



www.ijvdc.org

Removal of Impulse Noise in Images using Adaptive Decision Tree Based Image Denoising

S. VAISHNAVI¹, G. SRIDEVI²

¹PG Scholar, Dept of ECE(VLSID), Aditya Engineering College, Surampalem, India, E-mail:vaishnavi.ecm@gmail.com.

²Associate Professor & HOD, Dept of ECE, Aditya Engineering College, Surampalem, India, E-mail: sridevi_gamani@aec.edu.in.

Abstract: The main objective of this paper is to design an efficient architecture for removal of random-valued impulse noise from captured image. A good image denoising model property is that, it will remove noise while preserving edges. The decision-tree-based denoising method (DTBDM) is the proposed methodology. It consists of decision tree based impulse detector, to detect the noisy pixel and employs an effective design and to locate the edge by an edge preserving filter to reconstruct the intensity values of noisy pixels. The proposed technique can obtain better performance in terms of both qualitative and visual quality than the previous low complexity denoising methods. Further, it is enhanced using modified carry select adder in order to improve execution time. Here, serial adder is replaced with modified square root CSA to reduce computational time and area.

Keywords: Image Denoising, Impulse Noise, Impulse Detector, Carry Select Adder.

I. INTRODUCTION

Nowadays the image processing is playing a vital role in so many applications, such as in medical applications, printing and scanning programming, secret data communication and etc. mean while the images are infected with noises on the process of transmission and reception. The effected noise might be fade outs the image efficiency. Hence here it must require efficient denoising techniques. At present, in many practical applications, it is required to the use a good denoising technique. For this a low complexity denoising and enhancement techniques are essential and suitable for many VLSI architectures. In this paper, mainly concentrate on low complexity denoising technique for ease of implementation. Here the histogram equalization is used to improve image efficiency. Histogram equalization is a method in image processing of contrast adjustment using the image's histogram. This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. There are so many methods are used to reduce the noise by using decision tree. But only some can reduce efficiently. In general, images are frequently corrupted by impulse noise in the events of image acquisition and broadcast. The efficiency of the Image processing techniques mainly depends on the noise in the images. Hence, a competent denoising technique becomes a very important issue in image processing. In many of the applications, the image denoising is the basic technique for upcoming image processing operations, where as image segmentation, image enhancement, edge detection and processing, object extraction, etc. The main purpose to removal of noise is to reduce the noise by protecting image information. For this there are many different techniques are proposed for reducing the impulse noise. The goal of noise

removal is to suppress the noise while preserving image details. The median filter, one of famous techniques, which can able to restrain the noise. Median filter is used to change the neighborhood pixels value with median value. By this it can also change the image details and resulting a blurred image. In order to overcome the drawback in median filter and for increments in terms of efficiency the weighted median filter and the center-weighted median filters were proposed. Even though the above two filters can also preserve more image details than median filter, they are still implemented uniformly across the image without considering whether the current pixel is noise-free or not.

II. IMPULSE NOISE

Impulse noise is one of the different categories of noises which have an unwanted spikes and continuous pops. These noises are generally created by the effect of electromagnetic interference, effect of scratch on disks, and mismatches in digital communications. The high levels of noises are harmful for the inner human organs, there is even 180 dBs is enough to destroy the human ears. In order to improve the robustness in adaptive machineries, an impulse noise filter is necessary to improve noise signal quality. For achieving this, a typical filter named as median is used to delete impulse noise. In order to get better performing impulse noise filters, to use model-based systems that know the properties of the noise and source signal (in time or frequency), in order to remove only impulse obliterated samples.

A. Random-Valued Impulse Noise

The impulse noise is generated based on incorrect placement of pixels from sensors, hardware collisions, or using faulty transmitting channel. In this project, the digital color image is taken with interference of random valued impulse noise for

transmission. There are three components named as R, G, B are used for representing color image signal. The intersecting equation of image signal $X(i, j)$ and effected random valued impulse noise is shown in below:

$$X(i, j) = [x_R(i, j), x_G(i, j), x_B(i, j)]^T \quad (1)$$

$$X(i, j) = \begin{cases} s_k(i, j), & \text{probability } 1-p_k, \\ h_k, & \text{probability } p_k, \end{cases}$$

Where k is represented as R, G, B. the k^{th} component $s_k(i, j)$ is having 256 level values which is a 8-bit binary value and the noise probability is represented as $p_k(i, j)$. Here the noisy pixel value h_k , has the range between 0 to 255 with uniform distribution. This method of signal propagation is very famous and concentrated on transmitting bit errors. Though these three components are absolutely transmitting, there is no correlation with random valued impulse noise is occurred.

III. DENOISING TECHNIQUES

Image Denoising has remained a fundamental problem in the field of image processing. Various denoising techniques have been proposed so far and their application depends upon the type of image and noise present in the image. Image denoising is classified into two categories.

