International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:7909-15

AN EFFICIENT DENOISING ARCHITECTURE FOR REDUCING OF IMPULSE NOISE IN IMAGES

1. A.SUSWARA, 2. T.VENU GOPAL
1. PG Scholar, Dept of ECE, Anubose Institute of Technology, Palwancha, khammam
2. Dept of ECE, Anubose Institute of Technology, Palwancha, khammam

ABSTRACT:

The main objective of this project is to design an efficient architecture for removal of random-valued impulse noise from captured image. The decision-tree-based detector to detect the noisy pixel and employs an effective design to locate the edge is proposed in this design. Decision-Tree-Based Impulse Detector is used as proposed methodology for this project. If there are noisy values, edges, or blocks in this region, the distribution of the values is different. Therefore, we determine whether current pixel is an isolation point by observing the smoothness of its surrounding pixels. The pixels with shadow suffering from noise have low similarity with the neighboring pixels and the so-called isolation point. The difference between it and its neighboring pixel value is large. Further, this project is enhanced using carry select adder in order to improve execution time. Here, serial adder is replaced with CSA to reduce computational time.

KEYWORDS: decision-tree, Impulse Noise, Pixel, Carry Select adder, Isolation, Salt and Pepper noise, Random value Impulse Noise

INTRODUCTION:

Image processing techniques is widely used in so many applications, such as medical imaging, scanning techniques, printing skills and so on. Images are corrupted by noises in the procedures of image transmission and reception. The noise may seriously affect the performance of images and reduce clarity[4]. Hence, an efficient denoising technique becomes a very important in image processing[1]-[8]. Today, in many practical applications, the denoising process is used, so a good lower-complexity denoising and enhancement technique, whichis simple and suitable for VLSI implementation is necessary. In this, focus only on the lower complexity denoising techniques[1]because of its simplicity and easy implementation with the VLSI techniques. Clarity of the image is being improved using histogram equalization. Histogram equalizationis 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 thehistogram. There have been several methods usingdecision tree to deal with noises and some of them perform well. Based on above basic concepts, decision-tree-based denoising and enhancement method and its VLSI architecture for removing noises is used. In general, images are frequently corrupted by impulse noise in the events of image acquisition andbroadcast. The efficiency of the Image processing techniques mainly depends on the noise in the images. Hence, a competentdenoising technique becomes a very important issue in image processing [1], [2]. In most applications, denoising the image is fundamental tosubsequentimage processing operations, such as edge detection, image segmentation, object recognition, etc.The goal of noise removal is to suppress the noise while preserving image details. To this end, a variety of techniques have been proposed to remove impulse noise. One of the most popular methods is the median filter [2], which can suppress noise with highcomputational efficiency. However, since every pixel in the image is replaced by the median value in its neighborhood, the median filter often removes desirable details in the image andblurs it too. The weighted median filter [3] and the center-weighted median filter [4] were proposed as remedy to improve the median filter by giving more weight to some selected pixels in the filtering window. Although these two filters can preserve more details than the median filter, they are still implemented uniformly across the image without considering whether the current pixel is noise-free or not.

International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:7909-15

IMPULSE NOISE:

Impulse noise is a category of (acoustic) noise which includes unwanted, almost instantaneous (thus impulse-like) sharp sounds (like clicks and pops). Noises of the kind are usually caused by electromagnetic interference, scratches on the recording disks, and ill synchronization in digital recording and communication. High levels of such a noise (200 + Decibels) may damage internal organs, while 180 Decibels are enough to destroy or damage human ears. An impulse noise filter can be used to enhance the quality of noisy signals, in order to achieve robustness in pattern recognition and adaptive control systems. A classic filter used to remove impulse noise is the median filter, at the expense of signal degradation. Thus it's quite common, 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.

