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

April 2016
VOLUME -1
ISSUE-4
Page:7899-06

REMOVAL OF IMPULSE NOISE IN IMAGES USING DECISION BASED IMAGE DENOISING

1. S. VAISHNAVI, 2.G.SRIDEVI

1. PG Scholar, Dept of ECE, Aditya Engineering College, JNTUK

2. Dept of ECE, Aditya Engineering College, JNTUK

ABSTRACT:

The main objective of this project is to design an efficient architecture for removal of random-valued impulse noise from captured image. A good image de-noising model property is that, it will remove noise while preserving edges. The decision-tree-based denoising method (DTBDM) is the proposed methodology for this project. 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, 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: Image denoising, Impulse Noise, Impulse Detector, Carry Select adder.

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 [4]. Hence here it must require efficient denoising techniques [1]-[8]. 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 project, mainly concentrate on low complexity denoising technique for easy 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 [1], [2]. In many of the applications, the image denoising is the basic technique for upcoming image processing operations, where as image segmentation, image enhancement, and 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 [2], 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 could generate resulting a blurred image. In order to overcome the drawback in median filter and for increments in terms of efficiency the weighted median filter [3] and the center-weighted median filters [4] 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.

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

April 2016
VOLUME -2
ISSUE-4
Page:7899-06

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.

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:

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

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

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.

DENOISING TECHNIQUES:

Image Denoising has remain a fundamental problem in the field of image processing. 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.

THE PROPOSED DTBDM:

In this paper the random valued impulse noise is mainly used with uniform distribution has shown in [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18]. Here, the 3×3 mask is used for denoiseing an image. The denoised 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 $W_{BottomHalf}$. They can represented as

$W_{Top\ Half}$ = {a, b, c, d}…………….. (1)
$W_{Bottom\ Half}$ = {e, f, g, h}………….. (2)

	j-1	j	j+1
i-1	**a**	**b**	**c**
i	**d**	$\mathbf{f_{i,j}}$	**e**
i+1	**f**	**g**	**h**

W: 3X3 Mask

Fig1:3×3 mask centered on $p_{i,j.}$

DTBDM is having two basic elements: those are a) decision tree based impulse detector and b) 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.

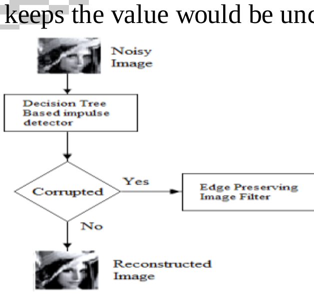


Fig2: Dataflow of DTBDM

International Journal of Engineering In Advanced Research Science and Technology
ISSN: 2278-256
April 2016
VOLUME -2
ISSUE-4
Page:7899-06

DECISION TREE BASED IMPULSE DETECTOR:

In order to determine whether is a noisy pixel, have to perform the correlations between $p_{i,j}$ and its neighboring pixels [10], [11], [14], [16], [17], [23], [24], [25], [26], [27],[28], [29], [30]. Surveying these methods, they can simply classify them into several ways; observing the degree of isolation at current pixel, [10], [11], [14], [16], [17], [23], [24],[25], determining whether the current pixel is on a fringe [16],[17], [26], [27] 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.

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 $W_{BottomHalf,}$ 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:

$$\text{TopHalf\_diff} = \text{TopHalf\_max} - \text{TopHalf\_min} \quad \text{…. (3)}$$

$$\text{BottomHalf\_diff} = \text{BottomHalf\_max} - \text{BottomHalf\_min}. \quad \text{…. (4)}$$

$$\text{DecisionI} = \begin{cases} \text{True,} & \text{if } (\text{TopHalf\_diff} \ge \text{Th\_IM}_a) \\ & \text{or } (\text{BottomHalf\_diff} \ge \text{Th\_IM}_a) \\ \text{False,} & \text{otherwise} \end{cases} \quad \text{…….. (5)}$$

$$\text{IM\_TopHalf} = \begin{cases} \text{True,} & \text{if } (|f_{i,j} - \text{TopHalf\_max}| \ge \text{Th\_IM}_b) \\ & \text{or } (|f_{i,j} - \text{TopHalf\_min}| \ge \text{Th\_IM}_b) \\ \text{False,} & \text{otherwise} \end{cases} \quad \text{…….. (6)}$$

$$\text{IM\_BottomHalf} = \begin{cases} \text{True,} & \text{if } (|f_{i,j} - \text{TopHalf\_max}| \ge \text{Th\_IM}_b \\ & \text{Or } (|f_{i,j}) - \text{TopHalf\_min}| \ge \text{Th\_IM}_b \\ \text{False,} & \text{otherwise} \end{cases} \quad \text{…….. (7)}$$

