



Article New Real-Time Impulse Noise Removal Method Applied to Chest X-ray Images

Nasr Rashid ^{1,2,*}, Kamel Berriri ³, Mohammed Albekairi ¹, Khaled Kaaniche ¹, Ahmed Ben Atitallah ^{1,4}, Muhammad Attique Khan ⁵ and Osama I. El-Hamrawy ^{1,6}

- ¹ Department of Electrical Engineering, College of Engineering, Jouf University, Sakaka 72388, Saudi Arabia
- ² Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Nasr City, Cairo 11884, Egypt
- ³ LAMMDA Laboratory, University of Sousse, Sousse 4054, Tunisia
- ⁴ LETI, ENIS, University of Sfax, Sfax 3038, Tunisia
- ⁵ Department of CS, HITEC University, Taxila 47080, Pakistan
- ⁶ Electrical Engineering Department, Faculty of Engineering, Suez Canal University, Ismailia 41522, Egypt
- * Correspondence: nasrrashid@ju.edu.sa

Abstract: In this paper, we propose a new Modified Laplacian Vector Median Filter (MLVMF) for real-time denoising complex images corrupted by "salt and pepper" impulsive noise. The method consists of two rounds with three steps each: the first round starts with the identification of pixels that may be contaminated by noise using a Modified Laplacian Filter. Then, corrupted pixels pass a neighborhood-based validation test. Finally, the Vector Median Filter is used to replace noisy pixels. The MLVMF uses a 5×5 window to observe the intensity variations around each pixel of the image with a rotation step of $\pi/8$ while the classic Laplacian filters often use rotation steps of $\pi/2$ or $\pi/4$. We see better identification of noise-corrupted pixels thanks to this rotation step refinement. Despite this advantage, a high percentage of the impulsive noise may cause two or more corrupted pixels (with the same intensity) to collide, preventing the identification of noise-corrupted pixels. A second round is then necessary using a second set of filters, still based on the Laplacian operator, but allowing focusing only on the collision phenomenon. To validate our method, MLVMF is firstly tested on standard images, with a noise percentage varying from 3% to 30%. Obtained performances in terms of processing time, as well as image restoration quality through the PSNR (Peak Signal to Noise Ratio) and the NCD (Normalized Color Difference) metrics, are compared to the performances of VMF (Vector Median Filter), VMRHF (Vector Median-Rational Hybrid Filter), and MSMF (Modified Switching Median Filter). A second test is performed on several noisy chest x-ray images used in cardiovascular disease diagnosis as well as COVID-19 diagnosis. The proposed method shows a very good quality of restoration on this type of image, particularly when the percentage of noise is high. The MLVMF provides a high PSNR value of 5.5% and a low NCD value of 18.2%. Finally, an optimized Field-Programmable Gate Array (FPGA) design is proposed to implement the proposed method for real-time processing. The proposed hardware implementation allows an execution time equal to 9 ms per 256×256 color image.

Keywords: MLVMF; impulsive noise; image denoising; chest X-ray images; cardiovascular diseases diagnosis; HLS; FPGA

1. Introduction

The processes of creating, acquiring, saving, and transmitting images often generate noise. A random alteration that an image undergoes can change the intensity of certain pixels to the minimum or maximum value, either 0 or 255. This phenomenon, known as impulsive noise or "salt and pepper" noise, can appear in a digital image due to data transmission errors, the presence of fine particles on the sensor elements of the camera, or faulty memory locations in the storage hardware. The presence of such noise considerably



Citation: Rashid, N.; Berriri, K.; Albekairi, M.; Kaaniche, K.; Ben Atitallah, A.; Khan, M.A.; El-Hamrawy, O.I. New Real-Time Impulse Noise Removal Method Applied to Chest X-ray Images. *Diagnostics* 2022, *12*, 2738. https://doi.org/10.3390/ diagnostics12112738

Academic Editors: Thomas Frauenfelder and Chiara Martini

Received: 2 October 2022 Accepted: 6 November 2022 Published: 9 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). affects the process of understanding the image. Edge detection and segmentation are thus biased. This is the reason why work dedicated to the filtering and denoising of images has, for several years, taken an important place in the field of computer vision.

Various filters for removing noise have emerged. Several approaches have been followed and validated. We can mainly cite the basic vector filters [1–7], the weighted vector filters [8,9], the adaptive vector filters [10–17], the hybrid vector filters [18,19], the fuzzy vector filters [20–22], the neural network based vector filters [23–26], and the morphological based vector median filters [27,28]. The choice of a method remains dependent on the application. In this work, we are looking for a real-time architecture capable of restoring the chest x-ray images as well as possible. This will allow experts to better diagnose possible pathologies such as cardiovascular diseases, asthma, lung cancer, pneumonia, COVID-19, and tumors [29]. Maximum restoration quality also improves the performance of recognition algorithms based on deep learning, multi-channel CNN architecture or angle transformation, which has become very popular in recent years [30–36].

Among the families of filters mentioned above, the basic vector filters, the adaptive vector filters, and the hybrid vector filter are best suited for real-time implementation [17–19,37–40]. These filters are not greedy in terms of resources. This generally involves applying one or more simple heuristics and performing a noisy image scan by calculating a measurement for each pixel (a normalized sum of multiplications), making it possible to distinguish the very strong frequency variations. Methods based on VMF (Vector Median Filter) have always shown very good behavior in the case of low percentage impulsive noise. Methods based on AVMF (Adaptive Vector Median Filter), as well as HVF (Hybrid Vector Filter), are more complex to implement but far better in the case of high percentage impulsive noise [41]. Cited families (based on VMF, AVMF, or HVF) look, firstly, for corrupted pixels: we talk about noisy pixels identification. This step is very important and determines the quality of the image restoration. If some noisy pixels are not identified, they remain unchanged in the final result and disrupt the comprehension process. Otherwise, if non-noisy pixels are identified as being noise, they will be replaced, which negatively affects the quality of the restored image.

