2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Real-Time Removal of Random Value Impulse Noise in Medical Images

Z. HosseinKhani[1], N. Karimi[1], S.M.R. Soroushmehr[2], M. Hajabdollahi[1], S. Samavi[1,3], K. Ward[4], K. Najarian[2,3]

[1]Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, 84156-83111 Iran
[2]Michigan Center for Integrative Research in Critical Care, University of Michigan, Ann Arbor, 48109 U.S.A.
[3]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, 48109 U.S.A.
[4]Department of Emergency Medicine, University of Michigan, Ann Arbor, 48109 U.S.A.

***Abstract*— With the increasing use of telemedicine there is a great demand in real-time processing and transmission of medical images. Noise is one of the important factors that degrade the quality of medical images. Impulse noise is a common noise that could be caused by malfunctioning of sensors or by data transmission errors. It is one the most common noises that have extensively been studied in recent years. For real-time noise removal hardware techniques are more suited, since software methods are complex and slow. Usually hardware techniques have low complexity and low accuracy. In this paper a low complexity, high accuracy, de-noising method is proposed. It first categorizes image pixels into a number of groups. Then noisy pixels are restored in different ways in each category. Local analysis of image blocks allows us to restore a noisy pixel by using its neighboring non-noisy pixels. All steps are designed to have low hardware complexity. Simulation results show that in the case of MR images, the proposed method removes impulse noise with acceptable accuracy.**

Keywords*— medical image restoration; random value impulse noise; low complexity; hardware implementation.

I. Introduction

Noise is introduced in medical images either during their generation, during compression and image processing routines, or during transmission of the images. Introduction of noise degrades the quality of the image. One of the most common types of noises is the random-value impulse-noise. To remove this type of noise many studies have been conducted which consist of two stages. These stages include detection of a noisy pixel and replacement of the noisy pixel with a proper value.

Some denoising methods are complex. For example in [1] a fuzzy method, for removal of random impulse noise in colored videos, is proposed. Noisy pixels are identified step by step by fuzzy rules and a reconstruction, based on adaptive mean absolute difference, is considered. Also in [2] an evolutionary algorithm and an improved median operation are used for the detection and restoration steps respectively. In [3] the noise-like pixels are obtained based on robust outlying ratio (ROR) criterion and then image pixels are divided into four clusters. In the second stage, noisy pixels are obtained from absolute deviation of the median in each cluster. Finally, noisy pixels are restored with a nonlocal mean filter. In [4], an uncertainty based detector finds the noisy pixel. Then a weighted fuzzy filter is applied and removes the noise effect.

Some denoising methods have low complexity. For example in [5] a 3×3 window is placed around each pixel and the content is sorted. Using the 4th, 5th and 6th pixels, in the sorted array, two thresholds of T_H and T_L are specified. By using these two thresholds the noisy pixel is detected. After detection of the noisy pixels, a median operation is used for the reconstruction of the noisy pixels. Also in [6] again a window is placed around a pixel and elements inside the window are sorted. Using 4th and 6th elements, two threshold values, called N_{max} and N_{min}, are determined. If a pixel is not in the range of N_{min} and N_{max}, then it is identified as a noisy pixel. For the restoration step, edge directions are considered and noisy pixels are restored in a correct edge direction. In [7] a 3×3 window around each pixel is considered and it is sorted in all directions. After detection of noisy pixels, restoration is performed by finding the median of the non-noisy pixels. In [8] for impulse noise removal, four edge directions are considered in 5×5 windows. Using the difference between pixels in each direction, noisy pixels are detected. With median operation on non-noisy pixels, restoration is performed.

Noise and noise removal are of extra importance in medical images. For example, magnetic resonance (MR) images are affected by different noise sources, such machine generated artifacts, patient motion, signal processing noise, etc. [9]. Gaussian and impulse noises, which are created by malfunction of electrical circuits and imaging devices, are the most prevalent types of corruption factors in medical imaging [10]. Noise in MR process not only affects the quality of images but it could also ruin results of enhancement techniques [9]. Many studies have focused on impulse noise removal for medical images. In [10] a fuzzy rule-based approach is proposed for removal of impulse and Gaussian noise from angiograph images. A 2D fuzzy Kalman filtering is applied and its fuzzy rules are optimized. In [11] a fuzzy median filtering for the removal of impulse noise in MR images is proposed. Although the preservation of details in MR images is of major concern, but high computational complexity of [11] makes it unsuitable for hardware implementation. In [12] a neuro-fuzzy approach, which is an enhanced method of [11], is proposed. They use adaptive median filtering and many fuzzy rules are used for the removal of impulse noise.

