Available online at www.elixirpublishers.com (Elixir International Journal)

Advanced Engineering Informatics

ISSN: 2229-712X

Elixir Adv. Engg. Info. 38 (2011) 4105-4108

Removal of impulse noise using VLSI technology

N.J.R. Muniraj[1] and Surya Prabha[2]

Tejaa Sakthi Institute of Technology, Coimbatore, Tamilnadu.

ARTICLE INFO

Article history:
Received: 23 June 2011;
Received in revised form:
22 August 2011;
Accepted: 27 August 2011;

Keywords
Image denoising,
Impulse noise,
Pipeline architecture,
VLSI.

ABSTRACT

Impulse noise is the major factor that affects the image during signal acquisition and transmission. Here an efficient simple edge preserving denoising technique is used to remove the impulse noise. To avoid the possible misdetection this technique does not affect the noise free pixel. It uses different directional edges to preserve the edge information. The experimental result shows excellent performances in terms of quantitative evaluation and visual quality. The design cost is also low because it needs only two line memory buffer and less computational complexity.

Introduction

In applications, such as printing skills, medical imaging, scanning techniques, image segmentation, and face recognition, image are often corrupted by noise in the process of image acquisition and transmission. Hence an efficient denoising technique is very important for the image processing applications. Recently, many denoising techniques have been proposed to suppress the impulse noise. Some of them employ the standard median filter or its modifications to implement the denoising process. 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 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 the damage on noise-free pixels. The denoising methods for impulse noise suppression can be classified into two categories: lower complexity techniques and higher complexity techniques. Lower complexity technique uses a fixed-size local window and requires a few line buffers. Further, its computational complexity is low and can be comparable to conventional median filter or its modification. Higher complexity technique uses enlarging local window size. In this paper, lower complexity technique only used because of its simplicity.

In [4], Zhang and Karim proposed a new impulse detector (NID) for switching median filter. NID used the minimum absolute value of four convolutions which are obtained by using 1-D Laplacian operators to detect noisy pixels. The image quality is determined by calculating mean square error(MSE). The computational complexity is high in this method.

A method named as differential rank impulse detector (DRID) is used in Effective impulse detector based on Rank-order criteria by Aizenberg and Butakoff [5]. The impulse detector of DRID is based on a comparison of signal samples within a narrow rank window by both rank and absolute value. The impulse is identified by the height its brightness jump in comparison with the surrounding pixel. The computation mechanism is fast end efficient but the image smoothening is minimum.

In [6], Luo proposed a method which can efficiently remove the impulse noise (ERIN) based on simple fuzzy impulse detection technique. Impulse detection is made with fuzzy set.

An alpha-trimmed mean-based method (ATMBM) was presented by W. Luo [7]. 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. The process is used iteratively to get good results.

In [8], a decision-based algorithm (DBA) is proposed to remove the corrupted pixel by the median or by its neighbouring pixel value according the proposed decisions. The picture quality is determined by calculating PSNR. The computational complexity is high.

For real-time embedded applications, the VLSI implementation of switching median filter for impulse noise removal is necessary and should be considered. For customers, cost is usually the most important issue while choosing consumer electronic products. Hence low-cost denoising implementation is focused in this paper. The cost of VLSI implementation depends mainly on the required memory and computational complexity. Hence, less memory and few operations are necessary for a low-cost denoising implementation. Based on these two factors, a simple edge-preserved denoising technique (SEPD) and its VLSI implementation for removing fixed-value impulse noise is proposed. The storage space needed for SEPD is two line buffers rather than a full frame buffer. Only simple arithmetic operations, such as addition and subtraction, are used in SEPD. Here impulse noise detector is used to detect the noisy pixel and employs an effective design to locate the edge of it.

The rest of this paper is organized as follows. In Section II, the proposed SEPD is introduced. The VLSI implementation of SEPD is described briefly in Section III. The results and comparison are provided in Section IV. Conclusions are presented in Section V.

Proposed Sped

Assume that the current pixel to be denoised is located at

Tele:
E-mail addresses: njrmuniraj@yahoo.com

coordinate (i,j) and denoted as $P_{i,j}$, and its luminance value before and after denoising process are represented as $f_{i,j}$ and $\bar{f}_{i,j}$. If $P_{i,j}$ is corrupted by fixed value impulse noise, its luminance value jump to maximum or minimum value in gray scale.

