FPGA implementation of RSEPD technique based impulse noise removal

RAJADURAI.M
Saranathan college of engineering
Thiruchirapalli , India
rajadrajasce@gmail.com

Dr.M.SANTHI
Saranathan college of engineering
Thiruchirapalli , India

***Abstract*-In the process of signals transmission and acquisition, image signals might be corrupted by impulse noise. Generally, digital images are corrupted by impulse noises. These are short duration noises, which degrade an image and are randomly distributed over the image. An efficient FPGA implementation for removing impulse noise in an image is presented in this paper. Existing techniques use standard median filter. These existing approaches changes the pixel values of both noise less and noisy pixels, so image might be blurred in nature. To avoid the changes on noise less pixels, an efficient FPGA implementation of Simple Edge Preserved De-noising technique (SEPD) and Reduced Simple Edge Preserved De-noising technique (RSEPD) are presented in this paper. In this technique, noise detection and noise removal operations are performed. This VLSI design gives better image quality. For 10 percentage noise added image, the obtained PSNR value of the image is 29.22 while de-noising it.**

***Keywords*—SEPD, RSEPD, Noise, De-noising, FPGA.**

I. INTRODUCTION

Image processing is a form of image based signal processing and the processing based on pixels. Monochromatic image has either weakest intensity (black) (or) strongest intensity (white). Impulse noise corruption is very common in digital image. These are short duration noises, which degrade an image and are randomly distributed over the image. It occurs during image acquisition due to switching and occurs during image transmission due to interference of atmospheric disturbances. Impulse noise is always independent to the image pixels and is randomly distributed over the image. These are uncorrelated to the image pixels. For the impulse noise corrupted image, all the image pixels are not noisy, some number of pixels will be noised and some pixels will be noise free.

Two types of impulse noises are Salt and Pepper Noise (SPN), Random Valued Impulse Noise (RVIN). Salt and pepper noise appears dark pixel in bright region and bright pixel in dark region. It is otherwise called as spike noise. It appears white and black dots in the image and noisy pixels take graylevel of either 0 (or) 255. These noises are caused by sharp and sudden disturbances in the image signal; its appearance is white and black over the image. Random valued impulse noise is a complex one to detect the noise. Removal of Salt and pepper noise is easier compared to Random valued impulse noise.

Hwang and Haddad proposed [2] two median filter algorithms. First one is Ranked order based adaptive median filter (RAMF). It checks the centre pixel and neighbouring pixel is impulse corrupted or not. There are two level tests to check the impulse noised pixel. It is a simple technique to remove the positive and negative impulses. The second one is impulse size based adaptive median filter (SAMF). It changes the window length by determining the impulse size. It is better than Lin's scheme.

Zhang and Karim proposed [3] a new impulse detector for switching median filter. It detects the noisy pixels by finding the minimum absolute value of four convolutions, which is obtained by one dimensional Laplacian operator. Aizenberg and Butakoff proposed [4] Differential Rank impulse Detector (DRID). It compares rank and absolute value by using closest neighbour pixel difference.

Luo proposed [5] Efficient Removal of Impulse Noise (ERIN). It uses fuzzy based impulse detection, which is used to remove the impulse noise. It gives better performance in hardware processing. Srinivasan and Ebenesar [6] proposed Decision Based Algorithm (DBA). It removes the impulse noise corrupted pixels by finding its median value. All these existing approaches give blurred images due to changing the noisy as well as noise free pixels. Those techniques use median filter. It provides the median value of surrounding pixels.

II. PROPOSED SEPD TECHNIQUE

In those existing techniques noisy and noise free pixels are changed, that results in blur the image quality. To avoid the changes on noise less pixels, an efficient technique called as Simple Edge Preserved De-noising technique (SEPD) is used. The various application areas are Medical imaging, Scanning images, Image segmentation, Face recognition, printing skills. VLSI implementation gives easy way to implement, reduced complexity, high speed of implementation, parallel execution. The

FPGA implementation cost based on the memory and complexity. Hence, less memory and few operations are necessary for a low cost de-noising implementation. Based on these two factors, the VLSI implementation of SEPD based noise removal gives the efficient way of impulse noise removal.