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

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,

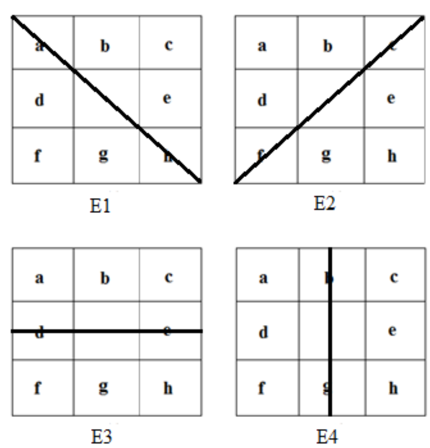


Fig3: Four directional difference of mask

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

$$\text{FM\_E1} = \begin{cases} \text{False,} & \text{If } (a - f_{i,j}| \ge \text{Th\_FM}_a) \\ & \text{Or } (|h - f_{i,j}| \ge \text{Th\_FM}_a) \\ & \text{Or } (|a - h| \ge \text{Th\_FM}_b) \end{cases}$$

$$\begin{array}{ll} \text{True,} & \text{otherwise} \end{array} \quad \text{..... (9)}$$

$$FM\_E2 = \begin{cases} \text{False,} & \text{If } (c-f_{i,j}| \geq Th\_FM_a) \\ & \text{Or } (|f-f_{i,j}| \geq Th\_FM_a) \\ & \text{Or } (|c-f| \geq Th\_FM_b) \\ \text{True,} & \text{otherwise} \end{cases} \quad \text{..... (10)}$$

$$FM\_E3 = \begin{cases} \text{False,} & \text{If } (b-f_{i,j}| \geq Th\_FM_a) \\ & \text{Or } (|g-f_{i,j}| \geq Th\_FM_a) \\ & \text{Or } (|b-g| \geq Th\_FM_b) \\ \text{True,} & \text{otherwise} \end{cases} \quad \text{...... (11)}$$

$$FM\_E4 = \begin{cases} \text{False,} & \text{If } (d-f_{i,j}| \geq Th\_FM_a) \\ & \text{Or } (|e-f_{i,j}| \geq Th\_FM_a) \\ & \text{Or } (|d-e| \geq Th\_FM_b) \\ \text{True,} & \text{otherwise} \end{cases} \quad \text{...... (12)}$$

$$\text{Decision III} = \begin{cases} \text{False,} & \text{If (FM\_E1) or (FM\_E2)} \\ & \text{or (FM\_E3) or (FM\_E1)} \\ \text{True,} & \text{otherwise} \end{cases} \quad \text{..... (13)}$$

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

$$Max_{i,j} = 6_{th}inW_{i,j} + ThSMa$$
$$Min_{i,j} = 4_{th}inW_{i,j} - ThSMa \quad \text{..... (14)}$$

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

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

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

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

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

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 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.

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× 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:

$$D_1 = |d - h| + |a - e|$$
$$D_2 = |a - g| + |b - h|$$
$$D_3 = |b - g| \times 2$$
$$D_4 = |b - f| + |c - g|$$

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

April 2016
VOLUME -2
ISSUE-4
Page:7899-06

$$D_5 = |c - d| + |e - f|$$
$$D_6 = |d - e| \times 2$$
$$D_7 = |a - h| \times 2$$
$$D_8 = |c - f| \times 2 \qquad \ldots. (18)$$

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

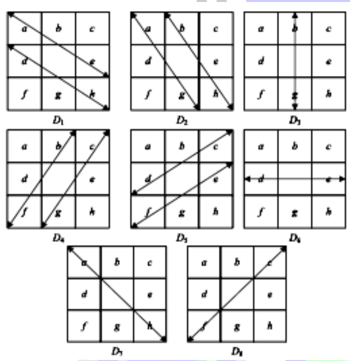


Fig 4. Eight directional differences of DTBDM.