A. Spatial Domain Filtering

This is the traditional way to remove the noise from the digital images to employ the spatial filters. Spatial domain filtering is further classified into linear filters and non-linear filters.

1. Linear Filters

A mean filter is the optimal linear for Gaussian noise in the sense of mean square error. Linear filters tend to blur sharp edges, destroy lines and other fine details of image. It includes Mean filter and Wiener filter.

2. Non-Linear

With the non-linear filter, noise is removed without any attempts to explicitly identify it. Spatial filters employ a low pass filtering on the group of pixels with the assumption that noise occupies the higher region of frequency spectrum. Generally spatial filters remove the noise to reasonable extent but at the cost of blurring the images which in turn makes the edges in the picture invisible. It includes median and weighted median filter.

B. Transform Domain Filtering

The transform domain filtering is subdivided into data adaptive and non-adaptive filters. Transform domain mainly includes wavelet based filtering techniques.

1. Wavelet Transform

Due to properties like sparsity and multi resolution structure, Wavelet transform have become an attractive and efficient tool in image denoising. With Wavelet Transform gaining popularity in the last two decades various algorithms for denoising in Wavelet Domain were introduced.

IV. THE PROPOSED DTBDM

In this paper the random valued impulse noise is mainly used with uniform distribution has shown in [5]. Here, the 3×3 mask is used for de-noising an image. The de-noised pixels can be

placed at the coordinates (i, j) and it displayed as $p_{i,j}$ and having luminance value represented as $f_{i,j}$, shown in below figure (1). From the input sequence of image denoising process, the other eight pixel values of input image signal, divided into two forms: named as W_{TopHalf} and $W_{\text{BottomHalf}}$. They can be represented as

$$W_{\text{Top Half}} = \{a, b, c, d\} \quad (2)$$

$$W_{\text{Bottom Half}} = \{e, f, g, h\} \quad (3)$$

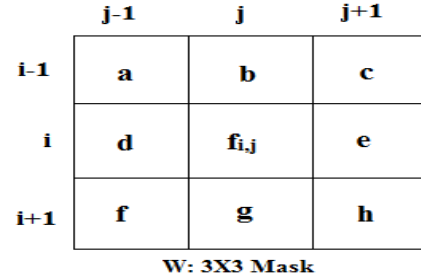


Fig.1. 3×3 Mask Centered on $p_{i,j}$.

DTBDM is having two basic elements, those are:

- decision tree based impulse detector and
- edge preserving image filter.

In order to perform correlation between pixels $p_{i,j}$ and its neighboring pixels, the detector can able to know whether it is noisy or not. If the result is positive, then only the edge preserving image filter based on direction-oriented filter generates the reconstructed value. Otherwise, it keeps the value would be unchanged.

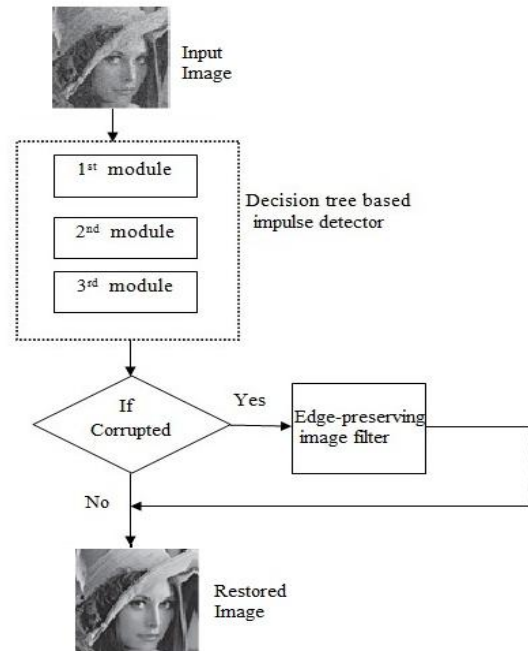


Fig.2. Dataflow of DTBDM

A. Decision Tree Based Impulse Detector

In order to determine whether it is a noisy pixel or not, have to perform the correlations between $p_{i,j}$ and its neighboring pixels. Surveying these methods, can simply classify them into several ways; observing the degree of isolation at current pixel,

Removal of Impulse Noise in Images Using Adaptive Decision Tree Based Image Denoising

determining whether the current pixel is on a fringe or comparing the similarities between current pixel and its neighboring pixels. For this reason, in our decision-tree-based impulse detector, we have design three modules. Those are isolation module (IM), fringe module (FM), and similarity module (SM) respectively. Three concatenating decisions of these modules are used to build a decision tree. The decision tree is a binary tree and determines the status of $p_{i,j}$ by the use of different equations in different modules. First of all we have to use an isolation module to decide whether the pixel value is in a smooth region or not. If there is negative response, we have to conclude that the current pixel belongs to noisy free. Otherwise, if there is positive response, it defines that the current pixel is may be a noisy pixel or just placed on an edge. The result is confirmed by the use of fringe module in the block. If the current pixel is placed on an edge, the result of fringe module would be negative (noisy free); otherwise, the result would be positive. If isolation module and fringe module could not determine whether current pixel is belongs to noisy free or noisy pixel, we have to use another module named as similarity module is used to producing the result. It compares the similarity between both current pixel and its neighboring pixels. If the result is positive, $p_{i,j}$ is a noisy pixel; otherwise, it could be a noise free pixel. The following sections are used to describe the three modules in detailed.