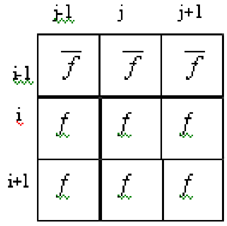


Fig 1: 3x3 mask centered on $P_{i,j}$.

Here we used 3X3 mask W centering on $P_{i,j}$ for image denoising. In the current W, we know that the three denoised values at coordinates (i-1,j-1),(i-1,j) and (i-1,j+1) are determined at the previous denoising process and six pixels at coordinates (i,j-1),(i,j), (i, j+1), (i+1,j-1),(i+1,j) and (i+1,j+1) are not denoised yet.

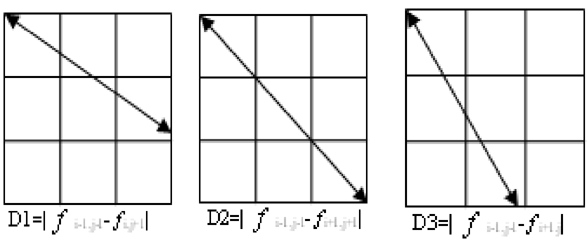
SEPD is composed of three components: extreme data detector, edge-oriented noise filter and impulse arbiter. The extreme data detector detects the minimum and maximum luminance values in W, and determines whether the luminance values of $P_{i,j}$ and its five neighbouring pixels are equal to the extreme data. By observing the spatial correlation, the edge-oriented noise filter uses a directional edges to generate the estimated value of current pixel. Finally, the impulse arbiter brings out the proper result. The three components of SEPD are described in detail in the following subsections.

Extreme Data Detector

The extreme data detector detects the minimum and maximum luminance value in the image. If a pixel is corrupted by the fixed-value impulse noise, its luminance value will jump to the minimum or maximum value in gray scale. If $f_{i,j}$ is not equal to minimum or maximum value, it is concluded as $P_{i,j}$ is noise-free pixel and the denoising process is skipped. If $f_{i,j}$ is equal to minimum or maximum value, its five neighbouring pixels are checked and the result is stored into register B.

Edge-Oriented Noise Filter

To locate the edge existed in the current W, a simple edge-catching technique which can be realized easily with VLSI circuit is adopted. To decide the edge, we consider 12 directional differences, from to D_1 to D_{12}, as shown in Figure 2. Only those composed of noise-free pixels are taken into account to avoid possible misdetection. If a bit in B is equal to 1, it means that the pixel related to the binary flag is suspected to be a noisy pixel. Directions passing through the suspected pixels are discarded to reduce misdetection. In each condition, at most four directions are chosen for low-cost hardware implementation. If there appear over four directions, only four of them are chose according to the variation in angle.



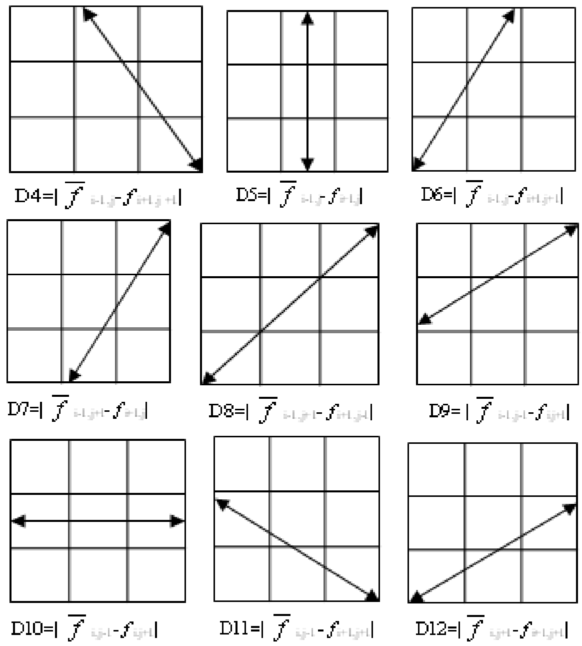


Fig 2: Twelve directional difference of SEPD

If all the five neighbouring pixels are suspected to be noisy pixel(B = "11111"), no edge can be processed, so $\hat{f}$ is calculated with the weighted average of luminance value of three previously denoised pixels as ($\bar{f}_{i-1,j-1}$ +2 x $\bar{f}_{i-1,j}$ + $\bar{f}_{i-1,j+1}$)/4. For other conditions except when B= "11111" the edge filter calculates the directional differences of the chosen directions and locates the smallest one. The mean of luminance values of the two pixel which possess the smallest directional difference is treated as $\hat{f}_{i,j}$.