SEPD is composed of three components Extreme data detector, Edge-oriented noise filter, Impulse arbiter. SEPD has low complexity. It consists only two line buffers, so its cost is low. The storage space needed for SEPD is two line buffer rather than full frame buffer. Only simple addition and subtraction operations are used in SEPD. Fig.1 shows the general block diagram for SEPD. Here the image input is given to the register bank, and then the reconstructed output pixel value is obtained from the impulse arbiter block. The extreme data detector is used to detect noisy pixel from the whole pixel value list of given input image. The edge oriented noise filter removes the noise from the detected pixel. Finally the impulse arbiter is used to reconstruct the pixel value by comparing with its threshold value.

A. Line buffer

Line buffer flushes the value in each new line. SEPD adopts only 3x3 masks.

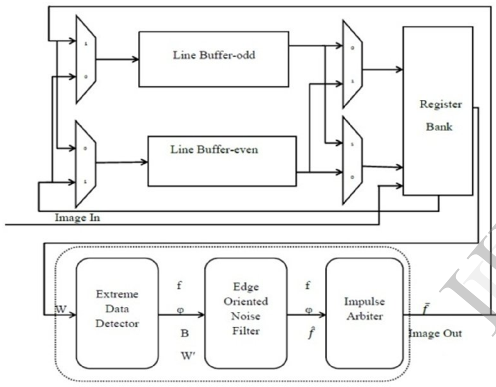


Fig.1. SEPD architecture

Assumed that the current pixel $p_{i,j}$ is located at coordinate (i,j) to be de-noised. The $f_{i,j}$ denotes the luminance value of before de-noised and the luminance value of after the de-noising process are represented as $\bar{f}_{i,j}$.

B. Register bank

Register bank (RB) is used to store the 3x3 sized mask pixel values. It uses 9 registers to store the pixel value. The 9 values are subsequently used by SEPD. Here Reg4 stores the centre pixel value. Here the each 3 serial registers provide three pixel values of a row in W. After $f_{i+1,j+1}$ entered to RB, SEPD starts de-noising process for $p_{i,j}$. Fig..3 shows the register bank architecture in SEPD technique. When $f_{i,j}$ is de-noised the reconstructed impulse arbiter output $\bar{f}_{i,j}$ is written back to the $f_{i,j}$ position.

Multiplexers are used to detect the odd or even rows. The selection signals are 1 or 0 based on odd or even rows respectively. The row3 of the RB gets the input pixel values from the image. The impulse arbiter output of current pixel and the previous values for Reg8 are written back to line buffers.

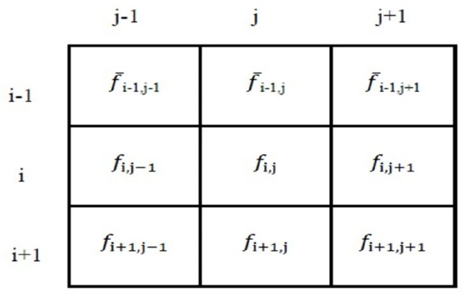


Fig.2 3x3 mask (W) centred on $p_{i,j}$

Fig.2 shows the 3x3 input mask (W) centred on $p_{i,j}$ and its luminance value is $f_{i,j}$. The pixels at odd and even rows are stored in Line buffer-odd and line buffer-even, respectively.

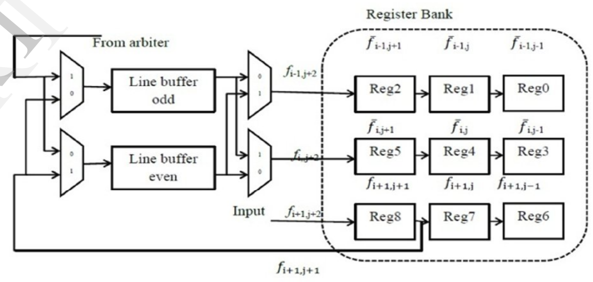


Fig.3. Register bank architecture

C. Extreme data detector

Any pixel is noisy means, its grayscale luminance value is maximum (or) minimum value. The extreme data detector detects the maximum and minimum ($MIN_{in}W$ and $MAX_{in}W$) luminance values in that mask from first one to the current one in the image.