1. Isolation Module

The pixels with shadow suffering from noise have low similarity compared to the neighboring pixels and so-called as an isolation point. The difference between noisy pixel and its neighboring pixel value is large. According to the above concepts, first we have to find the maximum and minimum luminance values in W_{TopHalf} , which is named as TopHalf_max and TopHalf_min, and then calculate the difference between them, it was named as TopHalf_diff. For $W_{\text{BottomHalf}}$, we have to apply the same process to obtain BottomHalf_diff. The resulting two difference values are compared with a threshold value Th_{IMa} to find out whether the surrounding region belongs to a smooth area or not. The equations would be generated as follows:

$$\text{TopHalf_diff} = \text{TopHalf_max} - \text{TopHalf_min} \quad (3)$$

$$\text{BottomHalf_diff} = \text{BottomHalf_max} - \text{BottomHalf_min}. \quad (4)$$

$$\text{DecisionI} = \begin{cases} \text{True,} & \text{if } (\text{TopHalf_diff} \geq Th_{IMa}) \\ & \text{or } (\text{BottomHalf_diff} \geq Th_{IMa}) \\ \text{False,} & \text{otherwise} \end{cases} \quad (5)$$

$$\text{IM_TopHalf} = \begin{cases} \text{True,} & \text{if } (|f_{i,j} - \text{TopHalf_max}| \geq Th_{IMb}) \\ & \text{or } (|f_{i,j} - \text{TopHalf_min}| \geq Th_{IMb}) \\ \text{False,} & \text{otherwise} \end{cases} \quad (6)$$

$$\text{IM_BottomHalf} = \begin{cases} \text{True,} & \text{if } (|f_{i,j} - \text{TopHalf_max}| \geq Th_{IMb}) \\ & \text{Or } (|f_{i,j} - \text{TopHalf_min}| \geq Th_{IMb}) \\ \text{False,} & \text{otherwise} \end{cases} \quad (7)$$

$$\text{DecisionII} = \begin{cases} \text{True,} & \text{if } (\text{IMTopHalf} = \text{true}) \\ & \text{or } (\text{IMBottomHalf} = \text{true}) \\ \text{False,} & \text{otherwise} \end{cases} \quad (8)$$

2. Fringe Module

It is difficult to find out whether the pixel is noisy or situated on an edge. In order to generate the result with this case, we have to define four directions as from E1 to E4,

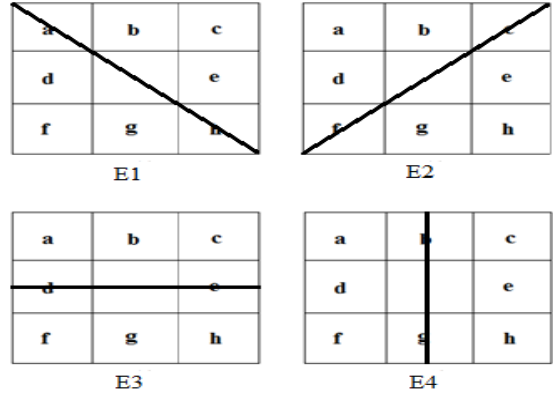


Fig.3. Four Directional Difference of Mask

We have to take direction E1 as an example in the above. In order to calculating the absolute difference between $f_{i,j}$ and the other two pixel values along the same direction, respectively, we have to determine whether there is an edge or not. The detailed equations are shown as follows

$$\text{FM_E1} = \begin{cases} \text{False,} & \text{If } (a-f_{i,j} \geq Th_{FMa}) \\ & \text{Or } (|h-f_{i,j}| \geq Th_{FMa}) \\ & \text{Or } (|a-h| \geq Th_{FMb}) \\ \text{True,} & \text{otherwise} \end{cases} \quad (9)$$

$$\text{FM_E2} = \begin{cases} \text{False,} & \text{If } (c-f_{i,j} \geq Th_{FMa}) \\ & \text{Or } (|f-f_{i,j}| \geq Th_{FMa}) \\ & \text{Or } (|c-f| \geq Th_{FMb}) \\ \text{True,} & \text{otherwise} \end{cases} \quad (10)$$

$$\text{FM_E3} = \begin{cases} \text{False,} & \text{If } (b-f_{i,j} \geq Th_{FMa}) \\ & \text{Or } (|g-f_{i,j}| \geq Th_{FMa}) \\ & \text{Or } (|b-g| \geq Th_{FMb}) \\ \text{True,} & \text{otherwise} \end{cases} \quad (11)$$

$$\text{FM_E4} = \begin{cases} \text{False,} & \text{If } (d-f_{i,j} \geq Th_{FMa}) \\ & \text{Or } (|e-f_{i,j}| \geq Th_{FMa}) \\ & \text{Or } (|d-e| \geq Th_{FMb}) \\ \text{True,} & \text{otherwise} \end{cases} \quad (12)$$

$$\text{Decision III} = \begin{cases} \text{False,} & \text{If } (\text{FM_E1}) \text{ or } (\text{FM_E2}) \\ & \text{or } (\text{FM_E3}) \text{ or } (\text{FM_E4}) \\ \text{True,} & \text{otherwise} \end{cases} \quad (13)$$