For example, if B is equal to "10011" it means that $f_{i,j-1}$, $f_{i+1,j}$ and $f_{i+1,j+1}$ are suspected to be noisy values. Therefore, D_2 - D_5, $D_7$and D_9 - D_{11} are discarded because they contain those suspected pixels. The four chosen directional differences are D_1, D_6, D_8 and D_{12}. Finally, $\hat{f}_{i,j}$ is equal to the mean of luminance values of the two pixels which possess the smallest directional difference among D_1, D_6, D_8 and D_{12}.

Impulse Arbiter

The value of a pixel, corrupted by the fixed-value impulse noise will jump to be the minimum or maximum value in gray scale. However, the converse is not true. Pixel with minimum or maximum luminance values might be identified as a noisy pixel even if it is not corrupted. To overcome this, additional condition is used to reduce the possibility of misdetection. If $P_{i,j}$ is a noise free pixel and the current mask has high spatial correlation, $f_{i,j}$ should be close to $\hat{f}_{i,j}$ and $|f_{i,j} - \hat{f}_{i,j}|$ is small. The value of $|f_{i,j} - \hat{f}_{i,j}|$ is measured and compared with the threshold value to determine whether $P_{i,j}$ is corrupted or not. If $P_{i,j}$ is judged as a corrupted pixel, the reconstructed luminance value $\bar{f}_{i,j}$ is equal to $f_{i,j}$.

VLSI Implementation of SEPD

SEPD has low computational complexity and requires only two line memory buffers, so its cost of VLSI implementation is low. For better timing performance, the pipelined architecture is used which produces an output at every clock cycle. Here SRAM is used to store the image luminance value. Figure 3 shows the block diagram of the 7-stage pipeline architecture for

SEPD. The architecture consists of five main blocks: line buffer, register bank, extreme data detector, edge-oriented noise filter and impulse arbiter. Each block is described briefly in the following subsections.

Line Buffer

SEPD adopts a 3 x 3 mask, so three scanning lines are needed. If $P_{i,j}$ are processed, three pixels from row_{i-1}, row_i and row_{i+1}, are needed to perform the denoising process. Four crossover multiplexers are used to realize three scanning lines with two line buffers. Line Buffer-odd and Line Buffer-Even are used to store the pixels at odd and even rows.

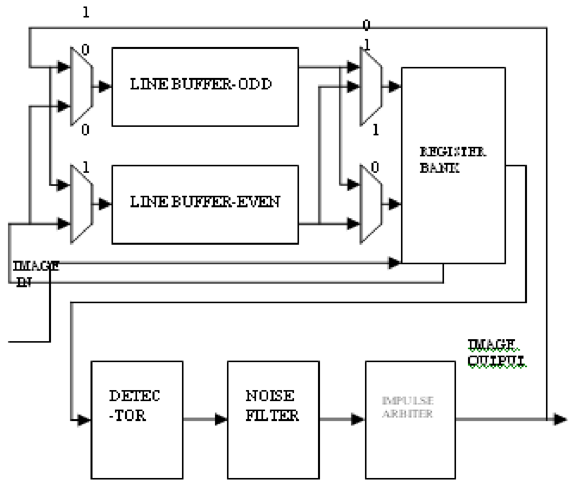


Fig 3: Block Diagram for the Architecture of SEPD

Register Bank

The Register bank (RB), consisting of 9 registers, is used to store the 3 x 3 pixel values. Each 3 registers are connected serially in a chain to provide three pixel values of a row and Reg4 keeps the luminance value of the current pixel to be denoised. The nine values stored in RB are then used simultaneously by subsequent extreme data detector and noise filter for denoising.

