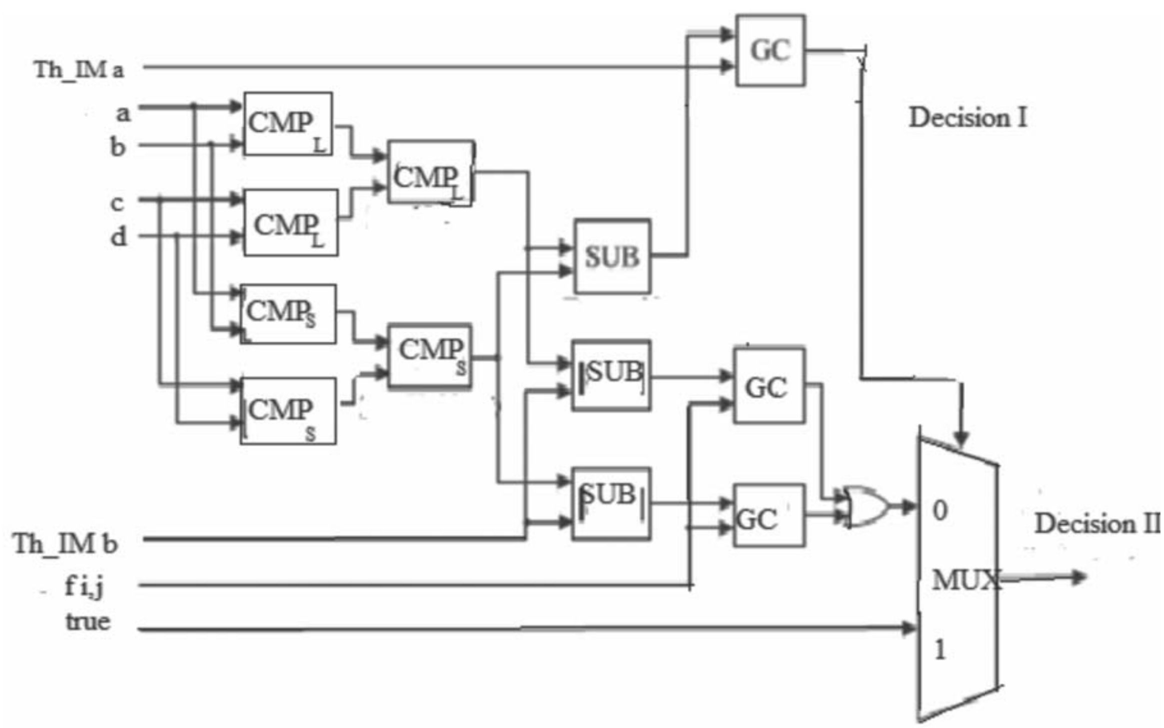


Fig. 3. Architecture Of Isolation Module

2. Fringe module and its Architecture

Fringe module used to check whether the pixel is a noisy pixel or edge pixel by considering along four edge directions.

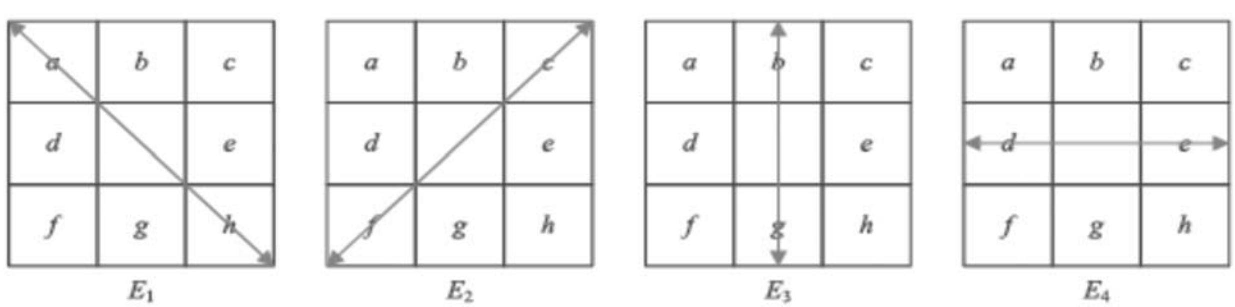


Fig. 4.

$$FM\_E_1 = \begin{cases} \text{False if } (|a - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ (|h - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ \text{True} \qquad \text{otherwise} \end{cases} \tag{7}$$

$$FM\_E_2 = \begin{cases} \text{False if } (|c - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ (|f - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ (|c - f| \geq Th_{FMb}) \\ \text{True} \qquad \text{otherwise} \end{cases} \tag{8}$$

$$FM\_E_3 = \begin{cases} \text{False if } (|b - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ (|g - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ (|b - g| \geq Th_{FMb}) \\ \text{True} \qquad \text{otherwise} \end{cases} \tag{9}$$

$$FM\_E_4 = \begin{cases} \text{False if } (|d - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ (|e - f_{i,j}| \geq Th\_FM_a) \\ \text{or} \\ (|d - g| \geq Th_{FMb}) \\ \text{True} \qquad \text{otherwise} \end{cases} \tag{10}$$