The need for real-time implementation of some image processing applications makes hardware techniques more desirable. For example in [13], for the detection step, the maximum and minimum values in a 3×3 window are calculated. For the restoration step, edge directions are considered and noisy pixels are restored in the correct edge direction. In [14] a real-time approach for the suppression of impulse noise is proposed. A noisy pixel detection method and a weighted filtering method are applied for the reconstruction of noisy pixels. The weight of filtering is based on the degree of detected noise in a pixel. In [15], for the detection of random-value noisy pixels, a decision-tree and similarity between neighbor pixels are analyzed. Same as in [13], edge direction is used to restore noisy pixels. There are many studies for the acceleration of medical image processing algorithms using hardware-accelerators such as FPGAs and GPUs [16]. In the case of MR images, a large amount of data must be processed which can be a time consuming task. Hence, hardware implementation of all algorithms in this area can be useful to achieve better performance [16].

In this paper, a method for removal of random value impulse noise is proposed. Because of complex effects that random value impulse noise has on image pixels, we consider five different regions in a noisy image. These five types of regions are smooth, noisy smooth, edge, noisy edge, and jittery areas. In the first step, noisy pixels are detected. Then they are restored in each aforementioned region in different ways. Noisy pixels are not involved in the restoration process. Efficient detection of noisy pixels in the detection stage, and their removal in the restoration stage, makes the proposed method suitable for de-noising of medical images.

The rest of this paper is organized as follows. In section II, different types of impulse noises are briefly reviewed. In section III, the proposed method for removal of random value impulse noise, including software algorithm and hardware architecture, are explained. Section IV is dedicated to simulation results, and after that, in section V concluding remarks are presented.

II. IMPULSE NOISE

Impulse noise is a common type of noise consisting of a value, which is randomly distributed throughout the image. Impulse noises are divided into two main types according to the range of the injected value.

A. Fixed value impulse noise (FVIN):

As shown in (1), in grayscale images the randomly injected values could belong to one of the two constant ranges [17]. In (1), m is a constant value, $x_{i,j}$ and $y_{i,j}$ are original and noisy value of the pixel respectively, and $p1$ and $p2$ are the probabilities that the pixel gets the noisy value, where $p = p1 + p2$.

$$y_{i,j} = \begin{cases} [0, m) & with\, probability\; p1 \\ x_{i,j} & with\, probability\; (1-p) \\ [255-m, 255] & with\, probability\; p2 \end{cases} \quad (1)$$

If $m = 0$ then the induced noise is salt and pepper noise.

B. Random value impulse noise (RVIN)

As shown in (2) for gray scale images, a pixel may get noisy with a probability p, where a value in the range of 0 to 255 is randomly chosen and replaces the original pixel [18]. In (2), r is a random value, $x_{i,j}$ and $y_{i,j}$ are the original and new values of the pixel respectively.

$$y_{i,j} = \begin{cases} r & with\, probability\; p \\ x_{i,j} & with\, probability\; 1-p \end{cases} \quad (2)$$

In this research we have considered RVIN, which is more common and more challenging.

III. PROPOSED METHOD

In all real-time applications, it is necessary to use a computationally efficient and accurate algorithm. This is also true for noise removal algorithms. Such algorithms could be used as a preprocessing step for many image processing applications. Even in complex software methods, such as neural networks and learning techniques, it is the intention to propose a low complexity and high accuracy algorithm. Our proposed method, which is explained in the following sections, can be implemented on hardware with an acceptable accuracy.

A. General structure of the algorithm

The block diagram of the proposed method is displayed in Fig. 1.

