INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

DENOISING OF THE IMAGE DURING ACQUISITION & TRANSMISSION USING AN EFFICIENT VLSI ARCHITECTURE

UMAR.M[1], SUBBULAKSHMI.N[2]

[1]PG SCHOLAR DEPT.OF VLSI DESIGN, SRI RAMAKRISHNA ENGINEERING COLLEGE COIMBATURE.
er.umar@yahoo.com
[2]ASSISTANT PROFESSOR DEPT OF EC (UG), SRI RAMAKRISHNA ENGINEERING COLLEGE COIMBATURE
lakshu.125@ygmail.com

Abstract: The aim of this project is to remove the impulse noise from the noisy image. The efficient de noising scheme and its VLSI architecture for the removal of random-valued impulse noise. To achieve the goal of low cost, a low-complexity using VLSI architecture . We employ a decision-tree-based impulse noise detector to detect the noisy pixels, and an edge-preserving filter to reconstruct the intensity values of noisy pixels. Furthermore, an adaptive technology is used to enhance the effects of removal of impulse noise. This technique can obtain better performances in terms of both quantitative evaluation and visual quality.

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. The former is also known as salt-and-pepper noise because the pixel value of a noisy pixel is either minimum or maximum value in gray-scale images. The values of noisy pixels corrupted by random-valued impulse noise are uniformly distributed in the range of [0, 255] for gray-scale images. There have been many methods for removing salt-and-pepper noise, and

some of them perform very well The random-valued impulse noise is more difficult to handle due to the random distribution of noisy pixel values. We only focus on removing the random-valued.

Recently, many image denoising methods have been proposed to carry out impulse noise suppression Some of them employ the standard median filter [or its modifications However, these approaches might blur the image since both noisy and noise-free pixels are modified. To avoid the damage on noise-free pixels, an efficient switching strategy has been proposed in the literature In general, the switching median filter consists of two steps: 1) impulse detection and 2) noise filtering. It locates the noisy pixels with an impulse detector, and then filters them rather than the whole pixels of an image to avoid causing the damage on noise-free pixels. In addition to median filer, there are other methods used to carry out impulse noise. In [proposed an alpha-trimmed mean-based method (ATMBM). It used the alpha-trimmed mean in impulse detection and replaced the noisy pixel value by a linear combination of its original value and the median of its local window.

II. THE PROPOSED DTBDM

The noise considered in this paper is random-valued impulse noise with uniform distribution as practiced. 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 $p_{i;j}$, and its luminance value is named as $f_{i;j}$, as shown in Fig. 1. According to the input sequence of image denoising process, we can divide other eight pixel values into two sets: $W_{TopHalf}$ and $W_{BottomHalf}$.

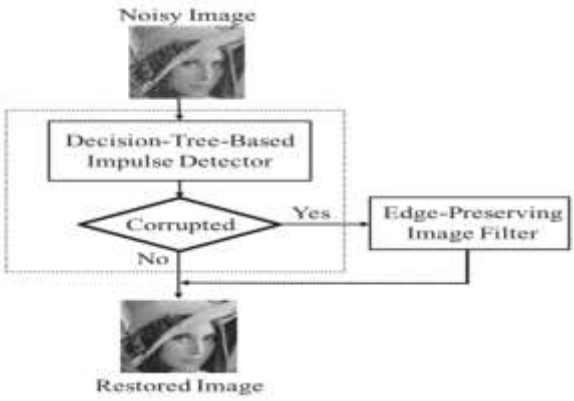


Fig. 1. The dataflow of DTBDM.

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, edge-preserving 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. 1. In order to determine whether $p_{i;j}$ is a noisy pixel, the correlations between $p_{i;j}$ and its neighboring pixels are considered There-fore, 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

A. Isolation Module

The pixel values in a smooth region should be close or locally slightly varying, as shown in Fig. 3. The differences between its neighboring pixel values are small. If there are noisy values, edges, or blocks in this region, the distribution of the values is different, as shown in Fig. 2. Therefore, we determine whether current pixel is an isolation point by observing the smoothness of its surrounding pixels. Fig. 3 shows an example of noisy image. 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.