RANDOM-VALUED IMPULSE NOISE:

Impulse noise is caused by malfunctioning pixels in camera sensors, faulty memory locations in hardware, or transmission in a noisy channel. In this paper, we consider a color image corrupted with the random-valued impulse noise in a transmission of the digitalized signal. A signal in the color image consists of R, G and B components. The signal X(i, j) corrupted with the random-valued impulse noise is represented by:

$$\boldsymbol{X}(i,j) = [x_{\mathrm{R}}(i,j), x_{\mathrm{G}}(i,j), x_{\mathrm{B}}(i,j)]^T,$$

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

where, k stands for R, G or B. sk(i, j) is the k-th component of the source signal, and takes 256-level (8 bit) values. pk(i, j) represents a noise occurrence probability. The noise-corrupted pixel value hk takes from 0 to 255 with a uniform distribution. This model is the most popular and focuses on a bit error in the digitalized signal transmission. When R, G and B components are transmitted independently, it can be assumed that the random-valued impulse noises have no correlation with other components for their occurring positions and the noise values.

DENOISING TECHNIQUES:

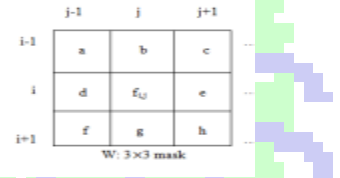
Image Denoising has remained a fundamental problem in the field of image processing. Due to properties like sparsity and multiresolution structure, Wavelet transform have become an attractive and efficient tool in image denoising. With Wavelet Transform gaining popularity in the last two decades various algorithms for denoising in Wavelet Domain were introduced.

THE PROPOSED DTBDM:

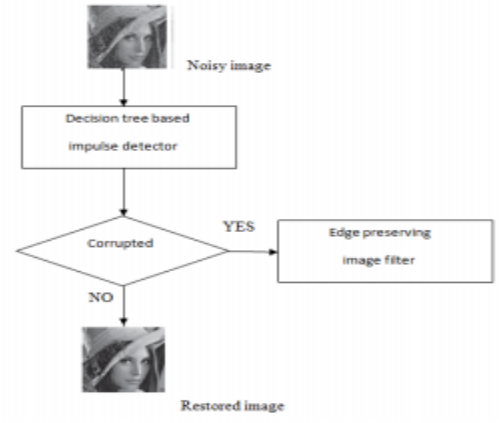
The noise considered in this paper is random-valued impulse noise with uniform distribution as practiced in [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Here, we adopt a 3 _ 3 mask for image denoising. Assume the pixel to be denoised is located at coordinate ði; jÞ and denoted as pi;j, and its luminance value is named as fi;j, as shown in Fig. 1. According to the input sequence of image denoising process, we can divide other eight pixel values into two sets: WTopHalf and WBottomHalf . They are given as

$$W_{TopHalf} = \{a, b, c, d\}. \quad (1)$$

$$W_{BottomHalf} = \{e, f, g, h\}. \quad (2)$$



DTBDM consists of two components: decision-tree-based impulse detector and edge-preserving image filter. The detector determines whether pi;j is a noisy pixel by using the decision tree and the correlation between pixel pi;j and its neighboring pixels. If the result is positive, edgepreserving image filter based on direction-oriented filter generates the reconstructed value. Otherwise, the value will be kept unchanged.j



