Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.517 – 524

RESEARCH ARTICLE

REALIZATION OF VLSI ARCHITECTURE FOR DECISION TREE BASED DENOISING METHOD IN IMAGES

Gobu.CK[1], Priya.R[2]

[1]PG Scholar, Electrical and Electronics Engineering &Anna University, India
[2]Senior Assistant Professor, Electrical and Electronics Engineering &Anna University, India
[1] priya77r@gmail.com; [2] ckgobu@gmail.com

ABSTRACT-In the process of signal acquisition and transmission image signals might be corrupted by impulse noise. Efficient VLSI implementation is presented in this paper, in order to remove impulse noise. In order to perform better visual quality, edge features should be preserved. Pixels that are detected as noisy are filtered, the others remain unchanged. Here fixed value impulse noise is removed and implemented in VLSI. The VLSI architecture of our design yields a processing rate of about 200 MHz by using TSMC 0.18µm technology. Compared with the state-of-the-art techniques, this work can reduce memory storage by more than 99%. The design requires only low computational complexity and two line memory buffers. Its hardware cost is low and suitable to be applied to many real-time applications.

Index terms: Image denoising, impulse noise, impulse detector, architecture

I. INTRODUCTION

Image processing is widely used in many fields, such as medical imaging, scanning techniques, printing skills, license plate recognition, face recognition, and so on. In general, images are often corrupted by impulse noise in the procedures of image acquisition and transmission. The noise may seriously affect the performance of image processing techniques. Hence, an efficient denoising technique becomes a very important issue in image processing. According to the distribution of noisy pixel values, impulse noise can be classified into two categories: fixed-valued impulse noise and random-valued impulse noise. Impulse noise is also known as salt-and-pepper noise because the pixel value of a noisy pixel is either minimum or maximum value in grayscale images. We only focus on removing the random-valued impulse noise from the corrupted image in this paper.

The Proposed DTBDM

The noise considered in this paper is random-valued impulse noise with uniform distribution as practiced in Here, we adopt a 3 × 3 mask for image denoising. Assume the pixel to be denoised is located at coordinate and denoted as $p_{i,j}$, and its luminance value is named as $f_{i,i}$, as shown in

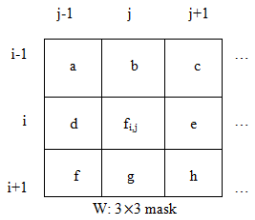

Fig 1. 3×3 mask

According to the input sequence of image denoising process, we can divide other eight pixel values into two sets: $W_{TopHalf}$ and $W_{BottomHalf}$. 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 $p_{i,j}$ is a noisy pixel by using the decision tree and the correlation between pixel $p_{i,j}$ 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. The design concept of the DTBDM is displayed in Fig. 2.

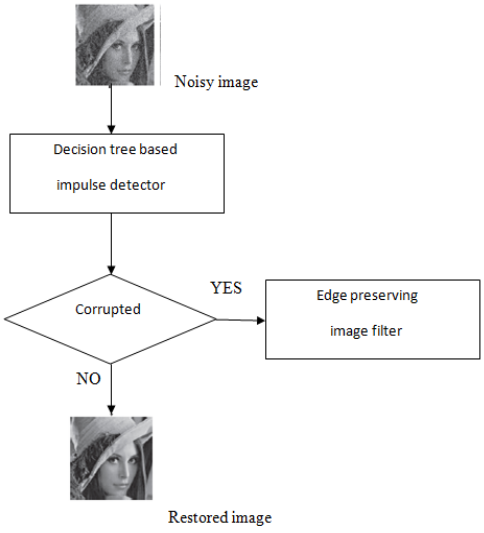

Fig 2. Dataflow diagram of DTBDM

II. OVERVIEW OF DTBDM

2.1 Decision Tree Based Impulse Detector

In order to determine whether $p_{i,j}$ 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 $p_{i,j}$ 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, $p_{i,j}$ 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 $W_{BottomHalf}$, 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

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

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

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

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, as shown in Fig. 7.