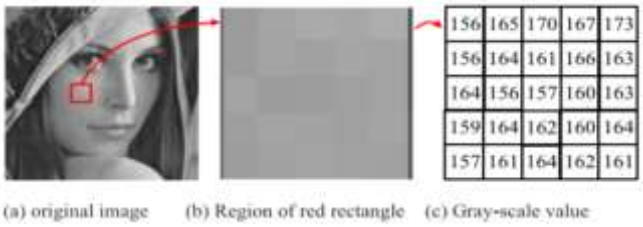


Fig. 2 A smooth region in Lena.

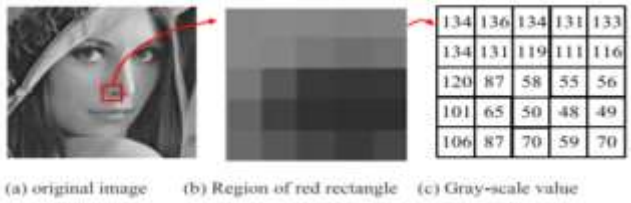


Fig. 3 A non-smooth region in Lena.

B. Fringe module

If $p_{i;j}$ has a great difference with neighboring pixels, it might be a noisy pixel (discussed in Section 2.1.1 IM) or just situated on an edge. 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 E_1 to E_4, as shown in Fig. 4. We take direction E_1 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 edgeornot.

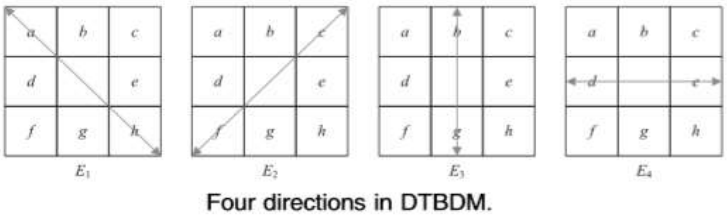


Fig. 4 Four directions in DTBDM

C. Similarity module

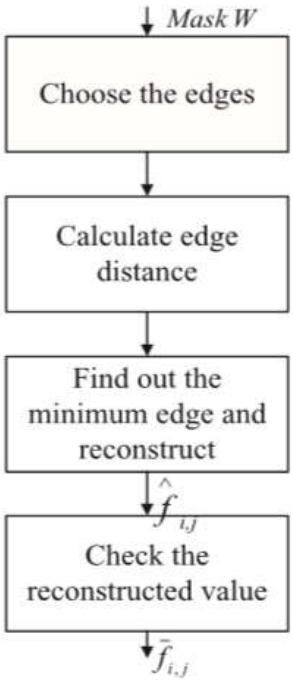
The last module is similarity module. The luminance values in mask W located in a noisy-free area might be close. The median is always located in the center of the variational series, while the impulse is usually located near one of its ends. Hence, if there are extreme big or small values, that implies 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.

III. EDGE-PRESERVING IMAGE FILTER

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 Maxi;j and Mini;j aslTo locate the edge existing in the current W, a simple edge-preserving technique which can be realized easily with VLSI circuit is adopted.

The dataflow and the pseudo code of our edge-preserving image filter are shown in Fig 5 and 6, respectively. Here, we consider eight directional differences, from D_1 to D_8, to reconstruct the noisy pixel value, as shown in Fig. 5 Only those composed of noise-free pixels are taken into account to avoid possible misdetection. Directions passing through the suspected pixels are dis-carded 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



Dataflow of edge-preserving image filter.

Fig. 5 Dataflow of edge-preserving image filter

.