International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:7909-15

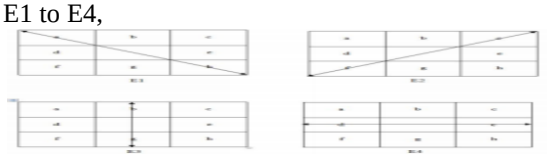
Decision Tree Based Impulse Detector In order to determine whether pi,j is a noisy pixel, the correlations between pi,j and its neighboring pixels are considered . Surveying these methods, we can simply classify them into several ways—observing the degree of isolation at current pixel , determining whether the current pixel is on a fringe or comparing the similarity between current pixel and its neighboring pixels Therefore, in our decision-tree-based impulse detector, we design three modules—isolation module (IM), fringe module (FM), and similarity module (SM). Three concatenating decisions of these modules build a decision tree. The decision tree is a binary tree and can determine the status of pi,j by using the different equations in different modules. First, we use isolation module to decide whether the pixel value is in a smooth region. If the result is negative, we conclude that the current pixel belongs to noisy free. Otherwise, if the result is positive, it means that the current pixel might be a noisy pixel or just situated on an edge. The fringe module is used to confirm the result. If the current pixel is situated on an edge, the result of fringe module will be negative (noisy free); otherwise, the result will be positive. If isolation module and fringe module cannot determine whether current pixel belongs to noisy free, the similarity module is used to decide the result. It compares the similarity between current pixel and its neighboring pixels. If the result is positive, pi,j is a noisy pixel; otherwise, it is noise free. The following sections describe the three modules in detail. Isolation Module: The pixels with shadow suffering from noise have low similarity with the neighboring pixels and the so-called isolation point. The difference between it and its neighboring pixel value is large. According to the above concepts, we first detect the maximum and minimum luminance values in WTopHalf, named as TopHalf_max, TopHalf_min, and calculate the difference between them, named as TopHalf_diff. For WBottomHalf , we apply the same idea to obtain BottomHalf_diff. The two difference values are compared with a threshold Th_IMa to decide whether the surrounding region belongs to a smooth area. The equations are as

$$IM\_TopHalf = \begin{cases} true, & \text{if}(|f_{i,j} - TopHalf\_max| \geq Th\_IM_b \\ & \text{or}(|f_{i,j} - TopHalf\_min| \geq Th\_IM_b \\ false, & \text{otherwise} \end{cases}$$

$$IM\_BottomHalf = \begin{cases} true, & \text{if}(|f_{i,j} - BottomHalf\_max| \geq Th\_IM_b \\ & \text{or}(|f_{i,j} - BottomHalf\_min| \geq Th\_IM_b \\ false, & \text{otherwise} \end{cases} \quad (3)$$

$$DecisionII = \begin{cases} true, & \text{if}(|IMTopHalf = true|) \\ & \text{or}(IMBottomHalf = true) \\ false, & \text{otherwises} \end{cases}$$

Fringe Module: How to conclude that a pixel is noisy or situated on an edge is difficult. In order to deal with this case, we define four directions, from E1 to E4,



Four directional difference of mask

We take direction E1 for example. By calculating the absolute difference between fi,j and the other two pixel values along the same direction, respectively, we can determine whether there is an edge or not. The detailed equations are as

$$FM\_E1 = \begin{cases} false, & \text{if}(|a - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|h - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|a - h| \geq Th\_FM_b) \\ true, & \text{otherwise} \end{cases}$$

$$FM\_E2 = \begin{cases} false, & \text{if}(|c - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|f - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|c - f| \geq Th\_FM_b) \\ true, & \text{otherwise} \end{cases}$$

$$FM\_E3 = \begin{cases} false, & \text{if}(|b - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|g - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|b - g| \geq Th\_FM_b) \\ true, & \text{otherwise} \end{cases}$$

$$FM\_E4 = \begin{cases} false, & \text{if}(|d - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|e - f_{i,j}| \geq Th\_FM_a) \\ & \text{or}(|d - e| \geq Th\_FM_b) \\ true, & \text{otherwise} \end{cases}$$

$$Decision\ III = \begin{cases} false, & \text{if}(FM\_E1)\text{or}(FM\_E2) \\ & \text{or}(FM\_E3)\text{or}(FM\_E1) \\ true, & \text{otherwise} \end{cases} \quad (4)$$

Similarity Module: The last module is similarity module. The luminance values in mask W located in a noisy-free area might be close. The median is always located in the center of the variational series, while the impulse is usually located near one of its ends. Hence, if there are extreme big or small values, that implies the possibility of noisy signals. According to this concept, we sort nine values in ascending order and obtain the fourth, fifth, and sixth values which are close to the median in mask W. The fourth, fifth, and sixth values are represented as