In this paper, we propose a new approach to refine the search and identification of noisy pixels. Considering a 5×5 window, our proposed MLVMF uses a modified Laplacian filter to observe the intensity variations around each pixel of the image with a rotation step of $\pi/8$ while the classic Laplacian filters often use rotation steps of $\pi/2$ or $\pi/4$ [42–45]. Reducing the angle of rotation around the pixel makes it possible to take advantage of additional information about its neighborhood. MLVMF does not stop at this stage. Identification of corrupted pixels can still fail against a high percentage of impulsive noise. Indeed, two or more noisy pixels having the same intensity (0 or 255) can be 4-connected or 8-connected. The local second derivative around this kind of pixel will not be able to detect strong variations. Once the noisy pixels are detected thanks to the new proposed filters, then eliminated by a simple VMF, a second round is initiated to search for the adjacent noisy pixels. We propose a second set of filters calculating, in eight directions, the intensity variations around each pixel of the image without taking into account the eight direct neighbors.

To validate our method, MLVMF is tested on standard images with a noise percentage varying from 3% to 30%. Obtained performances in terms of PSNR (Peak Signal to Noise Ratio) and NCD (Normalized Color Difference) are compared to the performances of VMF (Vector Median Filter) [1], VMRHF (Vector Median-Rational Hybrid Filter) [40], and MSMF (Modified Switching Median Filter) [46]. The results obtained show the effectiveness of the proposed approach.

The third and last contribution in this paper is the hardware implementation of the proposed nonlinear approach. Research has adopted hardware acceleration as a solution to reduce implementation complexity. In [47], using two hardware architectures, authors implement standard and multi-level median filters. According to [48], a novel 3×3 window median filtering algorithm is developed based on a bit-serial sorting algorithm with a high

speed of operation and low hardware complexity. In [40], a hardware implementation of the VMRHF for color images is described. The relational function implementation is simplified using some approximations in this hardware architecture. To implement VMF efficiently, authors in [49] suggest a fast parallel architecture. VMF filter implementations in this architecture use approximation to implement L2 norms. However, these hardware architectures need more development time and lack the adaptability of design upgrading. In fact, Low-Level Synthesis (LLS) using Hardware Description Language (HDL) on an FPGA circuit is used to build and implement these systems. With LLS design, the Register Transfer Level (RTL) description can be modified to produce an excellent, efficient netlist. However, creating such an RTL description takes a lot of work and time, especially for complicated applications [50–52]. This is because each low-level circuit's activities must be described. Nevertheless, only hardware designers with specialized knowledge and abilities are capable of creating a sophisticated system. Therefore, in order to decrease the complexity of the FPGA design, it is imperative to go from Low-Level Synthesis (LLS) to High-Level Synthesis (HLS) [53–55]. The HLS tool synthesizes the formalized algorithms into a behavioral and structural RTL hardware description utilizing software high-level languages (systemC, C/C++, etc.). Thereby, several academic and commercial HLS tools are developed, such as Xilinx Vivado HLS, Intel OpenCL, Catapult-C, and ROCCC. Thus, our objective in this part is to use HLS flow to generate various hardware architectures for our proposed denoising algorithm. These architectures are designed in terms of FPGA cost and execution time. The best-designed architecture will be implemented and validated on the Xilinx Zynq FPGA.

This paper is organized as follows. In Section 2, we present the proposed denoising algorithm (MLVMF) by detailing its different stages as well as the filters used. To validate the proposed method, Section 3 shows the MLVMF results tested on standard images with a noise percentage varying from 3% to 30%. Obtained performances in terms of processing time, as well as image restoration quality through the PSNR and the NCD measurement, are compared to the performances of VMF, VMRHF, and MSMF. An additional test is performed on noisy chest x-ray images used in cardiovascular disease diagnosis as well as COVID-19 diagnosis. Finally, the HLS designs for the MLVMF are described in Section 4.

2. Modified Laplacian Vector Median Filter—MLVMF

A noisy pixel is always associated with a high frequency in its direct neighborship. Impulsive noise is no exception to this rule. To identify a noisy pixel, it is, therefore, necessary to calculate the variations in intensity locally. A double derivative around the pixel makes it possible to measure this variation: this is the Laplacian operator. Let I be an image of size $n \times m$. Let (x,y) be the position of a pixel in the image *I*. The Laplacian L(x,y) of an image with pixel intensity values I(x, y) is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$
(1)

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the Laplacian. Two commonly used small kernels are shown in Figure 1.

0	-1	0	-1	-1	-1
-1	4	-1	-1	8	-1
0	-1	0	-1	-1	-1

Figure 1. Two commonly used discrete approximations to the Laplacian filter.

When searching for and identifying noisy pixels of the impulsive type, there is a tendency to widen the Laplacian filters. In general, these filters are of size 5×5 instead of 3×3 . This is dictated by the concern to ensure that it is indeed an isolated pixel in terms of intensity. For these same reasons and contrary to the search for edges, the convolution is not conducted all at once but rather in several directions around the pixel. Figure 2 shows the four commonly used kernels in the case of impulsive noise identification.