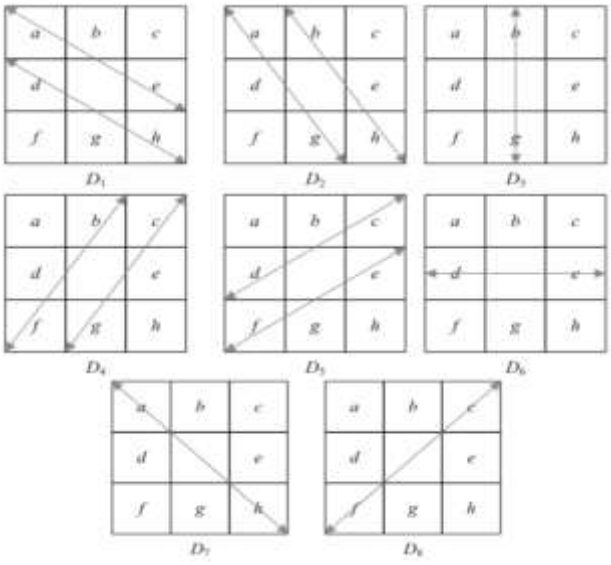


Fig.6 Eight directional differences of DTBDM.

Estimated value of $p_{i;j}$) is equal to the weighted average of luminance values of three previously denoised pixels and calculated as ða þ b _ 2 þ cÞ=4. In other conditions, the edge filter calculates the directional differences of the chosen directions and locates the smallest one ðD_{min}Þ among them in the third block.

IV.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 _m TSMC/Artisan memory compiler and each of them are 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.

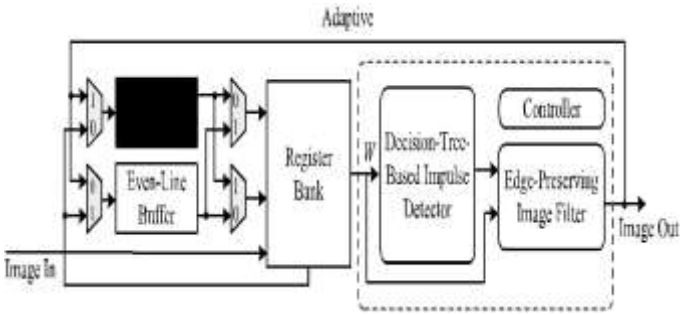


Fig.7 Block diagram of VLSI architecture of DTBDM.

Fig. 7 shows block diagram of the VLSI architecture for DTBDM. The architecture adopts an adaptive technology and consists of five main blocks: line buffer, register bank (RB), decision-tree-based impulse detector, edge-preserving im-age filter, and controller.

V.OUTPUT RESPONSE

The output response of the original, noisy and the improved image is shown in the following fig 8,fig 9 and fig 10.

Fig.8 Original image

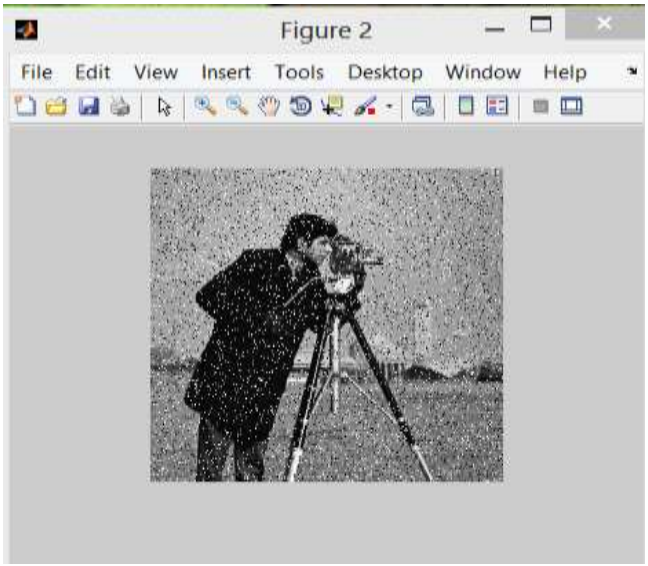


Fig.9 Noisy image

Fig.10 Improved image after noise removed.

VI. CONCLUSION

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. 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-Inter science, 1991.

[3] H. Hwang and R.A. Haddad, "Adaptive Median Filters: New Algorithms 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 ,"EEE 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