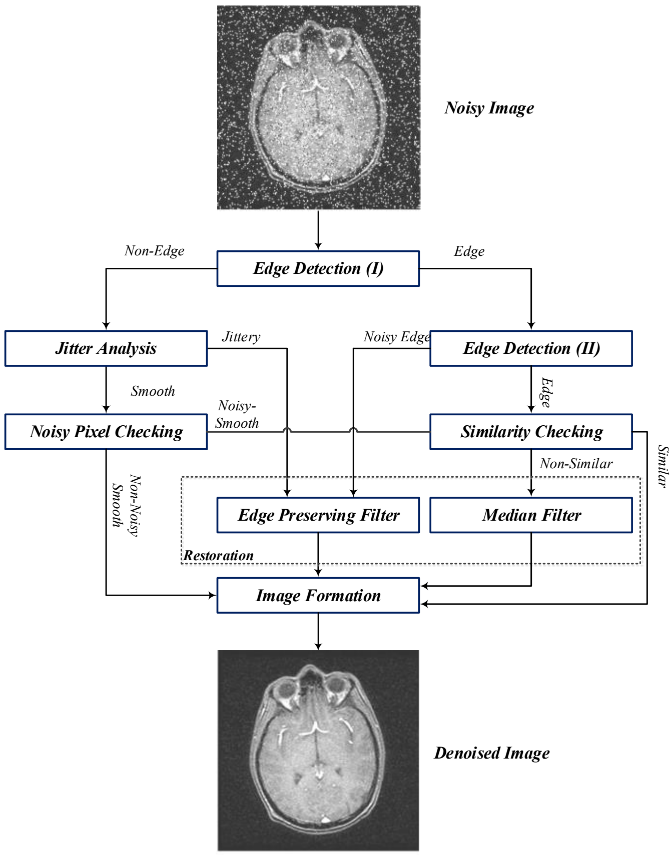


Fig 1. Block diagram of the proposed method

For RVI noise it is not easy to label a pixel as being noisy because it could have any grayscale value. That is why we have to analyze the neighborhood of each pixel and find out if the pixel is a normal part of that neighborhood or not. Hence, the proposed method categorizes 3×3 blocks of the image into five categories, which are called smooth, noisy smooth, edge, noisy edge, and jittery blocks. The 9 pixels of the block are called $P_1, P_2, \ldots, P_9$, where P_5 is the center pixel. Pixel intensities are sorted and the sorted pixel values are $F_1, F_2, \ldots, F_9$. The 6 stages of the algorithm are as follow:

1) Edge Detection

Edges are detected in two steps. In the first step the detected edges are not necessarily very strong ones. In the second step strong edges are detected. The two steps of edge detection are as follows:

- Edge detection (step I)

A 3×3 window around each pixel is considered and elements of the window are sorted. The differences between 5th element with 4th and 6th elements are considered by computing $(F_5 - F_4)$ and $(F_6 - F_5)$. Then using a threshold T_1 the center pixel is labeled based on the following criterion:

$$P_5 \begin{cases} edge & if \quad (F_5 - F_4)\ or\ (F_6 - F_5) > T_1 \\ non-edge & otherwise \end{cases} \quad (3)$$

- Edge detection (step II)

For all those labeled edge pixels, from the previous step, the second edge detection criterion is applied. In this step, four main directions of horizontal, vertical, diagonal, and anti-diagonal, are considered as shown in Fig. 2. In each direction sum of differences between the central pixel and the two other pixels located on a particular direction is calculated. Minimum value in four directions shows the possible direction of the edge. If this value is less than a threshold (T_2) then the pixel is considered as edge, otherwise it is labeled as a noisy edge pixel.

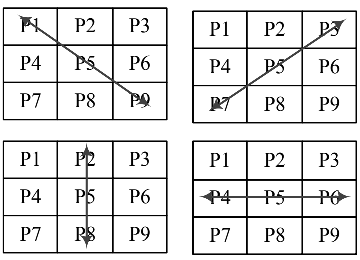


Fig. 2. Edge directions in edge detection (II)

2) Jitter Analysis

When a pixel is labeled as non-edge by edge detection (I), we want to know whether it is in a smooth area or in a jittery area. We use threshold T_3 and the following criterion.

$$P_5 \begin{cases} jittery & if \quad (F_6 + T_3) < P_5\ or\ P_5 < (F_4 - T_3) \\ smooth & otherwise \end{cases} \quad (4)$$

3) Noisy Pixels Checking

After jitter analysis, the differences between the central pixel P_5 and the maximum and minimum elements inside the window are considered.

$$P_5 \begin{cases} noisy & if \quad (F_9 - P_5)\ or\ (P_5 - F_1) < T_4 \\ not\ noisy & otherwise \end{cases} \quad (5)$$

If either $(F_9 - P_5)$ or $(P_5 - F_1)$ is less than a threshold (T_4) then the pixel is considered as noisy pixel which is located on a smooth area. For such a pixel, its similarity with its neighbors must be checked.