0	0	0	0	0	0	0	0	0	-1	0	0	-1	0	0	-1	0	0	0	0
0	0	0	0	0	0	0	0	-1	0	0	0	-1	0	0	0	-1	0	0	0
-1	-1	4	-1	-1	0	0	4	0	0	0	0	4	0	0	0	0	4	0	0
0	0	0	0	0	0	-1	0	0	0	0	0	-1	0	0	0	0	0	-1	0
0	0	0	0	0	-1	0	0	0	0	0	0	-1	0	0	0	0	0	0	-1

Figure 2. Four commonly used kernels in the case of impulsive noise identification. Angle of rotation (from left to right): $0, \pi/4, \pi/2, 3\pi/4$.

As shown in Figure 2, the rotation step of the Laplacian is equal to $\frac{\pi}{4}$: kernel K_1 at 0, kernel K_2 at $\frac{\pi}{4}$, kernel K_3 at $\frac{\pi}{2}$, and kernel K_4 at $\frac{3\pi}{4}$. A smaller rotation step allows a deeper investigation of the neighborhood of a pixel. However, the rectangular aspect of the pixels and the small size of the search window show no more than four discernible directions. Increasing the size of the window will increase the calculation time. It is nevertheless possible to approximate new directions. We propose, in Figure 3, four new kernels in order to estimate the variations of intensity according to the directions $\frac{\pi}{8}$, $\frac{3\pi}{8}$, $\frac{5\pi}{8}$, and $\frac{7\pi}{8}$. New kernels are: kernel K_5 at $\frac{\pi}{8}$, kernel K_6 at $\frac{3\pi}{8}$, kernel K_7 at $\frac{5\pi}{8}$, and kernel K_8 at $\frac{7\pi}{8}$.

0	0	0	0	0	0	0	-1	-1	0	0	-1	-1	0	0	0	0	0	0	0
0	0	0	-1	-1	0	0	-1	-1	0	0	-1	-1	0	0	-1	-1	0	0	0
-1	-1	8	-1	-1	0	0	8	0	0	0	0	8	0	0	-1	-1	8	-1	-1
-1	-1	0	0	0	0	-1	-1	0	0	0	0	-1	-1	0	0	0	0	-1	-1
0	0	0	0	0	0	-1	-1	0	0	0	0	-1	-1	0	0	0	0	0	0

Figure 3. Proposed four new kernels according to the directions $\pi/8$, $3\pi/8$, $5\pi/8$, and $7\pi/8$.

Identification of corrupted pixels can still fail against a high percentage of impulsive noise. Indeed, two or more noisy pixels having the same intensity (0 or 255) can be 4-connected or 8-connected. The local second derivative around this kind of pixel will not be able to detect strong variations. We propose a second set of filters calculating in eight directions the intensity variations around each pixel of the image without taking into account the eight direct neighbors. Figure 4 shows the proposed eight kernels: K_9 to K_{16} .

0	0	0	0	0	0	0	0	0	-1	0	0	0	-1	-1	0	0	-1	-1	0
0	0	0	0	-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0
-1	0	4	0	-1	0	0	4	0	0	0	0	4	0	0	0	0	4	0	0
-1	0	0	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-1	0	0	0	0	-1	-1	0	0	0	0	-1	-1	0	0
0	-1	-1	0	0	-1	-1	0	0	0	-1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	-1	0	0	0	0
0	0	4	0	0	0	0	4	0	0	0	0	4	0	0	-1	0	4	0	-1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	-1
0	0	-1	-1	0	0	0	0	-1	-1	0	0	0	0	-1	0	0	0	0	0

Figure 4. Eight new kernels proposed without taking into account the eight direct neighbors, K_9 to K_{16} .

The calculation of the variation is carried out as follows. For each position (x, y) in the image *I*, we compute the absolute value of the convolution product noted V_{ij} between the kernel K_i and the image I_j .

$$V_{ij}(x,y) = |K_i \otimes I_j|, \qquad (2)$$

where *j* varies from 1 to 3, such as I_1 is the image red component, I_2 is image green component, and I_3 is the image blue component. A first scan of the image is performed with filters K_1 to K_8 . For each image pixel, we, therefore, obtain 24 measurements of variation. To judge the importance of the intensity variation around a pixel (x, y), we observe the minimum value $M_1(x, y)$ among the 24 measurements obtained.

$$M_1(x,y) = \min[V_{ij}(x,y), i = 1..8, j = 1..3],$$
(3)

If $M_1(x, y)$ exceeds a predefined threshold *T* then the pixel (x, y) is considered a candidate to be an impulsive noise. At this precise moment, we retrieve the color layer of the image giving this minimum measurement and declare the pixel as noise if its intensity at this layer is 0 or 255.

$$\begin{split} M_1(x,y) &= \min[V_{ij}(x,y), \ i = 1..8, \ j = 1..3] \\ \textit{if } M_1(x,y) > T \textit{ then } (i_{ind}, j_{ind}) = \textit{index} (\min[V_{ij}(x,y), \ i = 1..8, \ j = 1..3]) \\ \textit{if } I_{j_{ind}}(x,y) = 0 \textit{ or } 255 \textit{ then } \textit{pixel } (x,y) \textit{is an impulsive noise} \end{split}$$

To eliminate the noisy pixel, we use the VMF method [1]. The image obtained and noted I_c is then scanned again. This scan is performed with filters K_9 to K_{16} .

$$V_{ij}(x,y) = |K_i \otimes I_{cj}|, \quad i = 9..16,$$
 (4)

where *j* varies from 1 to 3, such as I_{c1} is the image red component of I_c , I_{c2} is image green component of I_c , and I_{c3} is the image blue component of I_c .