If the value of the pixel ($f_{i,j}$) is maximum (or) minimum luminance value ($MIN_{in}W$ (or) $MAX_{in}W$), a variable φ is assumed to one and then checked its five neighbouring pixel values are corrupted (or) not. That is checked five neighbouring pixels are equal to the extreme data ($MIN_{in}W$ (or) $MAX_{in}W$), and then the five neighbouring pixels compared results are stored in B. Fig.4 shows the architecture of extreme data detector.

The extreme data detector block consists of Min-Max tree, Equality comparator (EC), OR gates. Here W denotes the

3x3 mask input value for appropriate input, which is received from the register bank. The Min-Max tree is used to find the minimum and maximum ($MIN_{in}W$ and $MAX_{in}W$) luminance values in the grayscale. W′ represents the eight neighbouring pixel values of the current pixel ($f_{i,j}$) in the 3x3 mask W.

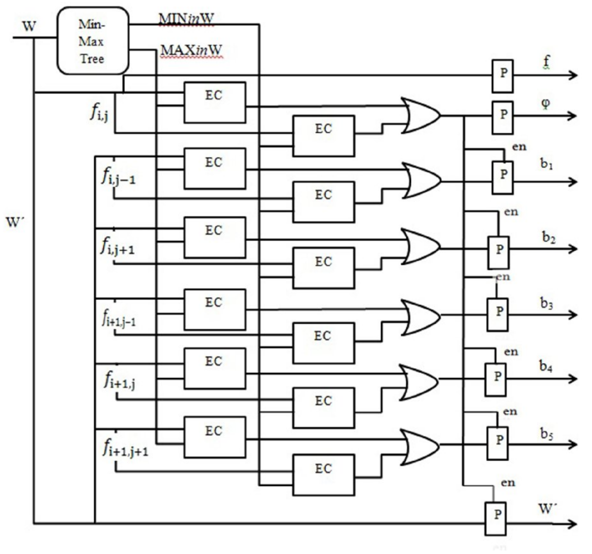


Fig.4. Extreme data detector architecture

Here the Equality Comparator (EC) produces logic1, when two input are equal otherwise its output is logic 0. The two columns of equality comparator units are used to combine the neighbouring pixel values are corrupted by noise (or) not. OR gates provide the binary comparisons. It compares maximum (or) minimum value and the elements of W mask. The binary compared results are stored in $b_1$to b_5.

D. Edge oriented noise filter

The edge oriented noise filter selects the edge and provides the de-noised pixel value of noise corrupted pixel. The architecture of edge oriented noise filter block is shown in Fig..5.

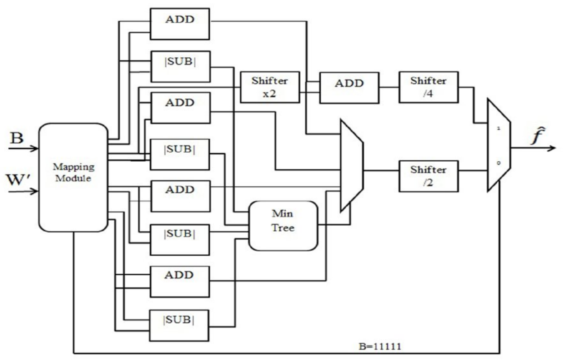
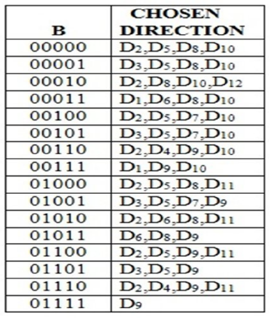
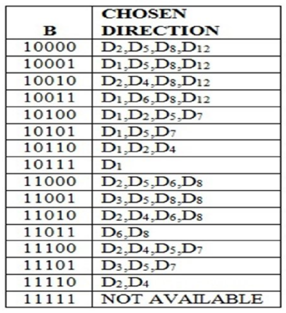


Fig.5. Architecture of edge oriented noise filter

Here B and W′ are the two inputs of edge oriented noise filter. B denotes the binary stored comparison value and W′ denotes the eight neighbouring pixel values of the current pixel ($f_{i,j}$) in the 3x3 mask W. The mapping module implements the corresponding direction from the binary stored values. The table of binary stored value and corresponding chosen directions are shown in fig.6. The |SUB| unit provides the absolute value of difference of two inputs. Here the 12 possible directions are used to find the corresponding directional differences. The directional lines are drawn by, the two ends are placed opposite (or) approximately opposite, in which the line must passed through the centre pixel ($f_{i,j}$) in the 3x3 mask.