4) Similarity checking

Absolute differences between the central pixel and its eight neighbors are calculated. Using threshold T_4, these absolute differences reveal similarity or non-similarity among these 8 pairs. If number of the similar pixel around a pixel be less than a threshold (T_5) then it is considered as a noisy pixel.

5) Restoration

The restoration mechanisms are different in different regions. The median operation is used in the noisy smooth regions. An edge preserving method is used for restoration in noisy edge regions and in jittery areas. In the edge preserving step the noisy pixels are restored with respect to the edge direction. In [15] details of an edge preserving method are presented.

6) Image Formation

Noise-free pixels which are detected in the previous steps as well as restored pixels are placed back to form the noise free image.

B. Hardware architecture

The proposed noise removal algorithm is designed to be suitable for hardware implementation. As illustrated in Fig. 3, a window around each pixel is considered and its elements are sorted by a sorting module. Results of the sorting module are feed to three computational modules, including jitter analyzer, edge detector (I), and noisy pixel checker.

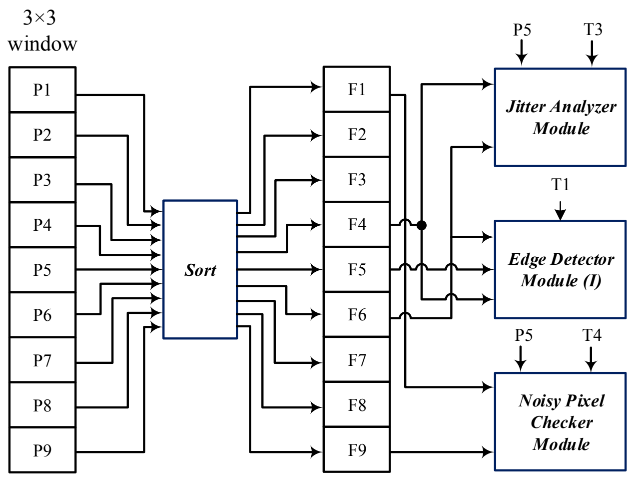


Fig. 3. Sorter module and other modules that feed from it.

Different parts of the hardware structure of the proposed algorithm are explained in the followings:

1) *Edge Detection Module (I)*

As illustrated in Fig. 4, two subtractors, and two comparator units, and a logical OR gate, form the circuit which can be used as edge detector (I). By changing the inputs of this circuit, it could be used to see if P_5 is in a jittery area or not, based on (4). Also, same circuit, with appropriate inputs, could be used for implementation of (5) as a noisy pixel checker module.

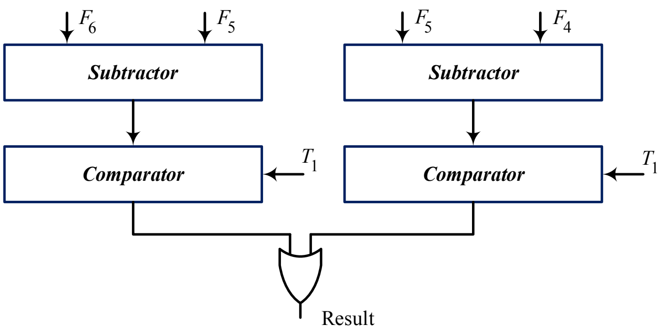


Fig. 4. Hardware realization of edge detector module (I).

2) *Edge Detection Module (II)*

Differences between the central pixel, P_5, and its 8 neighbors are needed. Eight subtractor units are used. Differences in each direction are added and the minimum of the four directions is found. The minimum is compared with a threshold (T_2) by a comparator. If the minimum is less than a threshold (T_2) then the central pixel is an edge pixel. In Fig. 5, the hardware structure of this module is illustrated.