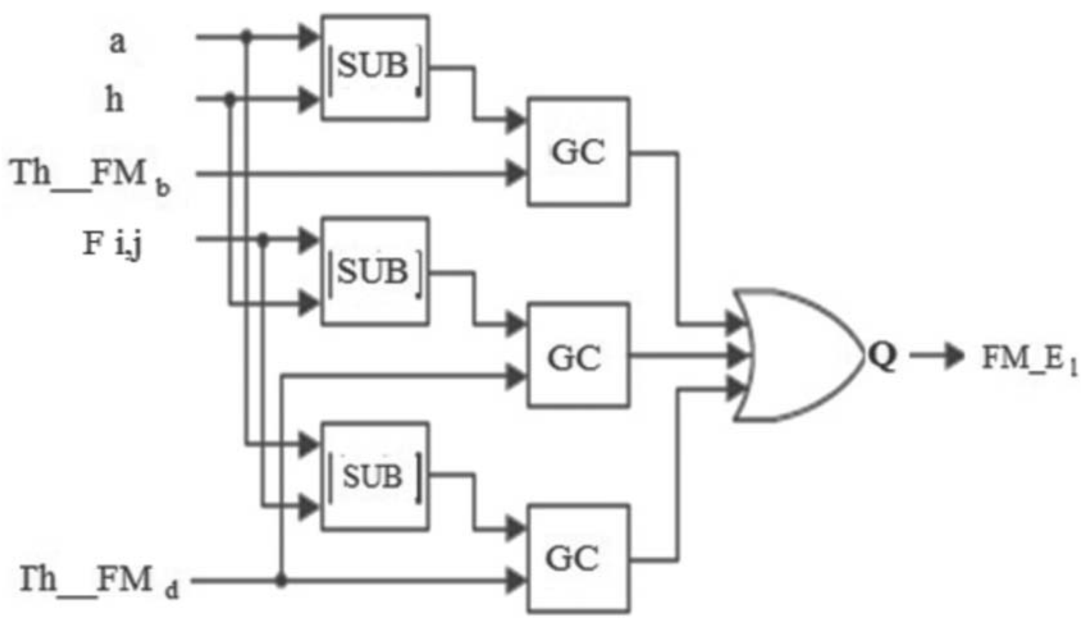


Fig. 5. Architecture Of Fringe Module

Fringe module consist of four modules FM_E_1 to FM_E_4 to determine its direction.If the NOR gate result is positive then the corresponding pixel is edge pixel and its noise free.If the result is negative then similarity module will confirm the result.

3. Similarity module and its Architecture

Similarity module used to confirm the result of noisy pixel.In the corresponding mask if there is a variation of extreme big or small values then there is a possibility of noisy pixel.The corresponding mask nine pixel values are sorted in ascending order and the fourth,fifth and sixth value close to median is chosen.

$$Max_{i,j} = 6^{th} \text{ in } W_{i,j} + Th\_SM_a \tag{11}$$

$$Min_{i,j} = 4^{th} \text{ in } W_{i,j} - Th\_SM_a \tag{12}$$

Max $_{i,j}$ and Min $_{i,j}$ are used to determine the status of pixel $P_{i,j}$. If f $_{i,j}$ is not between N max and N min, we conclude that $P_{i,j}$ is a noise pixel.

$$N_{max} = \begin{cases} Max_{i,j,} \text{ if}(Max_{i,j} \leq \text{Median } W_{i,j} + Th\_SM_b) \\ \text{Median ln } W_{i,j} + Th_{SM_b}, \text{ otherwise} \end{cases} \tag{13}$$

$$N_{min} = \begin{cases} Min_{i,j}, \quad \text{if}(Min_{i,j} \leq \text{ Median ln } W_{i,j} - Th\_SM_b) \\ \text{Median ln } W_{i,j} - Th_{SM_b}, \text{otherwise} \end{cases} \tag{14}$$

If $f_{i,j}$ is not between N_{max} and N_{min}, we conclude that $P_{i,j}$ is a noise pixel. The equation as

$$\text{Decision IV} = \begin{cases} \text{true, if } (f_{i,j} \geq N_{max}) \text{ or } (f_{i,j} \leq N_n \\ \text{false, otherwise} \end{cases} \tag{15}$$

The fixed values of thresholds Th IM_a, TH IM_b, Th FM_a, Th FM_b, Th SM_a, and Th SM_b are predefine values and set as 20, 25, 40, 80, 15, and 60, respectively.