Once the denoising process for $P_{i,j}$ is completed, the reconstructed pixel value $\bar{f}_{i,j}$ generated by the arbiter is outputted and written into the line buffer storing row_i to replace $f_{i,j}$. When the denoising process shifts from $P_{i,j}$ to $P_{i,j+1}$, only 3 new values are needed to be read into RB and other 6 pixel values are shifted to each one's proper register. At the same time, the previous value in Reg8 is written back to the line buffer storing row_{i-1} for subsequent denoising process.

Extreme Data Detector

The Extreme data detector uses 3 pipeline stages. It consists of pipeline register and equality comparator which gives output as logic 1 if both two input values are identical. The 2 pipeline stages are used to find the minimum and maximum luminance value. Two columns of equality comparator is used to determine whether the lower six pixel in the mask is equal to minimum or maximum value. If the pixel is noise-free the following denoising process is skipped.

Edge-Oriented Noise Filter

It uses 2 pipeline stages. It uses |SUB| unit to determine the absolute value of difference of two inputs. The smallest one is determined by using the minimum tree unit. The mean of luminance value of two pixels which possess the smallest directional difference is obtained. When B= "11111" the multiplexer will output ($\bar{f}_{i-1,j-1}$ +2 x $\bar{f}_{i-1,j}$ + $\bar{f}_{i-1,j+1}$)/4. When B ~= "11111" the multiplexer will output the mean of the two pixel value.

Impulse Arbiter

It uses comparator to give the output logic 1 if ($|f_{i,j} - \hat{f}_{i,j}|$) is greater than threshold. The multiplexer is used to output $\hat{f}_{i,j}$ generated by noise filter when pixel is corrupted or $f_{i,j}$ when pixel is noise free.

In the design, one clock cycle is used to fetch a value from line buffer and load it into Register bank. Three clock cycles are needed for the extreme data detector. The edge-oriented noise filter requires two clock cycles and impulse arbiter requires one clock cycle to complete their functions. Totally the design requires seven clock cycle to perform the denoising process for a pixel.

Results

To verify the characteristics and performances of various denoising algorithms, a variety of simulations are carried out on the well-known 256 x 256 8-bit gray-scale Lena image. In the simulations, image is corrupted by impulse noise (salt-and-pepper noise), where "salt" and "pepper" noise are with equal probability. The peak signal to noise ratio (PSNR) is calculated to illustrate the quantitative quality of the reconstructed image. Table I and II show PSNR of the experiments with impulse noise at various noise densities from 10% to 90% for the reference images. It can be observed from the results that the performances of the images processed by the proposed algorithm are always better.

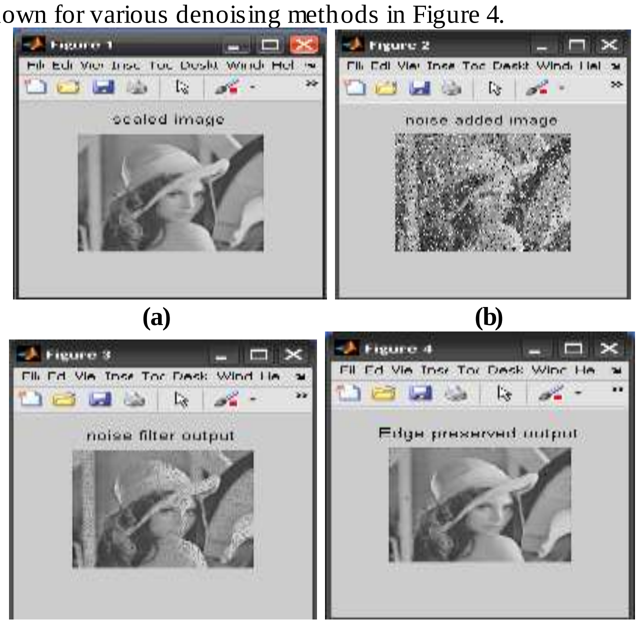
Table I Comparisons of PSNR for image "Lena"

Noise	10%	20%	30%	40%	90%
Median	23.68	23.50	22.48	19.25	17.34
DBA	25.19	24.13	24.10	24.05	23.45
SEPD	26.20	25.20	25.19	25.16	24.90

Table II Comparisons of PSNR for image "Onion"