3. Similarity Module

This is the last module in the block diagram. The luminance values in the mask W are located in a noisy-free area might be close. The median value is always located at the center of the variation series, while the impulse is usually located at one of its near ends of the pixel matrix. Hence, if there are extremely big or small values, then there is the possibility to production of noisy signals. According to consideration of this concept, we have moved to sort nine values in ascending order and take the fourth, fifth, and sixth values which are close to the median in

mask W . The fourth, fifth, and sixth values are represented as $4_{th}inW_{i,j}$, $Max_{i,j}$ and $Min_{i,j}$ as $Max_{i,j} = 6_{th}inW_{i,j} + ThSMa$

$$Min_{i,j} = 4_{th}inW_{i,j} - ThSMa \quad (14)$$

In order to show the status of pixel $p_{i,j}$, $Max_{i,j}$ and $Min_{i,j}$ are used and for more concurrent decision making, it requires changes

$$N_{max} = \begin{cases} Max_{i,j}, & \text{if } (Max_{i,j} \leq MedianInW_{i,j} + Th_SM_b) \\ MedianInW_{i,j} + Th_SM_b, & \text{otherwise} \end{cases} \quad (15)$$

$$N_{min} = \begin{cases} Min_{i,j}, & \text{if } (Min_{i,j} \leq MedianInW_{i,j} + Th_SM_b) \\ MedianInW_{i,j} + Th_SM_b, & \text{otherwise} \end{cases} \quad (16)$$

At lastly it have been concluded that $p_{i,j}$ is noisy pixel with respect to position of $f_{i,j}$ is not in between N_{max} and N_{min} . The edge preserving image filter will not have been supposed to construct the repeated value. Here the output value is $f_{i,j}$ and the equation would be

$$Decision\ IV = \begin{cases} True, & \text{if } (f_{i,j} \geq N_{max}) \\ & \text{or } (f_{i,j} \leq N_{min}) \\ False, & \text{otherwise} \end{cases} \quad (17)$$

Obviously, the threshold affects the quality of de-noised images of the proposed method. A more appropriate threshold contributes to achieve a better detection result. However, it is not easy to derive an optimal threshold through analytic formulation. The fixed values of thresholds make our algorithm simple and suitable for hardware implementation. According to our extensive experimental results, the thresholds $Th\ IM_a$, $Th\ IM_b$, $Th\ FM_a$, $Th\ FM_b$, $Th\ SM_a$, and $Th\ SM_b$ are all predefined values and set as 20, 25, 40, 80, 15, and 60, respectively.

B. Edge-Preserving Image Filter

To locate the edge existing in the current W , a simple edge preserving technique which can be realized easily with VLSI circuit is adopted. Only those composed of noise-free pixels are taken into account to avoid possible misdetection. Directions passing through the suspected pixels are discarded to reduce misdetection. Therefore, we use $Max_{i,j}$ and $Min_{i,j}$, defined in similarity module, to determine whether the values of d , e , f , g , and h are likely corrupted, respectively. If the pixel is likely being corrupted by noise, we don't consider the direction including the suspected pixel. In the second block, if d , e , f , g , and h are all suspected to be noisy pixels, and no edge can be processed, so $f_{i,j}$ estimated value of $p_{i,j}$ is equal to the weighted average of luminance values of three previously denoised pixels and calculated as $(a + b \times 2 + c) / 2$. In other conditions, the edge filter calculates the directional differences of the chosen directions and locates the smallest one D_{min} among them in the third block. The equations are as follows:

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

$$D_8 = |c - f| \times 2 \quad (18)$$

$$F_{i,j} = \begin{cases} (a + d + e + h) / 4 & \text{if } D_{min} = D_1, \\ (a + b + g + h) / 4 & \text{if } D_{min} = D_2, \\ (b + g) / 2 & \text{if } D_{min} = D_3, \\ (b + c + f + g) / 4 & \text{if } D_{min} = D_4, \\ (c + d + e + f) / 4 & \text{if } D_{min} = D_5, \\ (d + e) / 2 & \text{if } D_{min} = D_6, \\ (a + h) / 2 & \text{if } D_{min} = D_7, \\ (c + f) / 2 & \text{if } D_{min} = D_8. \end{cases} \quad (19)$$