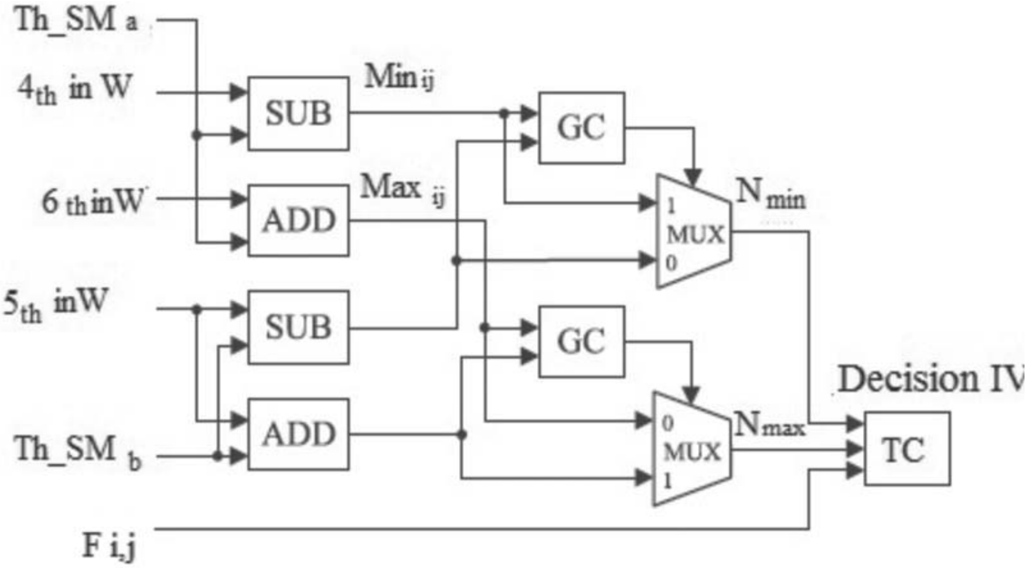


Fig. 6. Architecture Of Similarity Module

Max *i,j* and Min *i, j* is determined from ADD and SUB units.TC,Triple comparator outputs logic 1 if the lowest input is not between the upper two inputs.

D. Direction Oriented Edge-Preserving Median Filter

Filter is composed of two modules, min ED generator and average generator (AG).D1 to D8, eight directional differences are considered to reconstruct the noisy pixel value, as shown in Fig. 5. Only noise-free pixels are considered to remove the noise.