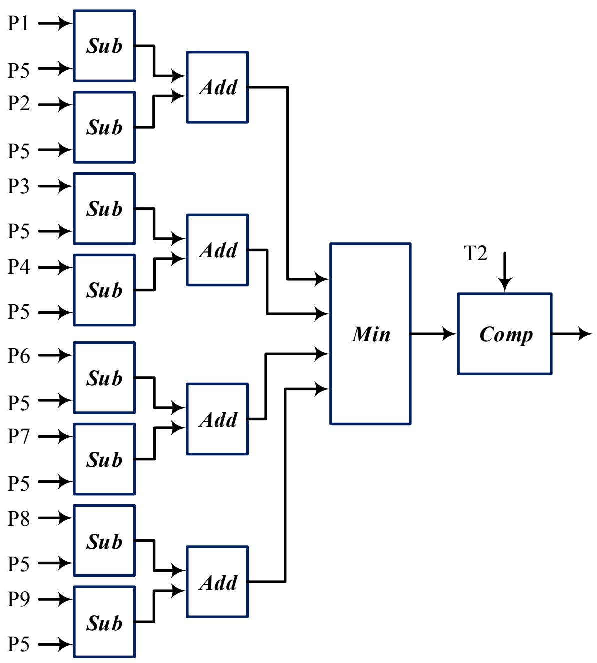


Fig. 5. Edge detector (II) module.

3) *Similarity Checker Module*

To calculate the absolute value of difference between the central pixel and each of its 8 neighbors, we use ABS modules. This is shown in Fig. 6, where eight ABS modules are used for this purpose. Then the results are compared with T_4 using eight comparator units. The positive result from each comparator indicates similarity of the two pixels. Finally sum of similar pixels are added by an adder unit and similarity is obtained with comparison with T_5 by a comparator unit.

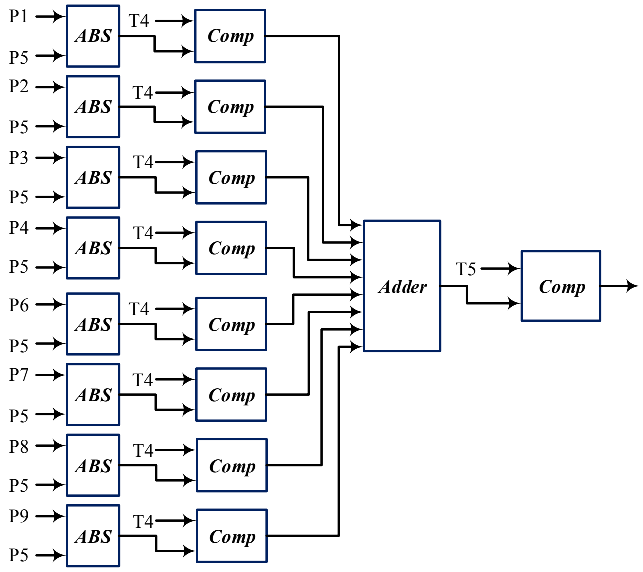


Fig. 6. Similarity checker module

4) *Other hardware modules*

For median and edge preserving filter two architectures in [19] and [15] are used respectively. In image formation the restored pixels are replaced in proper locations in terms of their types and a multiplexer as well as simple wiring is enough for its implementation.

IV. EXPERIMENTAL RESULTS

For verifying the accuracy and low complexity of our proposed method simulation are performed in two stages as follow:

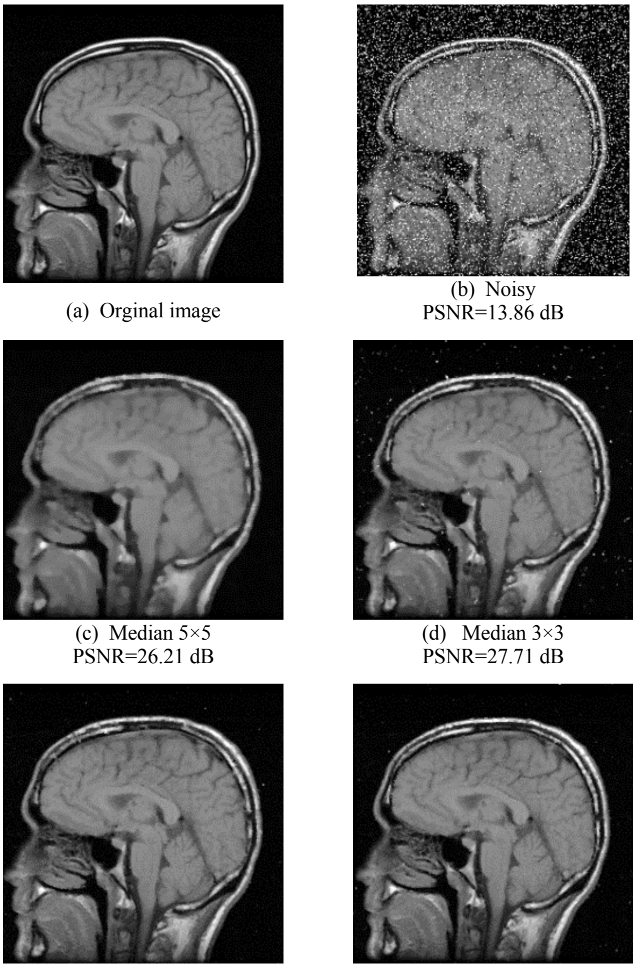
A. Software Simulations

Experiments are performed and verified with MATLAB. In our experiments 100 standard 8-bit gray-scale MR images are used with the size of 256×256 [20]. Noise densities between 5% to 20% are injected uniformly. Objective testing of peak signal-to-noise ratio (PSNR) is used to assess the quality of the restored images. In our proposed method we set thresholds $T_1 = 20$, T_2=100, T_3=30, T_4=10 and T_5=6 and in order to achieve better results, we repeat the algorithm twice. Two hardware architectures for removal of impulse noise in [5,15] and 3×3 and 5×5 median filters are used for comparison. As shown in the Table I, the proposed method has better results than the compared methods for all noise densities.

TABLE I. Comparison between denoised results in terms of PSNR (dB) for different noise densities.

	5%	**10%**	**15%**	**20%**
Median3×3	34.27	33.17	31.14	28.40
Median 5×5	30.18	29.97	29.66	29.28
LCNR[5]	38.27	35.65	32.18	28.65
Ref [15]	36.18	34.93	33.78	32.48
Proposed	**38.65**	**37.07**	**35.43**	**33.52**

In Fig. 7, visual results of the proposed method are compared with other methods. Simulation results in Fig. 7, show that the proposed method has better quality in term of PSNR.



(a) Orginal image

(b) Noisy PSNR=13.86 dB

(c) Median 5×5 PSNR=26.21 dB

(d) Median 3×3 PSNR=27.71 dB

(e) [15] PSNR=29.22 dB

(f) Proposed PSNR=30.92 dB

Fig. 7. Visual quality comparison of the proposed method with [15] and median filtering.

TABLE II. Comparison on implementation specifications between proposed method and methods of [5] and [15].

Method	**Target Device**	**Area**	**Delay (ns)**	**Average PSNR in 5%, 10%, 15% and 20 % noise**
LCNR [5]	Altera Cyclone II EP2C20F484C7 N	513 (Logic cell)	7.72	33.68 dB
Ref [15]	Altera FLEX10KE EPF10K200-SRC240-1	2166 (Logic cell)	14.90	34.34 dB
Proposed	Xilinx virtex4 xc4vfx12 -12-sf363	1841 (Slice)	31.59	36.16 dB

B. Complexity Analysis

For complexity analysis of the proposed method, FPGA implementation is performed. Proposed architecture is first described in VHDL and then it is implemented on a XILINX virtex4 family xc4vfx12 device. Implementation specifications as well as average PSNR for 5%, 10%, 15% and 20 % noise densities are reported and compared with other studies in Table II. It can be seen that the proposed method has better image quality and it has reasonable hardware implementation results.

V. CONCLUSION

In this paper a low complexity noise removal system was proposed for medical images. This method was shown to be suitable for hardware implementation as a part of medical image capturing and transmission devices. The proposed method consisted of two stages of detection and restoration. The goal was to separately improve the accuracy in each of the two stages. High accuracy of noisy-pixel detection in the first stage, and their removal in the next stage, led to better restoration of noisy images. Simulation results using MATLAB software, performed on MR images, showed that the proposed approach removes random value impulse noise with high accuracy. Also low hardware resource utilization of the proposed method makes it suitable for applying it in medical imaging hardware systems.