B	CHOSEN DIRECTION
00000	D_2, D_5, D_8, D_{10}
00001	D_3, D_5, D_8, D_{10}
00010	D_2, D_8, D_{10}, D_{12}
00011	D_1, D_6, D_8, D_{10}
00100	D_2, D_5, D_7, D_{10}
00101	D_3, D_5, D_7, D_{10}
00110	D_2, D_4, D_9, D_{10}
00111	D_1, D_9, D_{10}
01000	D_2, D_5, D_8, D_{11}
01001	D_3, D_5, D_7, D_9
01010	D_2, D_6, D_8, D_{11}
01011	D_6, D_8, D_9
01100	D_2, D_5, D_9, D_{11}
01101	D_3, D_5, D_9
01110	D_2, D_4, D_9, D_{11}
01111	D_9



B	CHOSEN DIRECTION
10000	D_2, D_5, D_8, D_{12}
10001	D_1, D_5, D_8, D_{12}
10010	D_2, D_4, D_8, D_{12}
10011	D_1, D_6, D_8, D_{12}
10100	D_1, D_2, D_5, D_7
10101	D_1, D_5, D_7
10110	D_1, D_2, D_4
10111	D_1
11000	D_2, D_5, D_6, D_8
11001	D_3, D_5, D_8, D_8
11010	D_2, D_4, D_6, D_8
11011	D_6, D_8
11100	D_2, D_4, D_5, D_7
11101	D_3, D_5, D_7
11110	D_2, D_4
11111	NOT AVAILABLE

Fig.6.Binary values and selected directions

The 12 directions are shown as fig.7. The mapping module provides the four chosen noise free directions from the binary stored value (B). If a bit of B value equal to 1 denotes noisy pixel. To reduce the misdetection, the directions passed through the suspected pixels are discarded. Those four directions are chosen according to the variation in angle. The directions are chosen by selecting in which the directional lines are ended with noise free pixels. Four |SUB| units provide the four directions. The Min tree provides the smallest value from those four directional difference values. The ADD unit provides the addition of two appropriate inputs. The four ADD units provide the four addition values of the corresponding eight inputs. The multiplexer selects the corresponding luminance addition values of the minimum direction value. The shifter provides the average value of the selected direction. If all the five neighbouring pixels are corrupted by noise (binary stored value, B=11111) means the output is

$$\hat{f}_{i,j} = (\bar{f}_{i-1,j-1} + 2x\, \bar{f}_{i-1,j} + \bar{f}_{i-1,j+1}) / 4$$

E. Impulse arbiter

Impulse arbiter provides the proper and final result of the SEPD algorithm. When a pixel is impulse noise corrupted, its intensity value will jump to Salt (255) (or) Pepper (0). Here that condition is not true. In some other condition a pixel intensity value is Salt (or) Pepper ($MIN_{in}W$ (or) $MAX_{in}W$) might be told as a noisy pixel even it is not noise corrupted. To overcome this drawback, the impulse arbiter is used to avoid the wrong detection.