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

$$F_{i,j} = \text{Median}\,(f_{i,j}, b, d, e, g) \qquad \ldots.. (19)$$

VLSI Implementation of DTBDM:

The DTBDM is taking only two line buffers in the place of total 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.

ISOLATION MODULE:

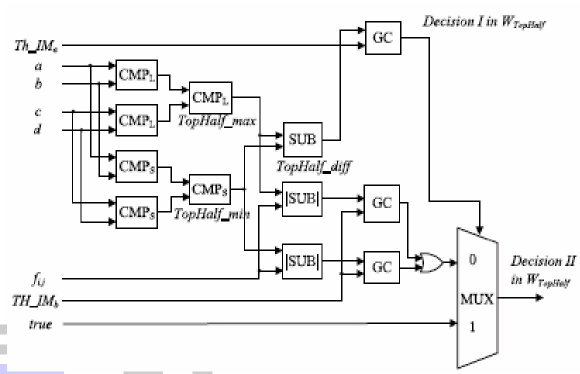


Fig5: Architecture of IM

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 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.

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.

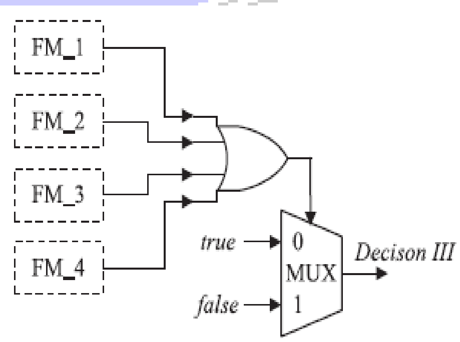


Fig6: Architecture of FM.

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

April 2016
VOLUME -2
ISSUE-4
Page:7899-06

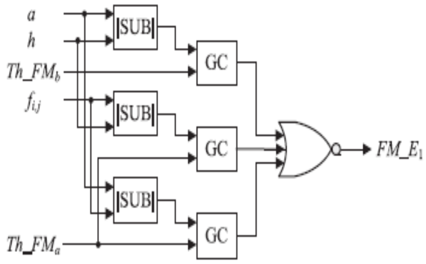


Fig7: Architecture of FM 1 module.

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.

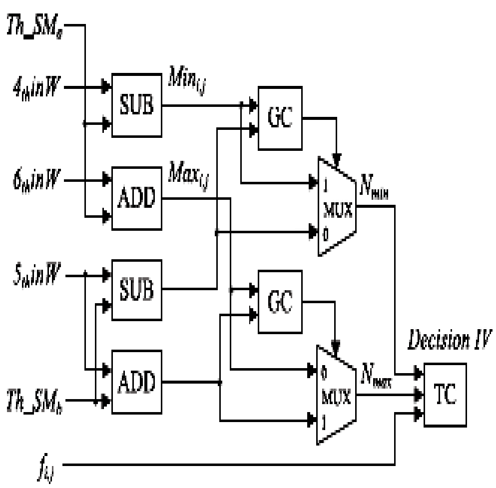


Fig8: Architecture of SM

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.

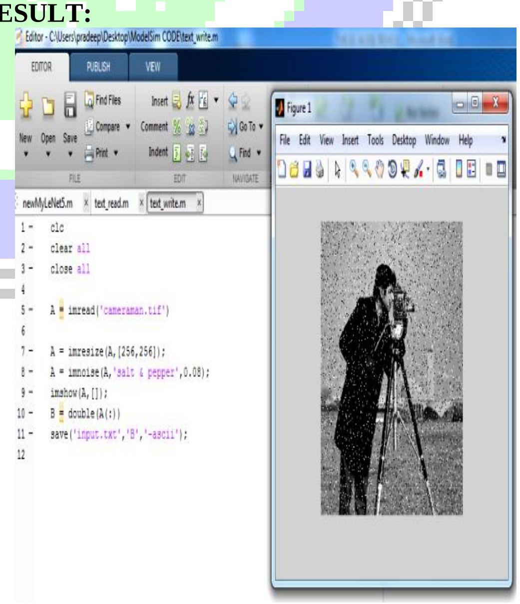
EDGE-PRESERVING IMAGE FILTER:

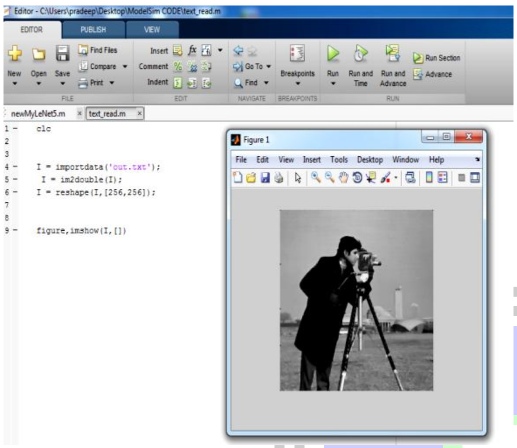
The Edge Preserving Image Filter is composed of two major blocks: one is mineED Generator block and another one is Average Generator block. The minED generator can able to generate edge value which have minuend change. These changes are measured by using four ADD, twelve |SUB| and four numbers of shifter 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 (Dmin). 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×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.

CARRY SELECT ADDER:

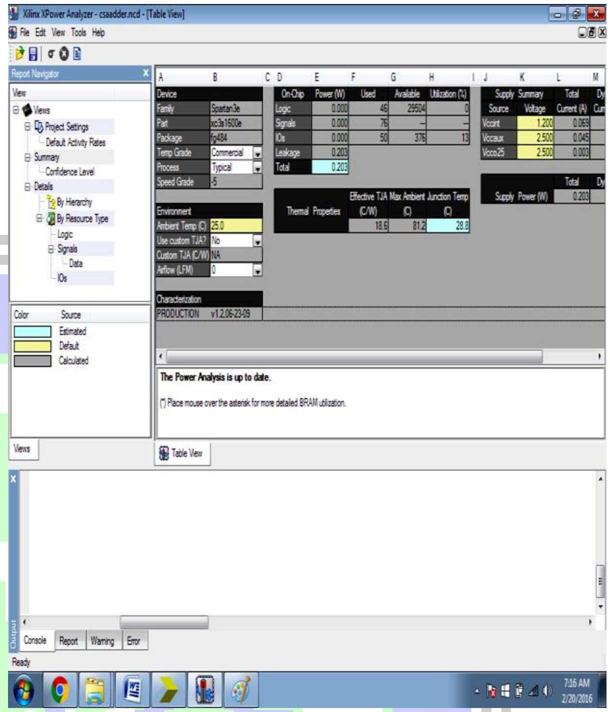
The carry select adder is having a multiplexer and two ripple carry adders. In that, two ripple carry adders are used to add the two inputs having bit length n. therefore the addition is performed twice by the use of two ripple carry adders, one time with carry high and another time carry is low. Finally 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.

RESULT:





PROPOSED:

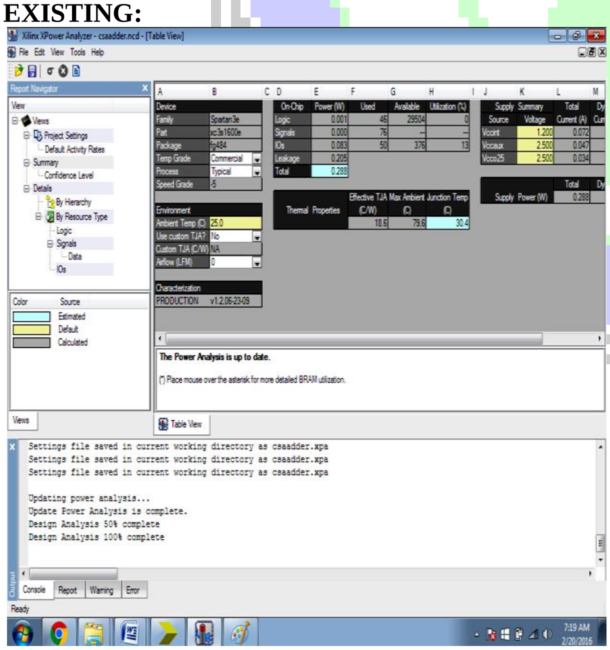


SYNTHESIS RESULT:

	TIME	AREA
EXISTING	14.744	166
PROPOSES	10.331	127

POWER REPORTS:

EXISTING:



CONCLUSION:

VLSI architecture for efficient removal of impulse noise and image enhancement 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 CSA the execution time and computational speed 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.