REFERENCES

[1] T. Mélange, M. Nachtegael, and E. E. Kerre, "Fuzzy random impulse noise removal from color image sequences," *IEEE Transactions on Image Processing*, vol. 20, pp. 959-970, 2011.

[2] V. S. Bhadouria and D. Ghoshal, "A study on genetic expression programming-based approach for impulse noise reduction in images," *Signal, Image and Video Processing*, pp. 1-10, 2015.

[3] B. Xiong and Z. Yin, "A universal denoising framework with a new impulse detector and nonlocal means," *IEEE Transactions on Image Processing*, vol. 21, pp. 1663-1675, 2012.

[4] Z. Zhou, "Cognition and removal of impulse noise with uncertainty," *IEEE Transactions on Image Processing,* vol. 21, pp. 3157-3167, 2012.

[5] T. Matsubara, V. G. Moshnyaga, and K. Hashimoto, "A FPGA implementation of low-complexity noise removal," *in IEEE International Conference on Electronics, Circuits, and Systems (ICECS),* 2010, pp. 255-258.

[6] C.-Y. Lien, C.-C. Huang, P.-Y. Chen, and H.-Y. Yang, "An Efficient Denoising Approach for Random-Valued Impulse Noise in Digital Images," *in International Conference on Innovations in Bio-inspired Computing and Applications (IBICA)*, 2011, pp. 13-16.

[7] K. Srinivasan and D. Ebenezer, "A new fast and efficient decision-based algorithm for removal of high-density impulse noises," *IEEE Signal Processing Letters,* vol. 14, pp. 189-192, 2007.

[8] J. Mandal and S. Mukhopadhyay, "A Novel Variable Mask Median Filter for Removal of Random Valued Impulses in Digital Images (VMM)," *in International Symposium on Electronic System Design (ISED)*, 2011, pp. 302-306.

[9] L. Erasmus, D. Hurter, M. Naudé, H. Kritzinger, and S. Acho, "A short overview of MRI artefacts," *SA Journal of Radiology*, vol. 8, 2004.

[10] 10A. Toprak and İ. Güler, "Angiograph image restoration with the use of rule base fuzzy 2D Kalman filter," *Expert Systems with Applications*, vol. 35, pp. 1752-1761, 2008.

[11] A. Toprak and İ. Güler, "Impulse noise reduction in medical images with the use of switch mode fuzzy adaptive median filter," *Digital signal processing*, vol. 17, pp. 711-723, 2007.

[12] A. Toprak, M. S. Özerdem, and İ. Güler, "Suppression of impulse noise in MR images using artificial intelligent based neuro-fuzzy adaptive median filter," *Digital signal processing*, vol. 18, pp. 391-405, 2008.

[13] P.-Y. Chen, C.-Y. Lien, and H.-M. Chuang, "A low-cost VLSI implementation for efficient removal of impulse noise," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* vol. 18, pp. 473-481, 2010.

[14] H. Hosseini, F. Hessar, and F. Marvasti, "Real-Time Impulse Noise Suppression from Images Using an Efficient Weighted-Average Filtering," *IEEE Signal Processing Letters*, vol. 22, pp. 1050-1054, 2015.

[15] C.-Y. Lien, C.-C. Huang, P.-Y. Chen, and Y.-F. Lin, "An efficient denoising architecture for removal of impulse noise in images," *IEEE Transactions on Computers,* vol. 62, pp. 631-643, 2013.

[16] J. J. Koo, A. C. Evans, and W. J. Gross, "3-D brain MRI tissue classification on FPGAs," *IEEE Transactions on Image Processing*, vol. 18, pp. 2735-2746, 2009.

[17] P.-E. Ng and K.-K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," *IEEE Transactions on Image Processing*, vol. 15, pp. 1506-1516, 2006.

[18] H. Hosseini and F. Marvasti, "Fast restoration of natural images corrupted by high-density impulse noise," *EURASIP Journal on Image and Video Processing*, vol. 2013, pp. 1-7, 2013.

[19] H. Z. HosseinAbadi, S. Samavi, and N. Karimi, "Low Complexity Median Filter Hardware for Image Impulsive Noise Reduction," *Information Systems & Telecommunication*, p. 85, 2014.

[20] http://overcode.yak.net/15?size=M.