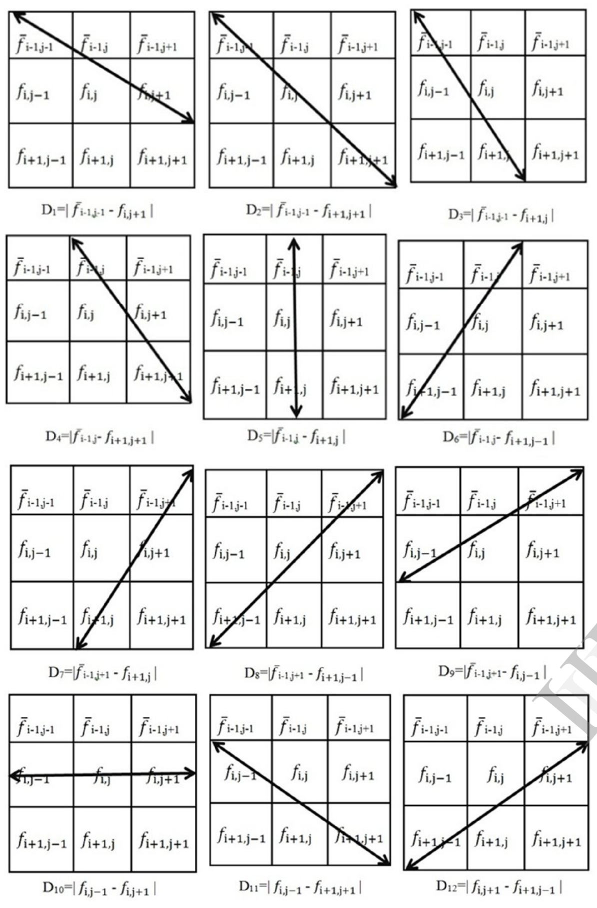

$D_1=|\bar{f}_{i-1,j-1} - f_{i,j+1}|$
$D_2=|\bar{f}_{i-1,j-1} - f_{i+1,j+1}|$
$D_3=|\bar{f}_{i-1,j-1} - f_{i+1,j}|$
$D_4=|\bar{f}_{i-1,j} - f_{i+1,j+1}|$
$D_5=|\bar{f}_{i-1,j} - f_{i+1,j}|$
$D_6=|\bar{f}_{i-1,j} - f_{i+1,j-1}|$
$D_7=|\bar{f}_{i-1,j+1} - f_{i+1,j}|$
$D_8=|\bar{f}_{i-1,j+1} - f_{i+1,j-1}|$
$D_9=|\bar{f}_{i-1,j+1} - f_{i,j-1}|$
$D_{10}=|f_{i,j-1} - f_{i,j+1}|$
$D_{11}=|f_{i,j-1} - f_{i+1,j+1}|$
$D_{12}=|f_{i,j+1} - f_{i+1,j-1}|$

Fig.7 Twelve directional lines of SEPD

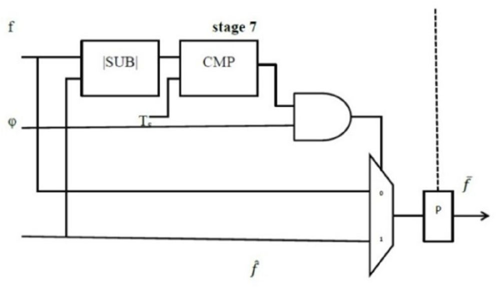


Fig.8 Architecture of impulse arbiter

Considered as a pixel ($f_{i,j}$) is noise free but its luminance value is maximum (or) minimum ($MIN_{in}W$ (or) $MAX_{in}W$). It can concluded that noise free when | $f_{i,j}$ - $\hat{f}_{i,j}$ | is small. Otherwise can be concluded that is noisy. This condition can be checked by using a threshold value Ts. The threshold value is assumed as 20. Those difference values are compared to the threshold value and then concluded that the pixel is noiseless (or) noise corrupted. The single stage impulse arbiter is shown as Fig.8. The |SUB| unit provides the difference between current pixel and edge oriented noise filter output. CMP is the comparator unit. If | $f_{i,j}$ - $\hat{f}_{i,j}$ | is greater than T_s means The comparator unit provides logic 1. Then this can be considered as noisy pixel. If it is less than T_s means noise free pixel. The AND gate provides the selection signal of φ with comparator output. The multiplexer unit decides whether the pixel is noise corrupted (or) noiseless based on the comparator selection signal.

III.**I**MPLEMENTATION **A**ND **R**ESULTS

An 8 bit 52x52 sized lena image is taken. Then the Salt and Pepper noise (impulse noise) with various noise densities is added to the input image. Fig.9 shows the various percentage of impulse noise added image and its corresponding de-noised image. The noise detection and filtering is processed using SEPD algorithm. The noise corrupted pixels are detected by comparing minimum, maximum grayscale luminance values with the each elements of the 3x3 pixel mask.

The filter replaces the noise corrupted pixel value into the reconstructed value on the centre pixel of the input mask. The filtering is performed by finding the minimum direction of the 12 directional differences of the 3x3 mask. The filter output is then compared to the threshold (T_s=0) and it provides the final output. The PSNR value of the de-noised image and its graphical representation are shown as fig.10.