International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:7909-15

4thinWi,j, MedianInWi,j, and 6thinWi,j. We define Maxi,j and Mini,j as

Maxi,j=6thinWi,j+ThSMa

Mini,j=4thinWi,j-ThSMa (5)

Maxi,j and Mini,j are used to determine the status of pixel pi,j. However, in order to make the decision more precisely,we do some modifications as

$$N_{max}=\begin{cases} Max_{i,j}, & \text{if}(Max_{i,j} \le MedianInW_{i,j} + Th\_SM_b \\ MedianInW_{i,j} + Th\_SM_b & \text{otherwise} \end{cases}$$

$$N_{max}=\begin{cases} Min_{i,j}, & \text{if}(Min_{i,j} \ge MedianInW_{i,j} - Th_{SM_b} \\ MedianInW_{i,j} - Th\_SM_b & \text{otherwise} \end{cases} \quad (6)$$

Finally, if fi,j is not between Nmax and Nmin, we conclude that pi,j is a noise pixel. Edge-preserving image filter will be used to build the reconstructed value. Otherwise, the original value fi,j will be the output. The equation is as

$$\text{Decision IV}=\begin{cases} true, & if(f_{i,j} \ge N_{max}) or (f_{i,j} \le N_{min}) \\ false, & otherwise \end{cases} \quad (7)$$

Obviously, the threshold affects the quality of denoised 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 IMa, TH IMb, Th FMa, Th FMb, Th SMa, and Th SMb are all predefined values and set as 20, 25, 40, 80, 15, and 60, respectively.

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 Maxi,j and Mini,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 fi,j estimated value of pi,j is equal to the weighted average of luminance values of three previously denoised pixels and calculated as (a+ b× 2+ c)/ 2.In other conditions, the edge filter calculates the directional differences of the chosen directions and locates the smallest one Dmin among them in the third block. The equations are as follows:

$$\begin{aligned} D_1 &= |d-h| + |a-e| \\ D_2 &= |a-g| + |b-h| \\ D_3 &= |b-g| \times 2 \\ D_4 &= |b-f| + |c-g| \\ D_5 &= |c-d| + |e-f| \\ D_6 &= |d-e| \times 2 \\ D_7 &= |a-h| \times 2 \end{aligned} \quad (8)$$

$$f_{i,j}=\begin{cases} (d+h+a+e)/4 \\ (a+g+b+h)/4 \\ (b+g)/2 \\ \frac{b+f+c+g}{4} \\ \frac{c+d+e+f}{4} \\ \frac{d+e}{2} \\ (a+h)/2 \end{cases}$$

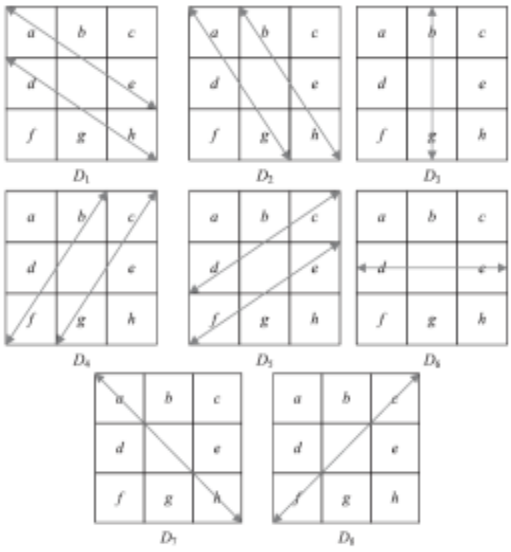


Fig 4. Eight directional differences of DTBDM.