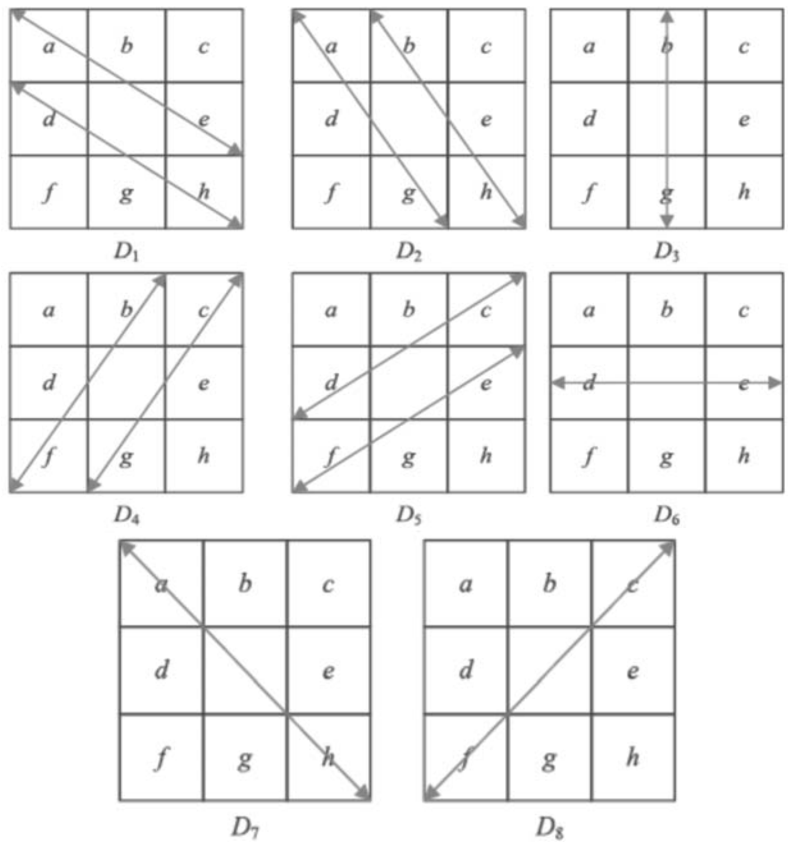


Fig. 7. Eight directional Differences

1. Architecture of min ED generator

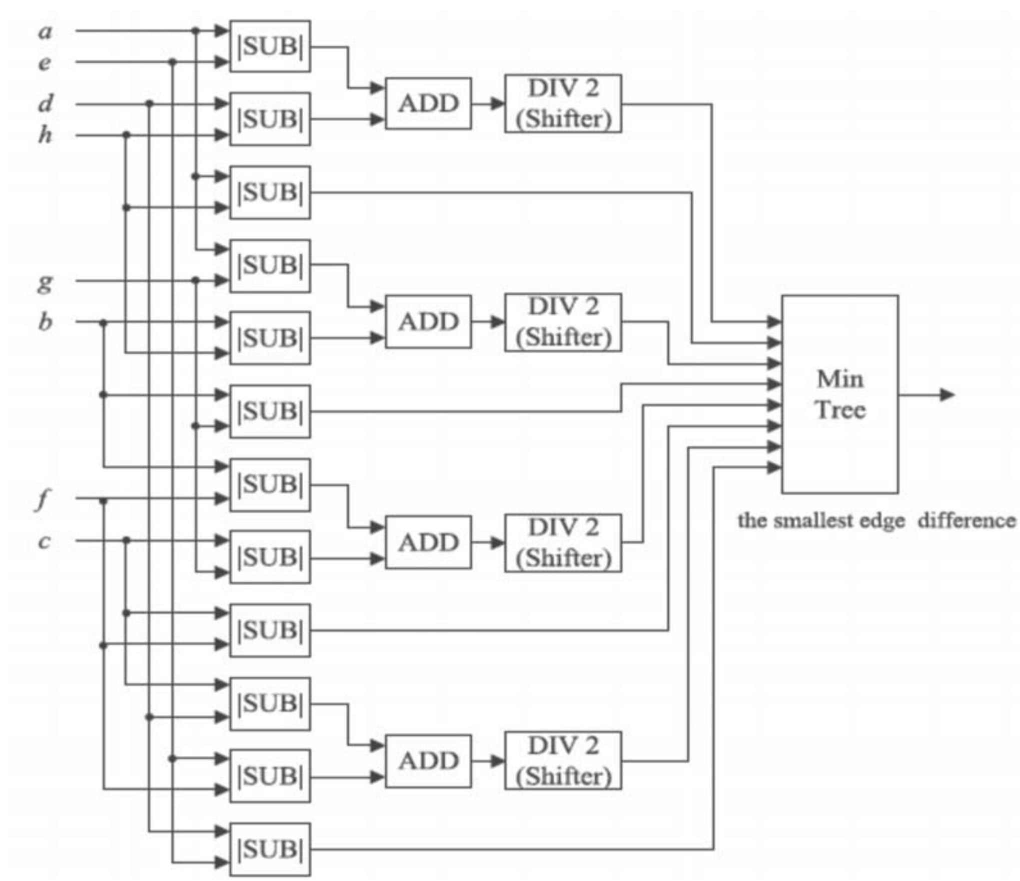


Fig. 8. Architecture Of MinED generator

The Edge that has the smallest difference determined using MinED generator. Eight directional differences are calculated with twelve |SUB|, four ADD, and four shifter units.

$$\begin{aligned} D_1 &= |d-h| + |a-c| \\ D_2 &= |a-g| + |b-h| \\ D_3 &= |b-g| \times 2 \\ D_4 &= |b-f| + |c-g| \\ D_5 &= |c-d| + |e-f| \\ D_6 &= |d-e| \times 2 \\ D_7 &= |a-h| \times 2 \\ D_8 &= |c-f| \times 2 \end{aligned} \tag{16}$$

The mean luminance value of the pixels is determined from average generator by selecting D_{min} value.

Equation of average generator :

$$\hat{f}_{i,j} = \begin{cases} (a+d+e+h)/4, & \text{if } D_{min} = D_1, \\ (a+b+g+h)/4, & \text{if } D_{min} = D_2, \\ (b+g)/2, & \text{if } D_{min} = D_3, \\ (b+c+f+g)/4, & \text{if } D_{min} = D_4, \\ (c+d+e+f)/4, & \text{if } D_{min} = D_5, \\ (d+e)/2, & \text{if } D_{min} = D_6, \\ (a+h)/2, & \text{if } D_{min} = D_7, \\ (c+f)/2, & \text{if } D_{min} = D_8, \end{cases} \tag{17}$$

The values of $f_{i,j}$ obtained by the median of four neighboring pixels (b, d, e, and g). We can express as

$$\hat{f}_{i,j} = \text{Median}(\hat{f}_{i,j}, b, d, e, g). \tag{18}$$

N_{max} and N_{min} defined in similarity module will act a threshold value in this filter stage.

4. EXPERIMENTS RESULTS AND DISCUSSIONS

4.1. Testing procedure

The implementation of the direction oriented edge preserving median filter are carried out using MATLAB R2014a for impulse noise corrupted on lena image and coin image.

It is illustrated in Fig 1. and Fig 2.

Fig. 9. Lena Image

Fig. 10. Coin Image

For these images the performance of the direction oriented filter is compared with other filters – Median Filter(MF), Adaptive Centre Weighted Median Filter(ACWMF), Decision based Median Filter(DBMF), Direction Oriented Edge Preserving Median Filter(DOEPMF).

4.2. Simulation Results

Intensive simulations were carried out using grayscale lena and coin images.These images are chosen for demonstration of low to high noise densities. The performance evaluation of the filtering operation is quantified by the PSNR (Peak Signal to Noise Ratio) and MSE (Mean Square Error)and IEF(Image Enhancement factor) calculated using formula:

$$\text{PSNR} = 10\log(L^{\wedge}2/\text{MSE})$$

Where, $$\text{MSE} = [g(i,j)\ f(i,j)]^2$$

Where, g denotes the Noise image and f denotes the filtered image.

$$\text{IEF} = \left(\sum_{m,n}[p(m,n), o(m,n)^2\right) \Big/ \left(\sum_{m,n}[\text{R}(m,n), o(m,n)^2\right)$$