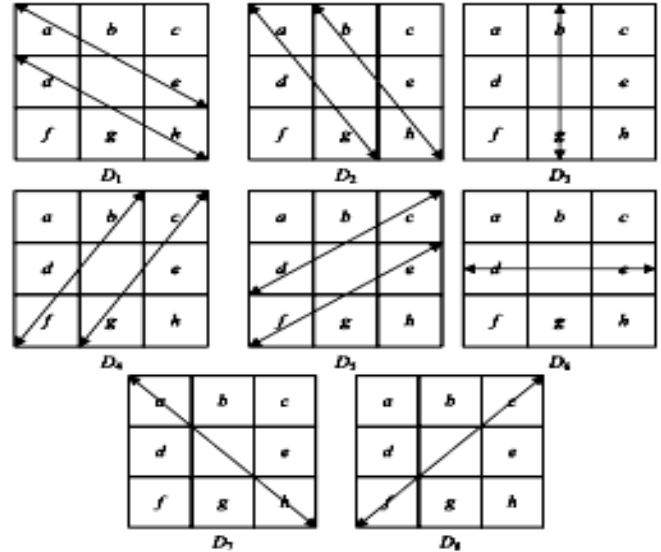


Fig.4. Eight Directional Differences of DTBDM.

As studied in the previous module of Fig. 4, the closest special relationship with pixel properties $p_{i,j}$ produced proper edge in the direction of it. Here the luminance values with mean of pixels generates the directional difference as $f_{i,j}$. whenever the difference $f_{i,j}$ gives correct edge, then it could be located at median b , d , e , and g . if it shows wrong edge, then the median values are written in the place of difference value $f_{i,j}$. It can showed in equation as

$$F_{i,j} = Median(f_{i,j}, b, d, e, g) \quad (20)$$

V. VLSI IMPLEMENTATION OF DTBDM

The DTBDM is taking only two line buffers for all images to generate low latency in computation and reduces the cost of the system in VLSI platform. In order to improve timing property, the pipelined architecture is introduced with clock.

A. Isolation Module

In the Above fig, comparator CMP_L is used to compare two input signals and separate the large value. Like that the comparator CMP_S generates small value. The first two-level comparators are used to generate $TopHalf_max$ and $TopHalf_min$. After that the result fed to the SUB unit for perform the subtraction operation in between them. The |SUB| block is used to produces absolute value from difference. The block GC is further compare the two outputs from previous comparators and results the output 1 if and only if the upper value is greater than the lower value. Then OR gate produces binary value for $IM_TopHalf$. Finally, the multiplexer confirm

Removal of Impulse Noise in Images Using Adaptive Decision Tree Based Image Denoising

the result and put out Decision II is positive. It resulting the pixel $p_{i,j}$ would have noise. The next module (FM) will be used to confirm the result.

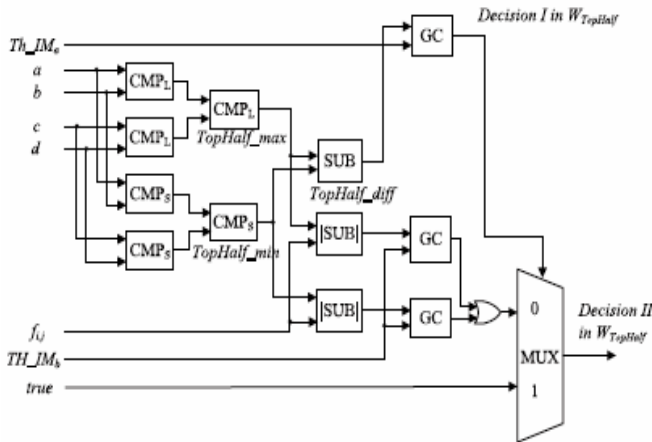


Fig.5. Architecture of IM

B. Fringe Module

The Fringe Module (FM) is the combination of four modules named as FM 1, FM 2, FM 3 and FM 4, and each of them is used to determine its directions. The below block diagram shows the detailed explanation of it. Here E1 shows direction of pixel values. The three |SUB| units are used to list out the resulting differences between the blocks.

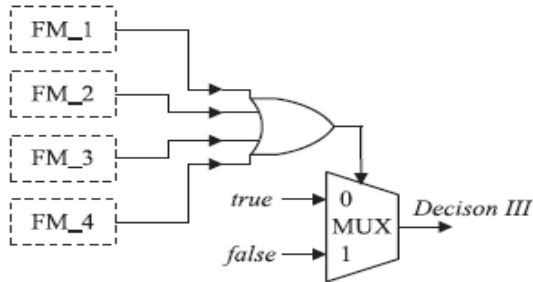


Fig.6. Architecture of FM.