In the last block of Fig. 4, the smallest directional difference implies that it has the strongest spatial relation with pi,j and probably there exists an edge in its direction. Hence, the mean of luminance values of the pixels which possess the smallest directional difference is treated as fi,j. After fi,j is determined, a tuning skill is used to filter the
bias edge. If fi,j obtain the correct edge, it will situate at the median of b, d, e, and g because of the spatial relation and the characteristic of edge preserving. Otherwise, the values of fi,j will be replaced by the median of four neighboring pixels (b, d, e, and g). We can express fi,j as

International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-256

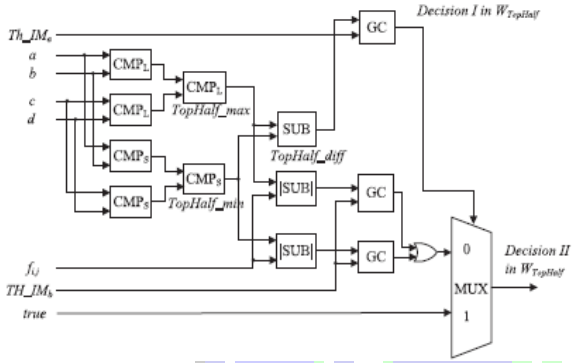
September 2016
VOLUME -2
ISSUE-5
Page:7909-15

Fi,j=Median(fi,j,b,d,e,g) (9)

VLSI Implementation of DTBDM:

DTBDM has low computational complexity and requires only two line buffers instead of full images, so its cost of VLSI implementation is low. For better timing performance, we adopt the pipelined architecture to produce an output at every clock cycle.

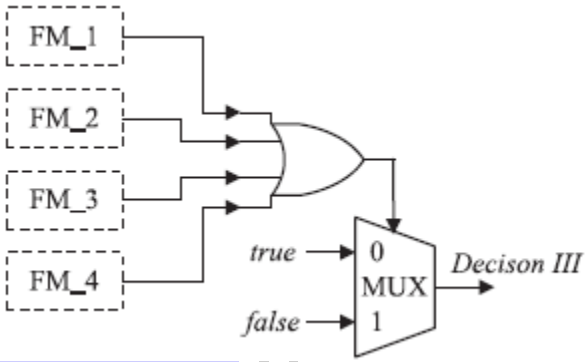
Isolation Module:



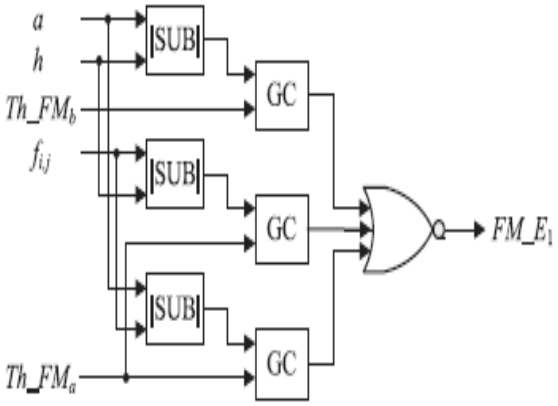
The comparator CMPL is used to output the larger value from the two input values while the comparator CMPS is used to output the smaller value from the two input values. The first two-level comparators are used to find TopHalf_max and TopHalf_min. The SUB unit is used to output the difference which is subtracted the lower input (TopHalf_min) from the upper one (TopHalf_max), and the jSUBj unit is used to output the absolute value of difference of two inputs. The GC is the greater comparator that will output logic 1 if the upper input value is greater than the lower one. The OR gate is employed to generate the binary result for IM_TopHalf. Finally, if the result of Decision II is positive, pi;j might be a noisy pixel or situate on an edge. The next module (FM) will be used to confirm the result

Fringe Module:

The FM is composed of four small modules, from FM_1 to FM_4, and each ofthem is used to determine its direction, as mentioned. is a detailed implementation of FM_1. Since E1 is the direction from a to h , the relation between a, h, and fi;j must be referenced. The three jSUBj units are used to determine the absolute differences between them.



Architecture of FM.