Table 1. Lena Image for Filters Vs Noise Densities

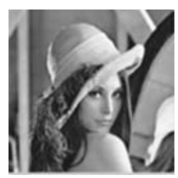
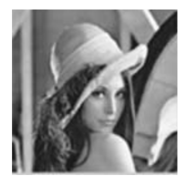
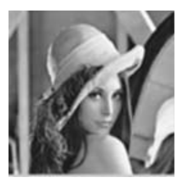
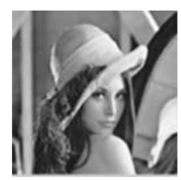
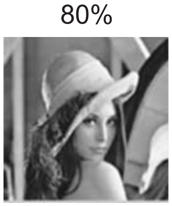
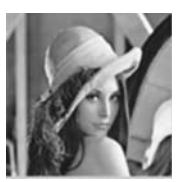
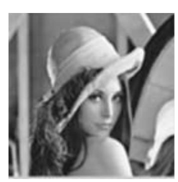
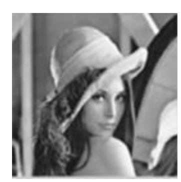
Noise Densities	***Median Filter***	***Lena Image Adaptive Centre Weighted median Filter***	***Decision based Median Filter***	***Direction Oriented Edge Preserving median Filter***
20% 				
40% 				
60% 				
80% 				

Table 2. Restoration Results MSE for Lena Image, Filters vs Noise Densities

Noise Densities	***PSNR Value-Filtered Len Image***			
	Median Filter	***Adaptive Centre Weighted Median Filter***	***Decision based Median Filter***	***Direction Oriented Edge Preserving median Filter***
20%	100	21	1925.2	2.39
40%	261.8	50.27	4291.67	2.58
70%	864.3	162.6	7785.2	3.52
90%	3998	963.1	13167.5	4.39

Table 3. Restoration Results PSNR for Lena Image, Filters vs Noise Densities

Noise Densities	*PSNR Value-Filtered Len Image*			
	Median Filter	*Adaptive Centre Weighted Median Filter*	*Decision based Median Filter*	*Direction Oriented Edge Preserving median Filter*
20%	28.16	15.32	36.90	44.38
40%	23.58	11.84	30.34	46.36
70%	18.00	9.25	26.79	44.30
90%	12.15	6.97	20.42	44.38

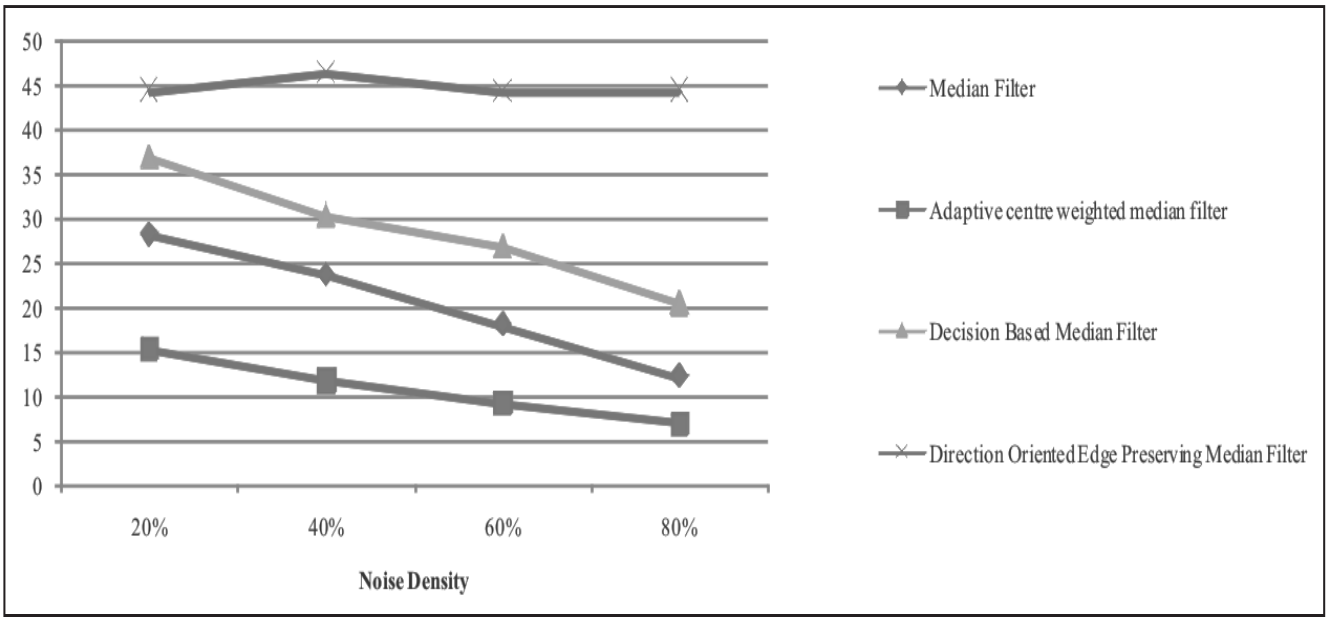


Fig. 11. Graphical Representation of Restoration Results PSNR for Lena Image, Filters vs Noise Densities.