For each image pixel, we also obtain 24 measurements of variation. We observe the minimum value $M_2(x, y)$ among the 24 measurements obtained.

$$M_2(x,y) = \min[V_{ij}(x,y), i = 9..16, j = 1..3],$$
(5)

To declare a pixel as noise, we proceed in the same way as for the first scan using the same value of predefined threshold T. To eliminate the noisy pixel, we use the VMF method [1]. Figure 5 shows the MLVMF flowchart.



Figure 5. MLVMF flowchart.

3. MLVMF Performances

The proposed MLVMF filter is assessed in terms of image restoration quality and execution time to remove the impulsive "salt and pepper" in color images. We compare MLVMF performances to the performances of VMF, VMRHF, and MSMF filters. The

metrics used in this comparative study are: The Peak Signal to Noise Ratio (PSNR) and the Normalized Color Difference (NCD). PSNR is defined by (6)–(8).

$$PSNR = 10 \, \log\left(\frac{255^2}{MSE}\right),\tag{6}$$

$$MSE_{l} = \frac{1}{nm} \sum_{i=1}^{n} \sum_{j=1}^{m} (I_{l}(i,j) - I_{cl}(i,j))^{2},$$
(7)

$$MSE = \frac{MSE_r + MSE_g + MSE_b}{3},$$
(8)

where *l* the index of the color channel of a color image (r = red, g = green, b = blue). *I* represents the original image. *I*_c is the filtered image. *n* is the number of rows of an image and *m* the number of columns.

Color distortion is estimated using the NCD, which is defined by (9).

$$NCD = \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} \sqrt{(Y(i,j) - Y_c(i,j))^2 + (U(i,j) - U_c(i,j))^2 + (V(i,j) - V_c(i,j))^2}}{\sum_{i=1}^{n} \sum_{j=1}^{m} \sqrt{(Y(i,j))^2 + (U(i,j))^2 + (V(i,j))^2}}, \quad (9)$$

where *Y* represents the original image luminance, Y_c is the filtered image luminance, *U* and *V* are original image chrominance components, U_c and V_c are filtered image chrominance components.

The proposed MLVMF is developed using C/C++ programming language, compiled by Visual C++ 2010 software tool. Execution is performed under CoreTMi7-1165g7@2.80ghz Intel processor. The processor timer is used to determine the execution time. We choose five test images (Figure 6), including two standard images (Lena and Peppers) and three chest x-ray images (CXR1, CXR2, and CXR3) [56]. All images used are 256×256 in size. Test images are corrupted by various levels of "salt and pepper" impulsive noise. These levels are 3%, 5%, 10%, 20%, and 30%. Based on several experimental tests, we set the threshold *T* at 75.

Table 1 shows the values of quality measurements according to PSNR and NCD after the filtering process performed by VMF, VMRHF, MSMF, and proposed MLVMF filters. According to this table, we can notice that the MLVMF is more efficient than the other three filters. Compared to the MSMF, which has the best performance among the three filters in question in terms of PSNR, the MLVMF provides a high PSNR value of 5.5%. Compared to the VMF, which has the best performance among the same three filters in terms of NCD, the MLVMF provides a low NCD value of 18.2%. Moreover, subjective measurement confirms the efficiency of our proposed MLVMF. Figure 6 confirms that the filtered images (Lena, Peppers, CRX1, CRX2, and CRX3) obtained after MLVMF processing are too close to the original images, thanks to a better quality of restoration. Figure 7 shows the comparison of performances measured in terms of PSNR and NCD metrics using Lena, Peppers, CRX1, CRX2, and CRX3 images for various levels of impulsive noise (3%, 5%, 10%, 20%, and 30%). In addition to the fact that the MLVMF is clearly superior in the case of a low and medium noise percentage, the proposed filter is clearly distinguished from other filters when the noise is high. This mainly comes down to the investigation carried out by the filters K9 to K16 and which succeeded in identifying the adjacent corrupted pixels. The execution time of the proposed MLVMF under CoreTMi7-1165g7@2.80ghz Intel processor reaches 1483 ms per 256×256 color image. To decrease this execution time and allow real-time calculation, Section 4 describes a proposed High-Level Synthesis (HLS) design for the MLVMF filter.



Figure 6. (a) Original images. (b) Corrupted images with 10% impulsive noise. Filtered images using (c) VMF, (d) VMRHF, (e) MSMF, and (f) MLVMF filters.