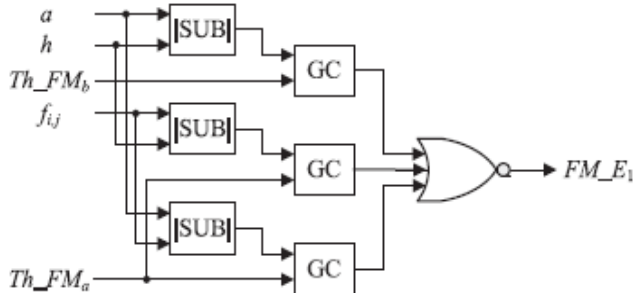


Fig.7. Architecture of FM 1 Module.

C. Similarity Module

Assuming that, if ISOLATION MODULE (IM) and FRINDGE MODULE (FM) cannot determine the noise free value or not in $f_{i,j}$, in that situation the SIMILARITY MODULE (SM) is required to produces the output. The architecture of SM is built with a mask w as fourth value for accelerating speed of sorting. The complete process of operation of module M0 is described as... whenever the value of a is greater than b , then C01 is going to high (1) otherwise it will be low (0). The eight number of GC values are defined as from C01 to C08. The

combine unit is used to combine the resulting comparing values from every comparator to generate numbers between 0 and 8. These numbers are used to specify the values order in mask W .

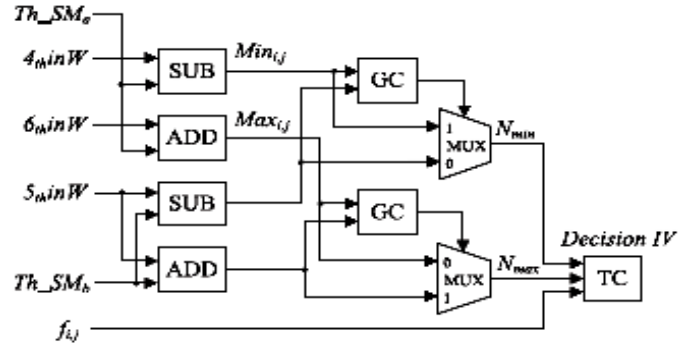


Fig.8. Architecture of SM

For suppose the value of a is small in mask W , then the output is written as 0 in mask $M0$ and when the value of a is biggest in mask W , then the out will be 8. Likewise the values are written for remaining outputs with only slighter difference. By the use of above method the order of GC is developed and the result of fringe module E1 is produced by NOR gate. If the result is positive, we consider that $f_{i,j}$ is on the edge E1 and regards it as noise free.

D. Edge-Preserving Image Filter

The Edge Preserving Image Filter is composed of two major blocks: one is minED Generator block and another one is Average Generator block. The minED generator can able to generate edge value which have minuend change. These changes are measured by using four ADD, twelve |SUB| and four numbers of shifter blocks. Here the minuend value is calculated with Min Tree unit and it was formed by comparator series. The second block Average Generator produces pixel luminance values based on simple directional difference function (D_{min}). According to the above said, $p_{i,j-1}$, $p_{i,j+1}$, $p_{i+1,j-1}$, $p_{i+1,j}$ and $p_{i+1,j+1}$ pixel positions are having noise then only the multiplexer unit produces an output as $(a+b \times 2 + c)/4$. Otherwise, it generates an output based on its mean value using the function D_{min} . Some directional differences are determined according to four pixel values, so its reconstructive values also need four pixel values.

E. Modified Carry Select Adder

The general 'carry select adder' is having a multiplexer and two ripple carry adders. In that, two ripple carry adders are used to add the two inputs having bit length 'n'. Therefore the addition is performed twice by the use of two ripple carry adders, one time with carry high and another time carry is low. Finally the multiplexer in the block used to generates the resulting output with sum and carry bits with reference to the carry bit applied at input. The Modified CSLA architecture has been developed using Binary to Excess -1 converter (BEC). This paper proposes an efficient method which replaces a BEC using common Boolean logic. The result analysis shows that the proposed architecture achieves the three folded advantages in terms of area, delay and power. The parallel RCA with $C_{in}=1$ (high) is replaced with BEC.

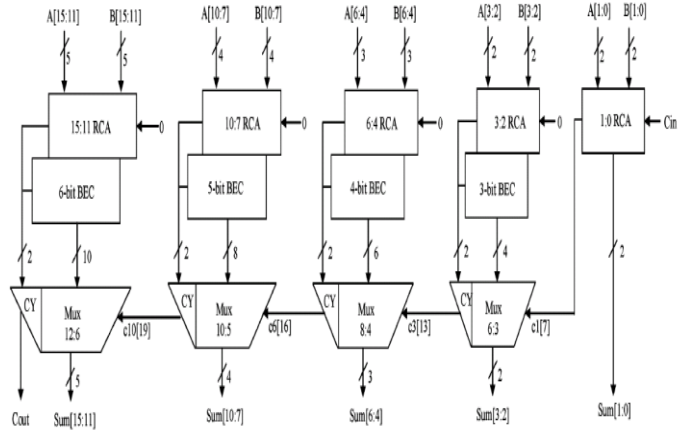


Fig.9. Block Diagram of 16-Bit Modified Square Root Carry Select Adder

1. BEC:

The main idea is to use BEC instead of the RCA with $C_{in}=1$, in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1bit BEC is required. Fig.10 illustrates how the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. One input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial results in parallel and the mux is used to select either the BEC output or the direct inputs according to the control signal C_{in} .