Table 4. Restoration Results IEF for Lena Image, Filters vs Noise Densities

Noise Densities	*IEF Value-Filtered Len Image*			
	Median Filter	*Adaptive Centre Weighted Median Filter*	*Decision based Median Filter*	*Direction Oriented Edge Preserving median Filter*
20%	19.00	10.06	309.26	1704.38
40%	31.28	4.64	213.07	3393.48
60%	14.13	2.60	88.70	5110.13
80%	4.09	1.53	17.33	6799.14

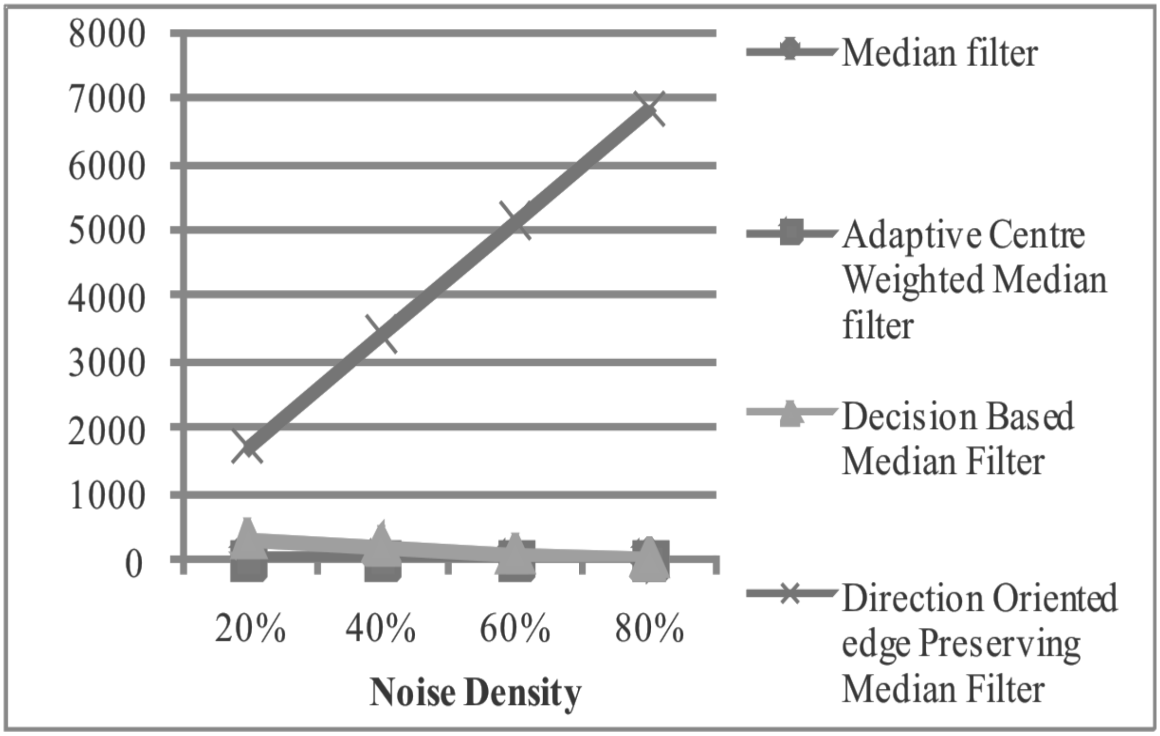


Fig. 12. Graphical Representation of Restoration Results IEF.

Table 5.

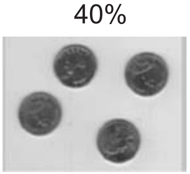
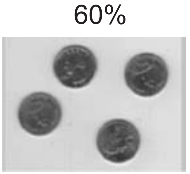
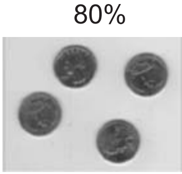
Noise Densities	*Coin Image Median Filter*	*Adaptive Centre Weighted Median Filter*	*Decision based Median Filter*	*Direction Oriented Edge Preserving median Filter*
20% 				
40% 				
60% 				
80% 				

Table 6. Restoration Results MSE for Coin Image, Filters vs Noise Densities

Noise Densities	*MSE Value-Coin Image*			
	Median Filter	*Adaptive Centre Weighted Median Filter*	*Decision based Median Filter*	*Direction Oriented Edge Preserving median Filter*
20%	173.45	37.75	4291.16	5.37
40%	391.87	50.58	8871.69	5.47
60%	874.19	222.13	14862.4	6.42
80%	4722.2	1497.2	22112.6	7.48

Table 7. Restoration Results PSNR for Coin Image, Filters vs Noise Densities

Noise Densities	*MSE Value-Coin Image*			
	Median Filter	*Adaptive Centre Weighted Median Filter*	*Decision based Median Filter*	*Direction Oriented Edge Preserving median Filter*
20%	25.77	11.64	35.37	43.87
40%	22.96	8.68	34.61	44.87
60%	18.75	6.44	25.43	40.87
80%	11.42	4.72	16.54	43.87