Filt	ter	VN	ИF	VM	RHF	MS	MF	MLVMF		
Image	Noise	PSNR (dB)	PSNR (dB) NCD		NCD	PSNR (dB)	NCD	PSNR (dB)	NCD	
	3%	32.340147	0.017115	32.782660	0.015688	35.472474	0.004766	37.423460	0.0042894	
	5%	31.728372	0.018811	32.100663	0.017784	33.911660	0.008693	35.844625	0.0079106	
Lena	10%	29.577089	0.023299	29.734905	0.024349	30.400635	0.017641	32.087870	0.0159651	
	20%	25.232852	0.041481	25.244443	0.052842	25.084609	0.049493	26.414093	0.0345932	
	30%	20.856752	0.096637	21.130595	0.118374	20.657001	0.116282	21.731165	0.0955841	
	3%	34.518151	0.012914	35.044101	0.011567	36.409296	0.005163	38.375398	0.0045951	
	5%	33.217095	0.014426	33.596334	0.013441	34.123878	0.009285	36.068939	0.0084494	
Peppers	10%	30.370972	0.018116	30.448538	0.019125	30.270752	0.020890	32.026456	0.0181743	
	20%	25.730744	0.035530	25.695894	0.046225	25.121437	0.057862	26.477995	0.0314972	
	30%	21.196992	0.083353	21.466820	0.103619	20.905560	0.117288	22.034460	0.0764975	
	3%	33.637944	0.007440	33.797730	0.008316	35.523118	0.002363	37.476889	0.0021267	
	5%	32.265345	0.008059	32.343503	0.010048	33.248306	0.004593	35.209956	0.0041796	
CXR1	10%	29.716387	0.011066	29.690418	0.016741	29.858881	0.011025	31.590696	0.0099776	
	20%	24.991928	0.027388	24.956495	0.046715	24.704409	0.038274	25.939629	0.0274849	
	30%	20.404829	0.081102	20.680126	0.113743	20.243625	0.101027	21.296294	0.0717325	
	3%	35.571872	0.009196	35.826513	0.007403	36.795335	0.003027	38.782283	0.0026940	
	5%	33.170338	0.010195	33.264613	0.008741	33.436260	0.005518	35.342127	0.0050214	
CXR2	10%	30.882192	0.011984	30.774950	0.013983	30.471932	0.011975	32.239304	0.0104183	
	20%	25.765780	0.025517	25.680567	0.040658	25.238697	0.033607	26.601587	0.0239102	
	30%	20.448200	0.079880	20.697093	0.112506	20.281939	0.091909	21.377164	0.0734534	
	3%	37.002201	0.004489	37.195887	0.004118	38.126856	0.002395	40.223833	0.0021555	
	5%	34.776916	0.005016	34.799607	0.005191	35.126686	0.004735	37.128907	0.0043089	
CXR3	10%	31.703067	0.006751	31.494476	0.009411	31.463504	0.010426	33.209728	0.0054355	
	20%	26.207766	0.021795	26.062978	0.038572	25.760547	0.036464	27.125856	0.0188541	
	30%	21.351850	0.077362	21.600711	0.107416	21.129428	0.097884	22.228158	0.0738787	
Average	-	28.906630	0.029957	29.044420	0.038663	29.350670	0.034503	30.970270	0.0253275	

Table 1. PSNR and NCD values of the VMF, VMRHF, MSMF, and MLVMF filters for various levels ofimpulsive noise.



Figure 7. Cont.





Figure 7. Performance comparison of the VMF, VMRHF, MSMF, and MLVMF filters using Lena, Peppers, CXR1, CXR2, and CXR3 images for various levels of impulsive noise. Left column: PSNR(dB) performance comparison. Right column: NCD performance comparison. Impulsive noise levels are: 3%, 5%, 10%, 20%, and 30%.

4. High-Level Synthesis (HLS) Designs for the MLVMF Filter

The HLS flow used with FPGA is an important tool for engineers to quickly explore the design space from a given behavioral description based on a high-level programming language (e.g., SystemC, C/C++). Because of this, the HLS has become a useful and effective tool for boosting design productivity and decreasing design cycle time. Various HLS tools have been developed in this regard, such as the Xilinx Vivado HLS tool, which offers a number of directives to produce an optimum hardware design for any algorithm. In fact, the RESOURCE directive allows the arrays to be implemented as registers or memories. Additionally, the ALLOCATION directive can be used to optimize how the arithmetic operation is implemented. Furthermore, using the PIPELINE and UNROLL directives, the loops can be pipelined, not unrolled, or fully/partially unrolled to increase the loop iterations' performance (i.e., to reach a higher throughput).

Figure 8 shows the block diagram of the hardware architecture generated for the MLVMF filter. The generation process is performed by Xilinx Vivado HLS 18.1 tool from a specific C/C++ code. Our architecture uses 5 DMA (Direct Memory Access) to transfer five image lines in parallel in order to increase the throughput of our MLVMF filter. The hardware architecture of the proposed MLVMF is based on a 5×5 modified Laplacian filter and a 3×3 VMF filter. Each denoising RGB pixel (in 24 bit) is concatenated and stored in $256 \times 256 \times 24$ -bit internal memory. To optimize the architecture, some directives (such as PARTITION and PIPELINE) are incrementally introduced to the MLVMF C/C++ code. So, we produce various MLVMF hardware designs. We then retain the optimized design, which gives a compromise between hardware cost and processing time. The hardware requirements in terms of Lookup-Table (LUT), Flip-Flops (FF), BRAM blocks, and DSP blocks for the various MLVMF designs on the Zynq XCZU9EG FPGA are shown in Table 2.

Table 2. FPGA resources and performances of the MLVMF-designed architectures.

	LUT	FF	BRAM_18K	DSP48E	Cycles
Solution1	10,687 (4%)	6590 (1%)	125 (7%)	13 (0.5%)	209,546,135
Solution2	43,343 (16%)	42,228 (8%)	125 (7%)	650 (26%)	5,304,902
Solution3	45,939 (17%)	43,685 (8%)	135 (8%)	665 (26%)	1,135,576

In Table 2, Solution 1 presents the hardware architecture without any optimization. It is clear that this solution is not greedy in terms of resources but requires a very large number of clock cycles to filter the entire image. With a clock frequency equal to 115 MHz, Solution 1 needs 1810 ms per 256×256 color image. In Solution 2, we propose to apply the PIPELINE directive. Hardware cost increases but is still well below the maximum capacities of the hardware. On the other hand, the number of clock cycles decreases by 97.5%. An image can, therefore, be processed in 46 ms. With the aim of further reducing the number of clock cycles, we add in Solution 3 the PARTITION directive, which allows splitting the filtering window (Figure 8) into small blocks and promoting parallel access to data. Thank to Solution 3, we reach a computation time equal to 9 ms, i.e., 200 times faster than Solution 1. Hardware resources will be consumed more but remain well below hardware capacity. Finally, it should be noted that the quality of the filtering with Solution 2 and Solution 3 are the same obtained during the software implementation of Section 3.