IV.**R**SEPD

In SEPD method there are 12 edge lines are needed to de-noise the noised pixel. It contains more edges. So it may lead to computational complexity. To reduce complexity and cost Reduced SEPD technique is used. In RSEPD method instead of 12 edge lines, only 3 edge lines are used. The three edge lines D_a,D_b and D_c are shown in the figure.11

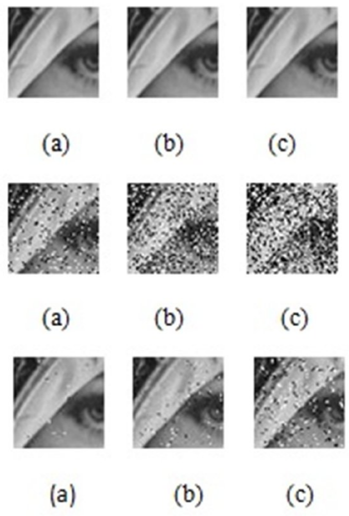
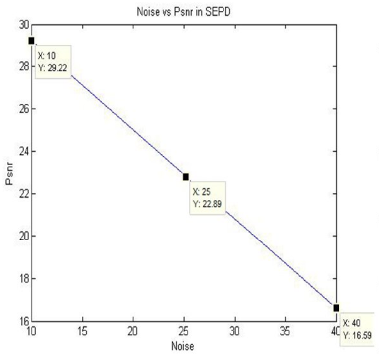
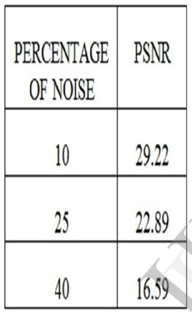


Fig.9 (a) original, 10% noise added image and de-noised image.(b) original, 25% noise added image and de-noised image.(c)original, 40% noise added image and de-noised image





PERCENTAGE OF NOISE	PSNR
10	29.22
25	22.89
40	16.59

Fig.10 PSNR values of various percentage of noise added image

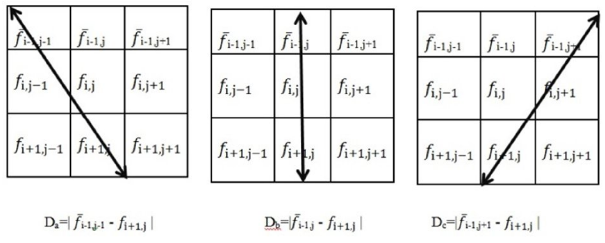


Fig.11 Three edge lines of RSEPD

RSEPD is composed of five main blocks Register bank, line buffer, noise detector, edge-based noise filter and impulse arbiter. The extreme data detector is shown as fig.12

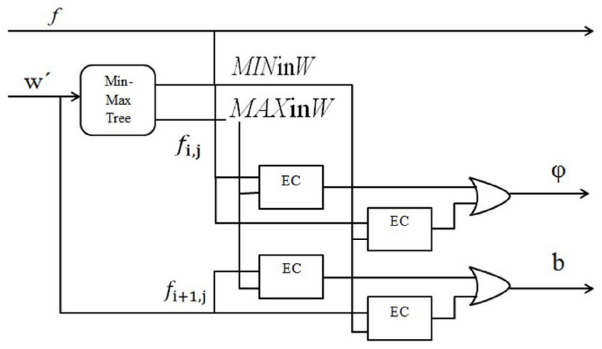


Fig.12. Extreme data detector architecture

Here instead of five neighbouring pixel only one neighbouring pixel ($f_{i+1,j}$) is considered for noise detection and filtering operation. The edge-oriented noise filter is shown as fig.13