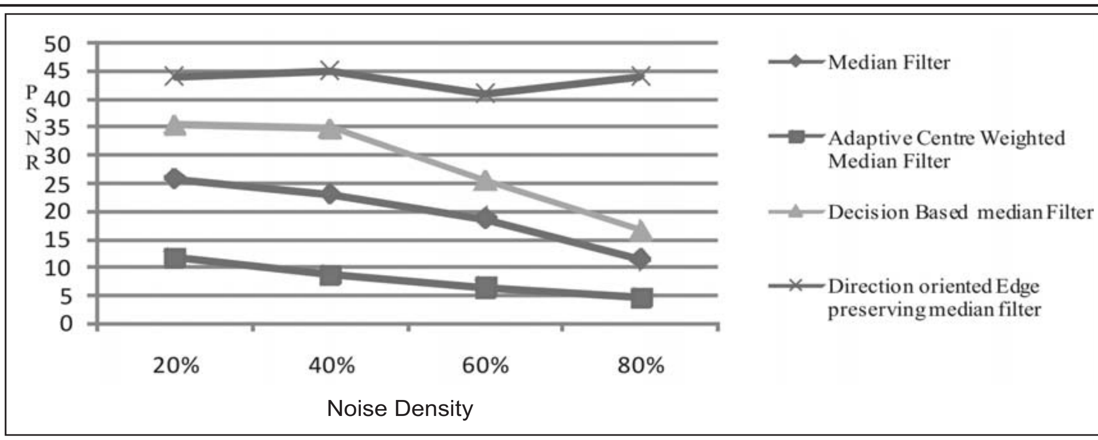


Fig. 13. Graphical Representation of Restoration Results PSNR-Coin.

Table 8. Restoration Results IEF for Coin Image, Filters vs Noise Densities

Noise Densities	*MSE Value-Coin Image*			
	Median Filter	*Adaptive Centre Weighted Median Filter*	*Decision based Median Filter*	*Direction Oriented Edge Preserving median Filter*
20%	29.05	8.99	248.39	911.09
40%	29.06	4.14	292.11	1795.46
60%	16.41	2.45	76.94	2751.69
80%	4.08	1.05	13.29	2677.89

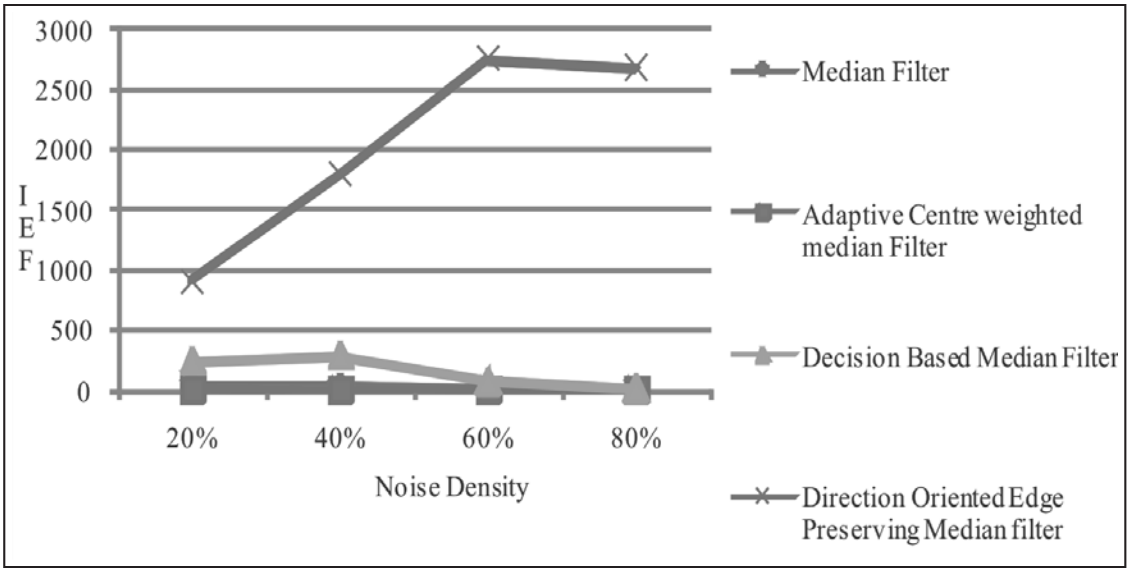


Fig. 14. Graphical Representation of Restoration Results IEF -Coin image.

From table 1 to 8 the performance metrics of MSE,PSNR,IEF it is clear that direction oriented edge preserving median filter outperforms other standard filters (MF,AWCMF,DBMF) in suppressing noises and preserving fine edge details. The DOEPMF filter is better in performance compared to the other filters.

The simulation of the entire DTBDM method is done using Xilinx Spartan 4s.The area and power estimated using cadence cmos 250nm lib.

In the DTBDM method it is understood that noisy pixels are detected by the detector and given to the filter where noisy pixels are removed.Another advantageous of our method is the use of predefined threshold value.The 3x3 window size is used because of lower complexity and there is no threshold measurement.

Table 9. Xilinx Spartan Area and Speed result

AREA	Number of Slices Used	743/12480
SPEED	Combinational Path Delay	28.96ns

As the noise increases more than 60% the other type of standard filters MF,ACWMF,DBMF degrade in performance rapidly whereas our technique shows very slow degradation even for high noise density.It can recover 80% to 90% of original image from the corrupted image.The simulation result of Xilinx results 743 slices out of 12480 with time delay of 28.96ns. The simulation result of cadence results an area of 176070 Sq.m and total power of 53.79mW.