Figure 8. Proposed MLVMF filter block diagram.

5. Conclusions

In this work, a new Modified Laplacian Vector Median Filter (MLVMF) for real-time denoising complex images corrupted by "salt and pepper" impulsive noise is proposed. The aim is to provide a real-time architecture capable of restoring the chest x-ray images as well as possible. This will allow experts to better diagnose possible pathologies such as cardiovascular diseases, asthma, lung cancer, pneumonia, COVID-19, and tumors. We propose two sets of modified Laplacian filters to identify corrupted pixels. The first set investigates images using a smaller rotation step compared to existing methods. This allows better detection of corrupted pixels. The second set of proposed filters allows the detection of adjacent noisy pixels. These pixels go unnoticed in front of conventional filters, which leads to a bad reconstruction of the image when the percentage of impulsive noise is high. New High-Level Synthesis (HLS) Designs for the proposed MLVMF Filter are presented. The proposed MLVMF shows better performances compared to VMF, VMRHF, and MSMF filters in terms of PSNR and NCD measurements. In fact, The MLVMF provides a high PSNR value of 5.5% and a low NCD value of 18.2%. A proposed Hardware implementation based on High-Level Synthesis (HLS) designs of the MLVMF enabled real-time processing allowing an execution time equal to 9 ms per 256×256 color image.

Our method allows, as shown by the experimental results, a very good detection of impulsive noise. However, the work presented here does not deal with the operation of replacing the corrupted pixel. To eliminate a noisy pixel after having identified it, we use the VMF approach. We believe that a more intelligent estimation of the true color of the noisy pixel is possible. However, this estimate can be costly in terms of resources and computation time. Future work will focus on this point because a better estimate of the true color of a corrupted pixel, added to the very good identification presented in this work, can bring us closer to the ideal denoising.

Author Contributions: Conceptualization, A.B.A. and O.I.E.-H.; methodology, K.K., N.R. and M.A.; software, A.B.A., O.I.E.-H. and N.R.; validation, K.K., M.A.K. and M.A.; formal analysis, K.B.; investigation, N.R. and K.K.; resources, O.I.E.-H. and M.A.; data curation, A.B.A.; writing—original draft preparation, O.I.E.-H., N.R. and K.B.; writing—review and editing, O.I.E.-H.; visualization, M.A. and M.A.K.; supervision, A.B.A. and O.I.E.-H.; project administration, K.K.; funding acquisition, N.R. All authors have read and agreed to the published version of the manuscript.

Funding: The authors extend their appreciation to the Deanship of Scientific Research at Jouf University for funding this work through research grant No (DSR-2020-04-480).

Conflicts of Interest: The authors declare no conflict of interest.

References

- 1. Astola, J.; Haavisto, P.; Neuvo, Y. Vector median filters. Proc. IEEE 1990, 78, 678–689. [CrossRef]
- Smolka, B. Robust Sharpening Vector Median Filter. In Proceedings of the International Automatic Control Conference (CACS), Taoyuan, Taiwan, 4–7 November 2018; pp. 1–6. [CrossRef]
- Smolka, B.; Szczepanski, M.; Plataniotis, K.; Venetsanopoulos, A. On the fast modified vector median filter. In Proceedings of the Canadian Conference on Electrical and Computer Engineering 2001, Toronto, ON, Canada, 13–16 May 2001; Volume 2, pp. 1315–1320. [CrossRef]
- Viero, T.; Oistamo, K.; Neuvo, Y. Three-dimensional median-related filters for color image sequence filtering. *IEEE Trans. Circuits* Syst. Video Technol. 1994, 4, 129–142. [CrossRef]
- 5. Lukac, R.; Smolka, B.; Plataniotis, K.N. Sharpening vector median filters. Signal Process. 2007, 87, 2085–2099. [CrossRef]
- Trahanias, P.E.; Venetsanopoulos, A.N. Vector directional filters-a new class of multichannel image processing filters. *IEEE Trans. Image Process.* 1993, 2, 528–534. [CrossRef]
- Karakos, D.; Trahanias, P. Combining vector median and vector directional filters: The directional-distance filters. In Proceedings of the International Conference on Image Processing, Washington, DC, USA, 23–26 October 1995; Volume 1, pp. 171–174. [CrossRef]
- Lukac, R. Another generalisation of vector filters. In Proceedings of the International Symposium on VIPromCom Video/Image Processing and Multimedia Communications, Zadar, Croatia, 16–19 June 2002; pp. 273–278. [CrossRef]
- 9. Smolka, B. Efficient Modification of the Central Weighted Vector Median Filter. In *Joint Pattern Recognition Symposium*; DAGM 2002; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany; Volume 2449. [CrossRef]