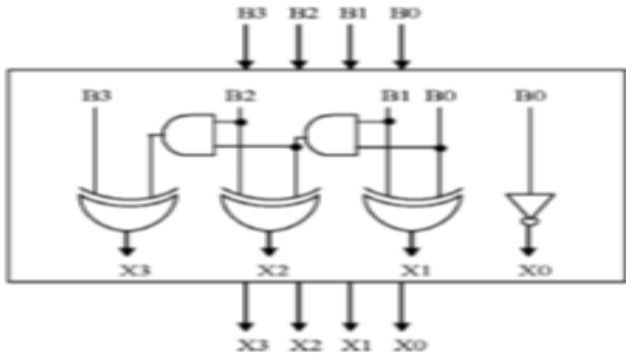


Fig.10. 4-Bit BEC Logic Diagram

The importance of the BEC logic stems from the large silicon area reduction when the CSLA with large number of bits are designed. The Boolean expressions of the 4-bit BEC is listed as (note the functional symbols NOT, AND, XOR).

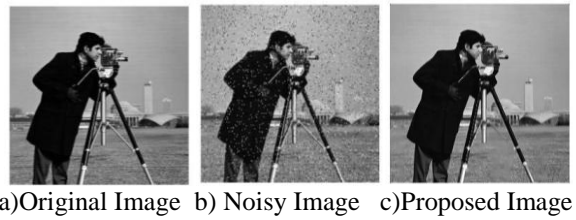
$$\begin{aligned}
 X0 &= \sim B0; \\
 X1 &= B0 \wedge B1; \\
 X2 &= B2 \wedge (B0 \& B1); \\
 X3 &= B3 \wedge (B0 \& B1 \& B2)
 \end{aligned}$$

VI. RESULT

A. Simulation Results

The characteristics and performance of the denoising method is verified by taking lena image as the test image. Consider the test image lena and by applying impulse noise of varying in MATLAB Environment. The digital grey scale image taken here, cannot process in VLSI directly. So, the image is converted to its corresponding pixel values(text format) and is fed to the denoising process. The proposed decision tree based

impulse detector and edge preserving filter based de-noising Method in VLSI is designed using VHDL. MODELSIM is used for the simulation. we employed the peak signal-to-noise ratio (PSNR) to illustrate the quantitative quality of restored images of various methods. The simulation result is shown in below figure.



a)Original Image b) Noisy Image c)Proposed Image



a)Original Image b) Noisy Image c)Proposed Image



a)Original Image b) Noisy Image c)Proposed Image

Fig.11. Results of Different Methods in Restoring 5 Percent Corrupted Images

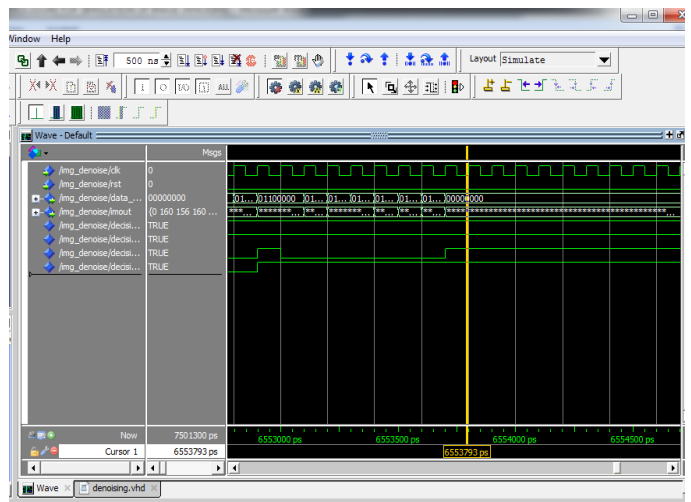


Fig.12. Simulation Waveform of DTBDM

B. Synthesis Result

TABLE I. Comparison Results

METHOD	Time (ns)	Area (no. of gates)
EXISTING (serial adder)	22.73	166
PROPOSED (modified CSA)	13.84	127

Removal of Impulse Noise in Images Using Adaptive Decision Tree Based Image Denoising

TABLE II. Comparative Results in PSNR (Db) of Images Corrupted by 5 Percent Impulses

Images \ Method	Cameraman	Peppers	Lena
Median	25.63	25.80	31.05
Weighted median	24.40	24.09	25.5
Rank order	28.07	28.93	29.74
Dtbdm	27.26	37.62	35.83

VII. CONCLUSION

VLSI architecture for efficient removal of impulse noise is proposed. The proposal is using decision tree based image detector to identify the noise values in pixels and improves image efficiency by adopting a powerful design. The resulting image quality will be improved by this adequate methodology. The above results showing the performance of proposed method is more excellent in terms of both quantitative evaluation and visual clarity of image. Final architecture consists of denoising and Image enhancement. This method is applied for different types of noises and compare results of image corrupted by impulses and equalized image using peak signal to noise ratio, mean square error, structural content shows that this method is effective for all types of noises. By using the modified square root CSA, the execution time, computational speed and area is reduced. It's suitable to be applied to many real time applications. Proposed system can applicable in color images and can consider different types of noises for further study.