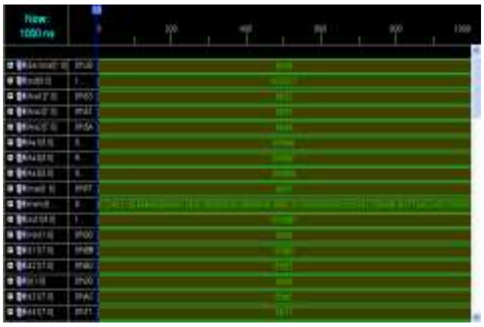
Noise	10%	20%	30%	40%	90%
Median	21.70	22.89	23.78	18.61	16.64
DBA	30.47	27.46	25.87	24.45	20.98
SEPD	44.99	40.74	39.06	37.55	33.27

To explore the visual quality, the restored image of lena is shown for various denoising methods in Figure 4.

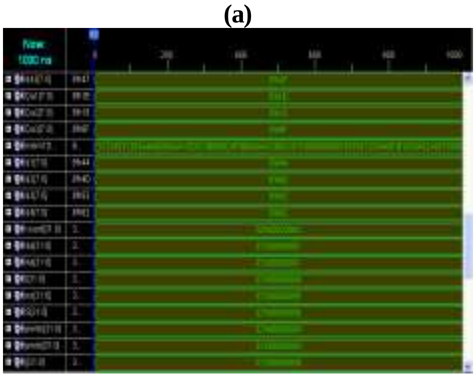


(a) (b) (c) (d)

Fig 4: Results of SEPD in MATLAB, (a) Noise-free image; (b)Noisy image; (c)Noise Filter output; (d) Edge Preserved output



(a)



(b)

Fig 5: Results of SEPD in Xilinx (a) input data (b) Output data

Conclusion

In this paper, the Matlab simulation for the removal of impulse noise is presented. The architecture is implemented in VLSI for efficient removal of impulse noise. The design requires only two line memory buffer and computational complexity is less, therefore the cost of implementation is less. The extensive experimental results shows that our design achieves excellent performance in terms of quantitative analysis and visual quality, even when the noise ratio is as high as 90%. This architecture can be extended for working with RGB color images and videos.

References

[a] T. Nodes and N. Gallagher, “Median filters: Some modifications and their properties,” IEEE Trans. Acoust., Speech, Signal Process., vol. ASSP-30, no. 5, pp. 739-746, Oct. 1982.

[b] S. J. Ko and Y. H. Lee, “Center weighted median filters and their applications to image enhancement,” IEEE Trans. Circuits Syst., Vol. 38, no. 9, pp. 984-993, Sep. 1991.

[c] H. Hwang and R. Haddad, “Adaptive median filters: New algorithms and results,” IEEE Trans. Image Process., vol. 4, no. 4, pp. 499–502, Apr. 1995.

[d] S. Zhang and M. A. Karim, “A new impulse detector for switching median filter,” IEEE Signal Process. Lett., vol. 9, no. 11, pp. 360–363, Nov. 2002.

[e] I. Aizenberg and C. Butakoff, “Effective impulse detector based on rank-order criteria,” IEEE Signal Process. Lett., vol. 11, no. 3, pp. 363–366, Mar. 2004.

[f] W. Luo, “Efficient removal of impulse noise from digital images,” IEEE Trans. Consum. Electron., vol. 52, no. 2, pp. 523–527, May 2006.

[g] W. Luo, “An efficient detail-preserving approach for removing impulse noise in images,” IEEE Signal Process. Lett., vol. 13, no. 7, pp. 413–416, Jul. 2006.

[h] K. S. Srinivasan and D. Ebenezer, “A new fast and efficient decision-based algorithm for removal of high-density impulse noises,” IEEE Signal Process. Lett., vol. 14, no. 3, pp. 189–192, Mar. 2007.

[i] S.-C. Hsia, “Parallel VLSI design for a real-time video-impulse noise-reduction processor,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 11, no. 4, pp. 651–658, Aug. 2003.

[j] V. Fischer, R. Lukac, and K. Martin, “Cost-effective video filtering solution for real-time vision systems,” EURASIP J. Appl. Signal Process., vol. 2005, no. 13, pp. 2026–2042, Jan. 2005.