- 10. Lukac, R. Adaptive vector median filtering. Pattern Recognit. Lett. 2003, 24, 1889–1899. [CrossRef]
- Plataniotis, K.; Androutsos, D.; Venetsanopoulos, A. Color image processing using adaptive vector directional filters. In *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 1998; Volume 45, pp. 1414–1419. [CrossRef]
- Singh, K.; Bora, P.; Singh, S. Rank-ordered mean filter for removal of impulse noise from images. In Proceedings of the 2002 IEEE International Conference on Industrial Technology, 2002. IEEE ICIT '02., Bangkok, Thailand, 11–14 December 2002; Volume 2, pp. 980–985. [CrossRef]
- Peris-Fajarnés, G.; Roig, B.; Vidal, A. Rank-Ordered Differences Statistic Based Switching Vector Filter. In *Image Analysis and Recognition, Proceedings of the Third International Conference, ICIAR 2006, Póvoa de Varzim, Portugal, 18–20 September 2006;* Lecture Notes in Computer Science; Campilho, A., Kamel, M.S., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; Volume 4141.
 [CrossRef]
- 14. Morillas, S.; Gregori, V.; Sapena, A. Adaptive Marginal Median Filter for Colour Images. Sensors 2011, 11, 3205–3213. [CrossRef]
- 15. Singh, K.M.; Bora, P.K. Switching vector median filters based on non-causal linear prediction for detection of impulse noise. *Imaging Sci. J.* **2013**, *62*, 313–326. [CrossRef]
- Roy, A.; Singha, J.; Manam, L.; Laskar, R.H. Combination of adaptive vector median filter and weighted mean filter for removal of high density impulse noise from color images. *IET Image Process.* 2017, *11*, 352–361. [CrossRef]
- Das, J.B.A.; Sarangi, A.; Mishra, D.; Mohanty, M.N. Design of RAMF for Impulsive Noise Cancelation from Chest X-Ray Image. In *Advances in Intelligent Computing and Communication, Lecture Notes in Networks and Systems*; Springer: Singapore, 2022; Volume 430. [CrossRef]
- 18. Khriji, L.; Gabbouj, M. Adaptive fuzzy order statistics-rational hybrid filters for color image processing. *Fuzzy Sets Syst.* **2002**, *128*, 35–46. [CrossRef]
- Abid, I.; Boudabous, A.; Ben Atitallah, A. A new adaptive vector median rational hybrid filter for impulsive noise suppression. In Proceedings of the 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Monastir, Tunisia, 21–23 December 2015; pp. 417–421. [CrossRef]
- Plataniotis, K.; Androutsos, D.; Venetsanopoulos, A. Adaptive fuzzy systems for multichannel signal processing. *Proc. IEEE* 1999, 87, 1601–1622. [CrossRef]
- 21. Wang, G.; Zhu, H.; Wang, Y. Fuzzy decision filter for color images denoising. Optik 2015, 126, 2428–2432. [CrossRef]
- 22. Roy, A.; Manam, L.; Laskar, R.H. Region Adaptive Fuzzy Filter: An Approach for Removal of Random-Valued Impulse Noise. *IEEE Trans. Ind. Electron.* 2018, 65, 7268–7278. [CrossRef]
- 23. Kong, H.; Guan, L. A neural network adaptive filter for the removal of impulse noise in digital images. *Neural Netw.* **1996**, *9*, 373–378. [CrossRef]
- 24. Solovyeva, E. Cellular neural network as a non-linear filter of impulse noise. In Proceedings of the 2017 20th Conference of Open Innovations Association (FRUCT), St. Petersburg, Russia, 3–7 April 2017; pp. 420–426. [CrossRef]
- Radlak, K.; Malinski, L.; Smolka, B. Deep Learning Based Switching Filter for Impulsive Noise Removal in Color Images. Sensors 2020, 20, 2782. [CrossRef] [PubMed]
- 26. Liang, S.-F.; Lu, S.-M.; Chang, J.-Y.; Lin, C.-T. A Novel Two-Stage Impulse Noise Removal Technique Based on Neural Networks and Fuzzy Decision. *IEEE Trans. Fuzzy Syst.* 2008, 16, 863–873. [CrossRef]
- Naveen, Y.; Gupta, S. Removal of Salt and Pepper Noise Using Sparse Representation. In Proceedings of the 2012 Eighth International Conference on Signal Image Technology and Internet Based Systems, Naples, Italy, 26–29 November 2012; pp. 58–63. [CrossRef]
- Lei, T.; Wang, Y.; Luo, W. Multivariate Self-Dual Morphological Operators Based on Extremum Constraint. *Math. Probl. Eng.* 2015, 2015, 596348. [CrossRef]
- 29. Das, P.; Mukherjee, S.; Das, P.; Banerjee, S. Characterizing chaos and multifractality in noise-assisted tumor-immune interplay. *Nonlinear Dyn.* **2020**, *101*, 675–685. [CrossRef]
- Sato, Y.; Yamamoto, N.; Inagaki, N.; Iesaki, Y.; Asamoto, T.; Suzuki, T.; Takahara, S. Deep Learning for Bone Mineral Density and T-Score Prediction from Chest X-rays: A Multicenter Study. *Biomedicines* 2022, 10, 2323. [CrossRef]
- Ramadhan, A.A.; Baykara, M. A Novel Approach to Detect COVID-19: Enhanced Deep Learning Models with Convolutional Neural Networks. *Appl. Sci.* 2022, 12, 9325. [CrossRef]
- 32. Sharma, N.; Saba, L.; Khanna, N.N.; Kalra, M.K.; Fouda, M.M.; Suri, J.S. Segmentation-Based Classification Deep Learning Model Embedded with Explainable AI for COVID-19 Detection in Chest X-ray Scans. *Diagnostics* 2022, 12, 2132. [CrossRef]
- 33. Kaya, Y.; Yiner, Z.; Kaya, M.; Kuncan, F. A new approach to COVID-19 detection from X-ray images using angle transformation with GoogleNet and LSTM. *Meas. Sci. Technol.* **2022**, *33*, 124011. [CrossRef]
- Eken, S. A topic-based hierarchical publish/subscribe messaging middleware for COVID-19 detection in X-ray image and its metadata. *Soft Comput.* 2020, 24, 1–11. [CrossRef] [PubMed]
- Yilmaz, A. Diagnosing COVID-19 from X-Ray images with using multi-channel CNN architecture. J. Fac. Eng. Archit. Gazi Univ. 2021, 36, 1761–1774. [CrossRef]
- Nair, R.; Alhudhaif, A.; Koundal, D.; Doewes, R.I.; Sharma, P. Deep learning-based COVID-19 detection system using pulmonary CT scans. *Turk. J. Electr. Eng. Comput. Sci.* 2021, 29, 2716–2727. [CrossRef]
- 37. Gökcen, A.; Kalyoncu, C. Real-time impulse noise removal. J. Real-Time Image Process. 2018, 17, 459–469. [CrossRef]