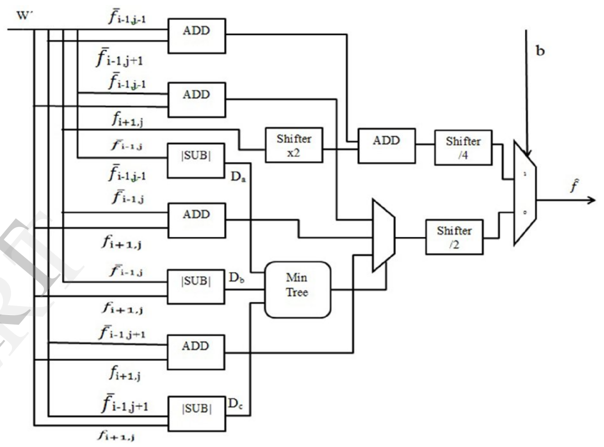


Fig.13.Edge oriented noise filter

Fig.14 shows the various percentage of impulse noise added image and its corresponding de-noised image. The noise detection and filtering is processed using SEPD algorithm. The noise corrupted pixels are detected by comparing minimum, maximum grayscale luminance values with the each elements of the 3x3 pixel mask. The filter de-noises the noised pixel value into the reconstructed value on the centre pixel of the input mask. The filtering is performed by finding the minimum direction of the 3 directional differences of the 3x3 mask. The filter output is then compared to the threshold (T_s=0) and it provides the final output.

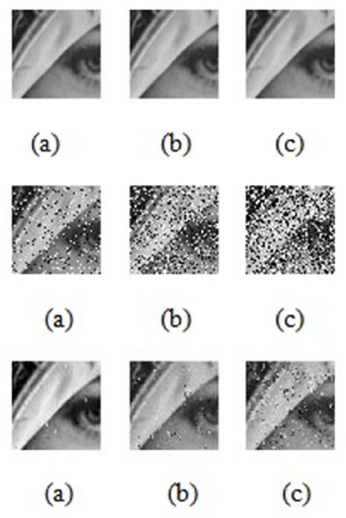
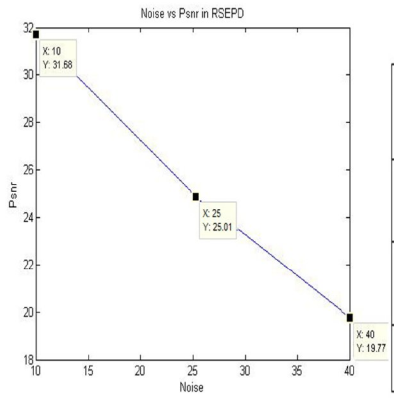
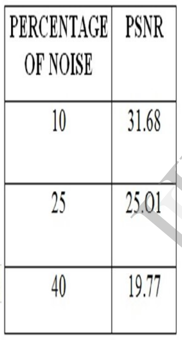


Fig.14 (a) original, 10% noise added image and de-noised image. (b) original, 25% noise added image and de-noised image. (c) original, 40% noise added image and de-noised image

The PSNR values of various de-noised image and its graphical representation are shown as fig.15





PERCENTAGE OF NOISE	PSNR
10	31.68
25	25.01
40	19.77

Fig.15 Graphical representation of PSNR values vs various percentage of noise

V. CONCLUSION

In this paper we proposed FPGA implementation for removal of impulse noise from an image. The FPGA implementation of Simple Edge Preserved De-noising technique (SEPD) and Reduced Simple Edge Preserved De-noising technique (RSEPD) gives better performance in terms quantitative evaluation and visual quality. Compared to SEPD technique RSEPD provides well PSNR values of various percentage of impulse noise with lowest hardware cost and lowest complexity.

REFERENCES

[1] W. K. Pratt, "Digital Image Processing" New York:Wiley-Inter- Science (1991).

[2] H. Hwang and R. Haddad (1995) "Adaptive median filters: New algorithms and results" IEEE Trans, Image process, Vol. 4, no.4.pp.499-502.

[3] S. Zhang and M. A. Karim, (2002) "A new impulse detector for switching mdian filter" IEEE signal process, Lett, Vol. 9, no.11,pp.341-347.

[4] I. Aizenberg and C. Butakoff, (2004) "Effective impulse detector based on rank order criteria", IEEE signal process, Lett, Vol. 11, no.3,pp.363-366.

[5] W. Luo, (2006) "Efficient removal of impulse noise from digital images", IEEE Trans, Consum, Electron, Vol. 52, no. 2, pp. 523-527.

[6] K. S. Srinivasan and D. Ebenezer, (2007), "A new fast and efficient decision based algorithm for removal og high density impulse noises", IEEE signal process, Lett. Vol. 14, no.3,pp. 189-192.

[7] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins, "Digital Image Processing", Gatesmark, LLC, (2009).

[8] Pei-Yen Chen, Chih-Yuan Lien and Hsu-Ming Chuang, (2010) "A low cost VLSI implementation for efficient removal of impulse noise".