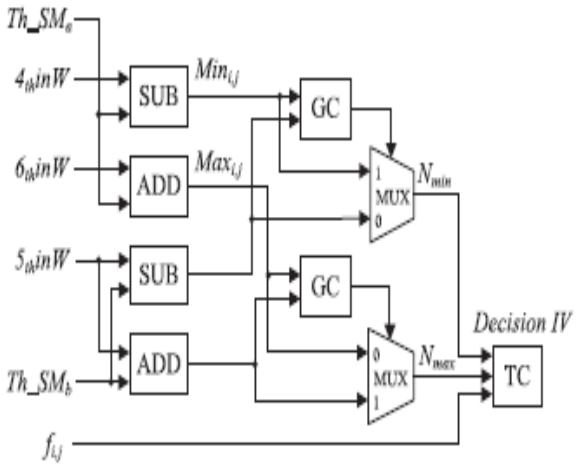
Architecture of FM 1 module.

SIMILARITY MODULE

If IM and FM can't determine whether fi;j belongs to a noisefree value or not, SM is used to confirm the result. This shows our architecture that is designed to accelerate the sorting speed to obtain the fourth value in mask W. The detailed implementation of module M0.. If a is greater than b, C01 is set to 1; otherwise, C01 is set to 0. The eight GC units are used to determine the values from C01 to C08. After comparing, a combined unit is used to combine the results of each comparator to obtain a number between 0 and 8. The number indicates the order of value in mask W. If a is the smallest value in mask W, the output of the M0 module is 0; if a is the biggest value in mask W, the output is 8. The architectures of other modules (M1 to M8) are almost the same as M0, with only little difference. By means of this implementation, we can find out the order of

International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-256

September 2016
VOLUME -2
ISSUE-5
Page:7909-15



The GC is described in the above section and the NOR gate is used to generate the result of FM E1. Ifthe result is positive, we consider that fi;j is on the edge E1 and regard it as noise free.

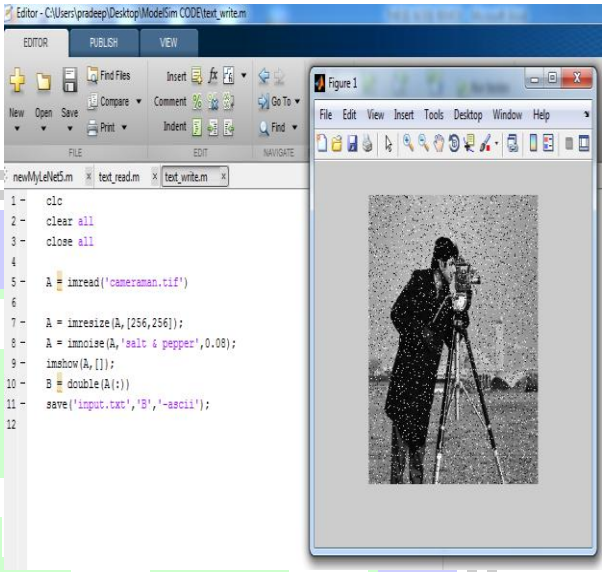
Edge-Preserving Image Filter:

The Edge-Preserving Image Filter is composed of two modules, minED generator and average generator (AG). the architecture of the minED generator which is used to determine the edge that has the smallest difference. Eight directional differences are calculated with twelve jSUBj, four ADD, and four shifter units. Then, the smallest one is determined by using the Min Tree unit. Min Tree is made up of a series of comparators. After that, the mean of luminance values of the pixels which process the smallest directional difference ðDminÞ can be obtained from the average generator. As mentioned in Section 2, if pi;j_1; pi;jþ1; piþ1;j_1; piþ1;j and piþ1;jþ1 are all suspected to be noisy pixels, the final MUX will output ða þ b _ 2 þ cÞ=4. Otherwise, the MUX will output the mean of the pixel values which process Dmin. Some directional differences are determined according to four pixel values, so its reconstructive values also need four pixel values.