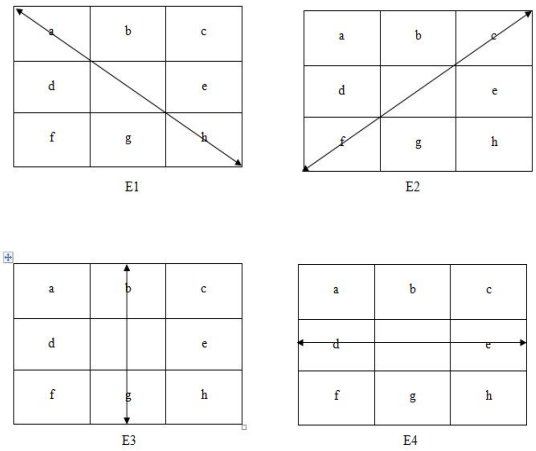

Fig 3. four directional difference of mask

We take direction E1 for example. By calculating the absolute difference between $f_{i,j}$ 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

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

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

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

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

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

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 4thinWi,j, MedianInWi,j, and 6thinWi,j. We define $Max_{i,j}$ and $Min_{i,j}$ as

$$Max_{i,j} = 6_{th}inW_{i,j} + ThSM_a$$
$$Min_{i,j} = 4_{th}inW_{i,j} - ThSM_a \quad (5)$$

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

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

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

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

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

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.

2.2 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:

$$D_1 = |d-h| + |a-e|$$

$$D_2 = |a-g| + |b-h|$$

$$D_3 = |b-g| \times 2$$

$$D_4 = |b-f| + |c-g|$$

$$D_5 = |c-d| + |e-f|$$

$$D_6 = |d-e| \times 2$$

$$D_7 = |a-h| \times 2 \quad (8)$$

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

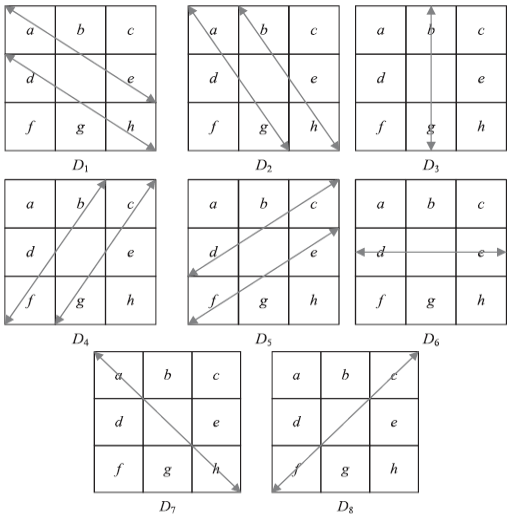

Fig 4. Eight directional differences of DTBDM.

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

$$F_{i,j}=\text{Median}(f_{i,j},b,d,e,g) \qquad (9)$$

III. 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. In our implementation, the SRAM used to store the image luminance values is generated with the 0:18 µm TSMC/Artisan memory compiler and each of them is 512 × 8 bits. According to the simulation results obtained from Design Ware of SYNOPSYS we find that the access time for SRAM is about 5 ns. Hence, we adopt the 7-stage pipelined architecture for DTBDM.

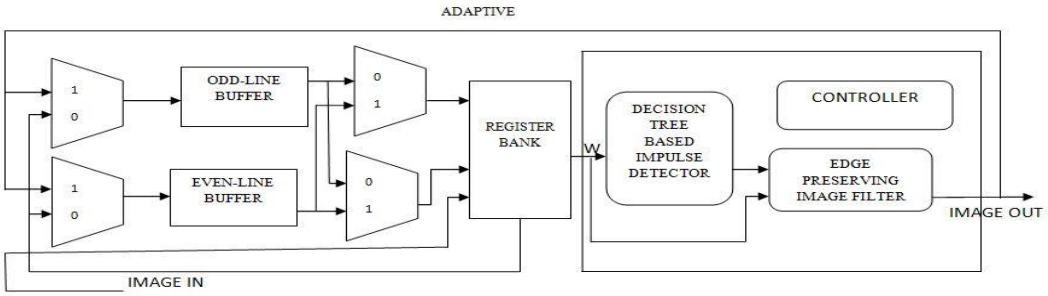

Fig.5. Block diagram of VLSI architecture of DTBDM.