5. CONCLUSION

Decision Tree Based Denoising Method (DTBDM) is composed of decision tree based impuse detector and direction oriented edge preserving median filter to detect and reconstruct the intensity of noisy pixels.Quantitave and qualitative measurement clearly says that this method preserved all the local information even after the image is degraded with high density impulse noise of 80% and enhances the robustness of the image. Contrary to many existing filters that only focus on a particular impulse noise , the DTBDM is capable of removing all kinds of impulse noise – the random-valued and/or fixed-valued impulse noise.Hence can be used as a denoising chip in consumer electronics real time applications.

6. REFERENCES

1. Tukey, J.W.: Conference Record EASCON'74, p. 673 (1974)
2. Sun, T., Neuvo, Y.: Pattern Recognit. Lett. 15, 341 (1994)

3. Ronse, C.: Signal Process. 21, 129 (1990)
4. Jacquin, A., Okada, H., Crouch, P.: Proceedings of the Data Compression Conference, p. 111 (1997)
5. Tomasi, C., Manduchi, R.: Sixth International Conference on Computer Vision, p. 839 (1998)
6. Takeda, H., Farsiu, S., Milanfar, P.: IEEE Trans. Image Process.16, 349 (2007)
7. Donoho, D.L.: IEEE Trans. Inf. Theory 41, 613 (1995)
8. Heidarzadeh, A., Avanaki, A.N.: 9th International Symposium on Signal Processing and Its Applications, p. 1 (2007)
9. Orchard, J., Ebrahimi, M., Wong, A.: 15th IEEE International Conference on Image Processing, p. 1732 (2008)
10. Matsui, S., Okabe, T., Shimano, M., Sato, Y.: 9th Asian Conference on Computer Vision, p. 213 (2009)
11. Tasdizen, T.: IEEE Trans. Image Process. 18, 2649 (2009)
12. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: IEEE Trans.Image Process. 16, 2080 (2007)
13. Wu, J., Tang, C.: Signal Image Video Process. 8, 349 (2014)
14. Buades, A., Coll, B., Morel, J.-M.: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2, p. 66 (2005)
15. Neuvo, Y., Ku, W.: Analysis and digital realization of a pseudorandom Gaussian and impulsive noise source. IEEE Trans. Commun. 23(9), 849–858 (1975)
16. Boncelet, CG.: Image noise models. In: Bovik, A. (ed.) Handbook of Image and Video Processing, pp. 325–335. Academic Press(2000)
17. Zheng, J., Valavanis, K.P., Gauch, J.M.: Noise removal from color images. J. Intell. Robot. Syst. 7(1), 257–285 (1993)
18. Faraji, H., James MacLean, W.: CCD noise removal in digital images. IEEE Trans. Image Process. 15(9), 2676–2685 (2006)
19. O. Yli-Harja, J. Astola, Y. Neuvo, Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation. IEEE Trans. Signal Process. 39(2), 395–410 (1991)
20. S.J. Ko, Y.H. Lee, Center weighted median filters and their applications to image enhancement. IEEE Trans. Circuits Syst. 38(9), 984–993 (2001)
21. T. Chen, H.R. Wu, Adaptive impulse detection using center weighted median filters. IEEE Signal Process. Lett. 8(1), 1–3 (2001).
22. Takis Kasparis, Nicolas S. Tzannes, and Qing Chen, "Detail- Preserving Adaptive Conditional Filters," Journal of Electronic Imaging, vol 1, no. 4, pp. 358-364, Oct. 1992.
23. T. Sun and Y. Neuvo, "Detail-Preserving Median Based Filters in Image Processing," Pattern Recognition Lett. , vol. 15, no. 4, pp. 341–347, April 1994.
24. Eduardo Abreu, Michael Lightstone, Sanjit K. Mitra, and Kaoru Arakawa, "A New Efficient Approach for the Removal of Impulse Noise From Highly Corrupted Images," IEEE transaction on Image Processing, vol. 5, no. 6, pp. 1012-1025, June 1996.
25. Aizenberg and C. Butakoff, "Effective Impulse Detector Based on Rank-Order Criteria," IEEE Signal Processing Letters, vol. 11, no. 3, pp. 363-366, Mar. 2004.
26. H. Yu, L. Zhao, and H. Wang, "An Efficient Procedure for Removing Random-Valued Impulse Noise in Images," IEEE Signal Processing Letters, vol. 15, pp. 922-925, 2008.
27. Y. Dong and S. Xu, "A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise," IEEE Signal Processing Letters, vol. 14, no. 3, pp. 193-196, Mar. 2007.
28. T. Matsubara, V.G. Moshnyaga, and K. Hashimoto, "A FPGA Implementation of Low-Complexity Noise Removal," Proc. 17th IEEE Int'l Conf. Electronics, Circuits, and Systems (ICECS '10), pp. 255-258, Dec. 2010.
29. P.-Y. Chen, C.-Y. Lien, and H.-M. Chuang, "A Low-Cost VLSI Implementation for Efficient Removal of Impulse Noise," IEEE Trans. Very Large Scale Integration Systems, vol. 18, no. 3, pp. 473-481, Mar. 2010.