VIII. REFERENCES

[1]R.C. Gonzalez and R.E. Woods, Digital Image Processing. Pearson Education, 2007.

[2]W.K. Pratt, Digital Image Processing. Wiley-Interscience, 1991.

[3]R.H. Chan, C.W. Ho, and M. Nikolova, "Salt-and-Pepper Noise Removal by Median-Type Noise Detectors and Detail-Preserving Regularization," IEEE Trans. Image Processing, vol. 14, no. 10, pp. 1479-1485, Oct. 2005.

[4]S. Zhang and M.A. Karim, "A New Impulse Detector for Switching Median Filter," IEEE Signal Processing Letters, vol. 9, no. 11, pp. 360-363, Nov. 2002.

[5]P.-Y. Chen and C.-Y. Lien, "An Efficient Edge-Preserving Algorithm for Removal of Salt-and-Pepper Noise," IEEE Signal Processing Letters, vol. 15, pp. 833-836, Dec. 2008.

[6]T. Nodds and N. Gallagher, "Median Filters: Some Modifications and Their Properties," IEEE Trans. Acoustics, Speech, Signal Processing, vol. ASSP-30, no. 5, pp. 739-746, Oct. 1982.

[7]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.

[8]I. Andreadis and G. Louverdis, "Real-Time Adaptive Image Impulse Noise Suppression," IEEE Trans. Instrumentation and Measurement, vol. 53, no. 3, pp. 798-806, June 2004.

[9]N.I. Petrovic and V. Crnojevic, "Universal Impulse Noise Filter Based on Genetic Programming," IEEE Trans. Image Processing, vol. 17, no. 7, pp. 1109-1120, July 2008.

[10]S. Rasoul Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," IEEE Trans. Systems Man, Cybernetics, vol. 21, no. 3, pp 660-674, May 1991.

[11]I. Aizenberg, C. Butakoff, and D. Paliy, "Impulsive Noise Removal Using Threshold Boolean Filtering Based on the Impulse Detecting Functions," IEEE Signal Processing Letters, vol. 12, no. 1, pp. 63-66, Jan. 2005.

[12]I. 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.

[13]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.

[14]T. Chen and H.R. Wu, "Adaptive Impulse Detection Using Center-Weighted Median Filters," IEEE Signal Processing Letters, vol. 8, no. 1, pp. 1-3, Jan. 2001.

[15]Z. Xu, H.R. Wu, B. Qiu, and X. Yu, "Geometric Features-Based Filtering for Suppression of Impulse Noise in Color Images," IEEE Trans. Image Processing, vol. 18, no. 8, pp. 1742-1759, Aug. 2009.

[16]A.S. Awad and H. Man, "High Performance Detection Filter for Impulse Noise Removal in Images," IEEE Electronic Letters, vol. 44, no. 3, pp. 192-194, Jan. 2008.

[17]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.

[18]T. Y. Ceiang and M. J. Hsiao, "Carry-select adder using single ripple carry adder," *Electron. Lett.*, vol. 34, no. 22, pp. 2101-2103, Oct. 1998.

[19]Y. He, C. H. Chang, and J. Gu, "An area efficient 64-bit square root carry-select adder for lowpower applications," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2005, vol. 4, pp. 4082-4085.

[20]O. J. Bedrij, "Carry-select adder," *IRE Trans. Electron. Comput.*, pp.340-344, 1962.

Author's Profile:



Sukkireddy Vaishnavi received B.Tech degree in Electronics and Computers Engineering from Usha Rama College of Engineering and Technology, Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India in 2013 and she is working towards her M.Tech from Aditya Engineering College, Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India. Her area of interest is in VLSI Design, Digital image processing.



Gamini Sridevi received B.Tech degree in Electronics and Communication Engineering from Nagarjuna University, Andhra Pradesh, India and Masters degree from Jawaharlal Nehru Technological University, Kakinada, Andhra Pradesh, India in 2000 and 2009 respectively. She is currently pursuing her Ph.D in Jawaharlal Nehru

S. VAISHNAVI, G. SRIDEVI

Technological University, Kakinada. Her areas of interest are Digital image processing and Digital signal processing. She has more than 14 years of teaching experience. She is presently working as an Associate professor in the department of Electronics and Communication Engineering in Aditya Engineering College, Surampalem, Andhra Pradesh. She is the member of Institution of Electronics and Telecommunication Engineers.