IV. Simulation Results of DTBDM

The proposed DTBDM is implemented in VHDL, simulated using Xilinx ISE Simulator and synthesized using Xilinx ISE tools version 9.2i to Vertex XC2VP50-7 FPGA device. Fig.7 shows the DTBDM results for coin image corrupted by impulse noise.

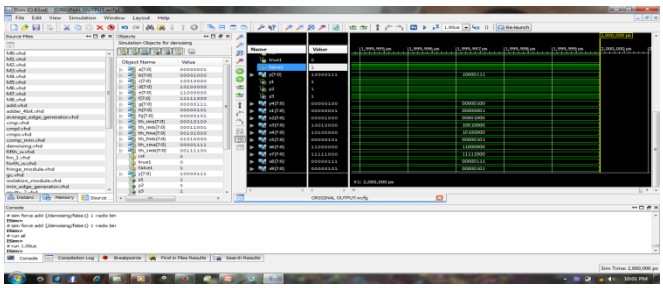

Fig 6. DTBDM result for 3×3 window

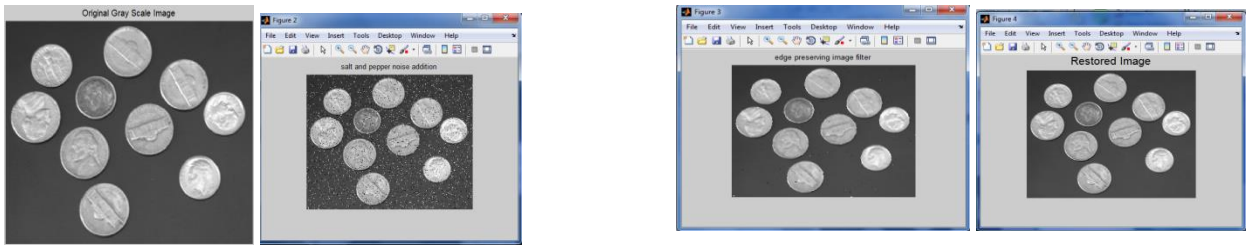

Fig 7. (a) Noise free image (b) Noisy image With noise density 0.4 (c) Edge preserving image filter(d) Restored image

Table 1. Simulation results

Noise density	Mean square Error	PSNR
0.2	4.7159	41.4292
0.3	6.75349	39.3940
0.4	10.5886	37.9164
0.5	14.0992	36.6728

V. CONCLUSIONS

A low-cost VLSI architecture for efficient removal of random-valued impulse noise is proposed in this paper. The approach uses the decision-tree-based detector to detect the noisy pixel and employs an effective design to locate the edge. With adaptive skill, the quality of the reconstructed images is notable improved. Our extensive experimental results demonstrate that the performance of our proposed technique is better than the previous lower complexity methods and is comparable to the higher complexity methods in terms of both quantitative evaluation and visual quality. The VLSI architecture of our design yields a processing rate of about 200 MHz by using TSMC 0:18 μm technology. It requires only low computational complexity and two line memory buffers. Therefore, it is very suitable to be applied to many real-time applications.

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] H. Hwang and R.A. Haddad, "Adaptive Median Filters: New and Results," IEEE Trans. Image Processing, vol. 4, no. 4, pp. 499-502, Apr. 1995.

[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] 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.

[6] P.E. Ng and K.K. Ma, "A Switching Median Filter with Boundary Discriminative Noise Detection for Extremely Corrupted Images," IEEE Trans. Image Processing, vol. 15, no. 6, pp. 1506-1516, June 2006.

[7] 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.

[8] T. Nodes 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.

[9] S.-J. Ko and Y.-H. Lee, "Center Weighted Median Filters and Their Applications to Image Enhancement," IEEE Trans. Circuits Systems, vol. 38, no. 9, pp. 984-993, Sept. 1991.

[10] T. Sun and Y. Neuvo, "Detail-Preserving Median Based Filters in Image Processing," Pattern RecognitionLetters, vol. 15, pp. 341-347, Apr. 1994.

[11] E. Abreu, M. Lightstone, S.K. Mitra, and K. Arakawa, "A New Efficient Approach for the Removal of Impulse Noise from Highly Corrupted Images," IEEE Trans. Image Processing, vol. 5, no. 6, pp. 1012-1025, June 1996.

[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.