- HosseinKhani, Z.; Hajabdollahi, M.; Karimi, N.; Soroushmehr, R.; Shirani, S.; Najarian, K.; Samavi, S. Adaptive Real-Time Removal of Impulse Noise in Medical Images. J. Med. Syst. 2018, 42, 216. [CrossRef]
- 39. Ben Atitallah, A. An FPGA Design for Real-Time Image Denoising. Comput. Syst. Sci. Eng. 2022, 43, 803–816. [CrossRef]
- 40. Boudabous, A.; Ben Atitallah, A.; Khriji, L.; Kadionik, P.; Masmoudi, N. HW/SW design-based implementation of vector median rational hybrid filter. *Int. Arab J. Inf. Technol.* **2010**, *7*, 70–74.
- Ben Atitallah, A.; Abid, I. An efficient FPGA implementation of AVMF filter using High-Level Synthesis. In Proceedings of the 2020 20th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA), Sfax, Tunisia, 20–22 December 2020; pp. 82–85. [CrossRef]
- Kim, J.; Wills, D.S. Fast vector median filter implementation using the color pack instruction set. In Proceedings of the 2002 IEEE 10th Digital Signal Processing Workshop, and the 2nd Signal Processing Education Workshop, Pine Mountain, GA, USA, 13–16 October 2002; pp. 339–343. [CrossRef]
- Khriji, L. Vector Directional Distance Rational Hybrid Filters for Color Image Restoration. J. Eng. Res. [TJER] 2005, 2, 1–12. [CrossRef]
- 44. Ben Atitallah, A. A New Adaptive Filter to Remove Impulsive Noise in Color Images. *IEEJ Trans. Electr. Electron. Eng.* 2022, 17, 1048–1053. [CrossRef]
- 45. Pilevar, A.H.; Saien, S.; Khandel, M.; Mansoori, B. A new filter to remove salt and pepper noise in color images. *Signal Image Video Process.* **2013**, *9*, 779–786. [CrossRef]
- 46. Wang, G.; Li, D.; Pan, W.; Zang, Z. Modified switching median filter for impulse noise removal. *Signal Process.* **2010**, *90*, 3213–3218. [CrossRef]
- Hu, Y.; Ji, H. Research on Image Median Filtering Algorithm and Its FPGA Implementation. In Proceedings of the 2009 WRI Global Congress on Intelligent Systems, Washington, DC, USA, 19–21 May 2009; pp. 226–230. [CrossRef]
- Lee, T.W.; Lee, J.H.; Cho, S.B. FPGA Implementation of a 3/spl times/3 window median filter based on a new efficient bit-serial sorting algorithm. In Proceedings of the 7th Korea-Russia International Symposium on Science and Technology, Proceedings KORUS 2003. (IEEE Cat. No.03EX737), Ulsan, Korea, 28 June–6 July 2003; Volume 2, pp. 237–242.
- Boudabous, A.; Khriji, L.; Ben Atitallah, A.; Kadionik, P.; Masmoudi, N. Efficient architecture and implementation of vector median filter in co-design context. *RadioEng.-Prague* 2007, 16, 113–119.
- Kthiri, M.; Le Gal, B.; Kadionik, P.; Ben Atitallah, A. A Very High Throughput Deblocking Filter for H.264/AVC. J. Signal Process. Syst. 2013, 73, 189–199. [CrossRef]
- Ben Atitallah, A.; Kadionik, P.; Ghozzi, F.; Nouel, P.; Masmoudi, N.; Marchegay, P. Optimization and Implementation on Fpga of the DCT/IDCT Algorithm. In Proceedings of the 2006 IEEE International Conference on Acoustics Speech and Signal Processing, Toulouse, France, 14–19 May 2006; p. III. [CrossRef]
- 52. Ben Atitallah, A.; Abid, I.; Boudabous, A.; Loukil, H. A new hardware architecture of the adaptive vector median filter and validation in a hardware/software environment. *Int. J. Circuit Theory Appl.* **2021**, *49*, 2329–2347. [CrossRef]
- Ben Atitallah, A.; Kammoun, M.; Ali, K.M.; Ben Atitallah, R. An FPGA comparative study of high-level and low-level combined designs for HEVC intra, inverse quantization, and IDCT/IDST 2D modules. *Int. J. Circuit Theory Appl.* 2020, 48, 1274–1290. [CrossRef]
- 54. Kammoun, M.; Ben Atitallah, A.; Ali, K.M.A.; Ben Atitallah, R. Case study of an HEVC decoder application using high-level synthesis: Intraprediction, dequantization, and inverse transform blocks. *J. Electron. Imaging* **2019**, *28*, 033010. [CrossRef]
- Alanazi, T.M.; Ben Atitallah, A.; Abid, I. An Optimized SW/HW AVMF Design Based on High-Level Synthesis Flow for Color Images. Comput. Mater. Contin. 2021, 68, 2925–2943. [CrossRef]
- 56. Winther, H.B.; Laser, H.; Gerbel, S.; Maschke, S.K.; Hinrichs, J.B.; Vogel-Claussen, J.; Wacker, F.K.; Höper, M.M.; Meyer, B.C. COVID-19 Image Repository. *Figshare Dataset* **2020**. [CrossRef]