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.

[12] 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.

[13] T. Chen and H.R. Wu, "Space Variant Median Filters for the Restoration of Impulse Noise Corrupted Images," IEEE Trans. Circuits Systems II, Analog Digital Signal Processing, vol. 48, no. 8, pp. 784-789, Aug. 2001.

[14] W. Luo, "An Efficient Detail-Preserving Approach for Removing Impulse Noise in Images," IEEE Signal Processing Letters, vol. 13, no. 7, pp. 413-416, July 2006.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] B. De Ville, Decision Trees for Business Intelligence and Data Mining. SAS Publishing, 2007.

[20] 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.

[21] H.-H. Tsai, X.-P. Lin, and B.-M. Chang, "An Image Filter with a Hybrid Impulse Detector Based on Decision Tree and Particle Swarm Optimization," Proc. IEEE Int'l Conf. Machine Learning and Cybernetics, July 2009.

[22] [22] H.-L. Eng and K.-K. Ma, "Noise Adaptive Soft-Switching Median Filter," IEEE Trans. Image Processing, vol. 10, no. 2, pp. 242-251, Feb. 2001.

[23] G. Pok, J. Liu, and A.S. Nair, "Selective Removal of Impulse Noise Based on Homogeneity Level Information," IEEE Trans. Image Processing, vol. 12, no. 1, pp. 85-92, Jan. 2003.

April 2016
VOLUME -2
ISSUE-4
Page:7899-06

[24] Z. Wang and D. Zhang, "Progressive Switching Median Filter for the Removal of Impulse Noise from Highly Corrupted Images," IEEE Trans. Circuits Systems II, Analog Digital Signal Processing, vol. 46, no. 1, pp. 78-80, Jan. 1999.

[25] 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.

[26] X. Zhang and Y. Xiong, "Impulse Noise Removal Using Directional Difference Based Noise Detector and Adaptive Weighted Mean Filter," IEEE Signal Processing Letters, vol. 16, no. 4, pp. 295-298, Apr. 2009.

[27] 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.

[28] F.J. Gallegos-Funes, V.I. Ponomaryov, S. Sadovnychiy, and L. Nino-de-Rivera, "Median M-Type K-Nearest Neighbour (MMKNN) Filter to Remove Impulse Noise from Corrupted Images," IEEE Electronics Letters, vol. 38, no. 15, pp. 786-787, July 2002.

[29] 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.

[30] Z. Wang and D. Zhang, "Restoration of Impulse Noise Corrupted Images Using Long-Range Correlation," IEEE Signal Processing Letters, vol. 5, no. 1, pp. 4-7, Jan. 1998.

[31] 0.18 _m TSMC/Artisan Memory Compiler, http://www.artisan.com, 2012.

[32] Synopsys Inc., "DesignWare Building Block IP," http://www. ynopsys.com, 2012.

[33] S.-C. Hsia, "Parallel VLSI Design for a Real-Time Video-Impulse Noise-Reduction Processor," IEEE Trans. Very Large Scale Integration Systems, vol. 11, no. 4, pp. 651-658, Aug. 2003.

[34] 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.

[35] V. Fischer, R. Lukac, and K. Martin, "Cost-Effective Video Filtering Solution for Real-Time Vision Systems," EURASIP J. Applied Signal Processing, vol. 13, pp. 2026-2042, 2005.

[36] 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.

[37] 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.

Author's Profile:

SUKKIREDDY VAISHNAVI completed her

B.Tech in 2013 from Usha Rama College of Engg and Technology, JNTUK and she is working towards her M.Tech from Aditya Engineering College, JNTUK. Her area of interest is in VLSI Design.

GAMINI SRIDEVI completed her B.Tech in

2000 from Nagarjuna University. She completed her M.Tech from University College of Engineering, JNTUK and she is working towards her Ph.D from the same University. Her area of interest is Digital Image Processing, Signal Processing, Fractional Calculus and Embedded Systems.