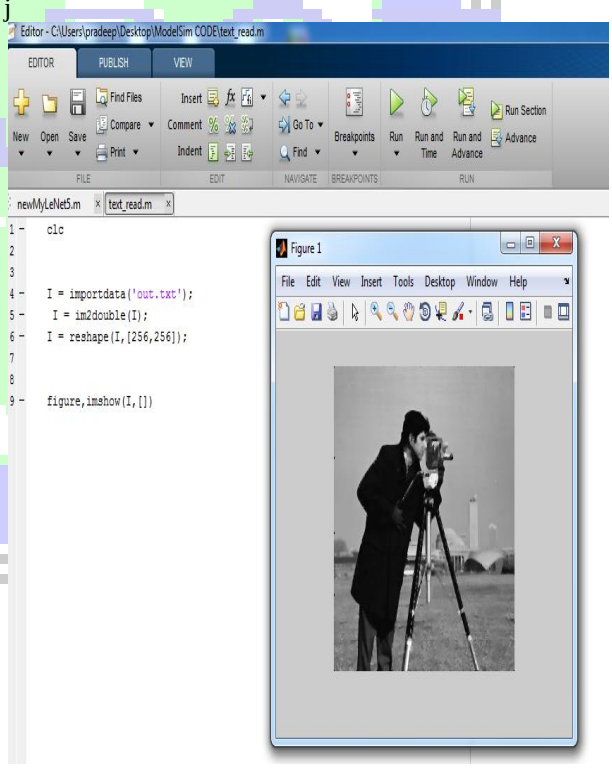
CARRY SELECT ADDER:

The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders (therefore two ripple carry adders) in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known.

RESULT:



j



CONCLUSION:

VLSI architecture for efficient removal of noises and image enhancement is proposed. The approach uses the decision-tree-based detector to detect the noisy

pixel and employs an effective design to increase the clarity of image. With adaptive method, the quality of the reconstructed images is improved. Results shows that the performance of proposed technique is better 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. 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

REFERENCES:

[1] Zhou Wang and David Zhang, "Progressive switching median filter for the Removal of impulse noise from highly corrupted images," IEEE Trans Circuits and Systems-II: Analog and Digital Signal Processing, vol. 46, no.1, Jan.1999.

[2] Hwang.H and Haddad.R.A, "Adaptive median filters: new algorithms and results," IEEE Trans. on Image Processing, vol.4, no.4, pp.499-502, 1995

[3] A. K. Jain, "Fundamentals of Digital Image Processing," Englewood Cliffs,NJ: Prentice-Hall, 1989.

[4] A. C. Bovik, T. S. Huang, and D. C. Munson, "Edge-sensitive image restoration using orderconstrained least squares methods," IEEE Trans. Acoust., Speech, Signal Processing, vol. 33, pp. 1253–1263, Oct.1985.

[5]T. A. Nodes and N. C. Gallagher Jr., "The output distribution of median type filters," IEEE Trans. Commun., vol. COM-32, pp. 532–541, May 1984.

[6] A.C.Bovik., "Streaking in median filtered images," IEEE Trans. Acoust.,Speech,Signal Processing, n., vol. 35, pp. 493-503, Oct.1985.

[7.] D. A. F. Florencio and R.W. Shafer, "Decision-based median filter using local signal statistics," Proc. SPIE Symp.in Visual Communication and Image Processing, vol. 2308, Sept. 1994, pp. 268–275.

[8] T. Chen and H.Wu, "Adaptive impulse detection using center-weighted median filters," Signal Processing Lett., vol. 8, no. 1, pp. 1–3, Jan. 2001.

[9] R. Yang, L. Lin, M. Gabbouj, J.Astola, and Y. Neuvo, "Optimal weighted median filters under structural constraints," IEEE Trans. Signal Processing, vol. 43, pp. 591–604, Mar. 1995.

[10] T. Song, M. Gabbouj, and Y. Neuvo, "Center weighted median filters: some properties and applications in image processing," Signal Processing, vol. 35, no. 3, pp. 213–229, 1994.

[11] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," IEEE Trans. Circuits Syst., vol. 38, pp.984–993, Sept.1991.