This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

A Robust Edge Detection Approach in the Presence of High Impulse Noise Intensity through Switching Adaptive Median and Fixed Weighted Mean Filtering

Mehdi Mafi, Hoda Rajaei, Mercedes Cabrerizo and Malek Adjouadi

Center for Advanced Technology and Education,
Department of Electrical and Computer Engineering,
Florida International University

***Abstract*— This study introduces a robust edge detection method that relies on an integrated process for denoising images in the presence of high impulse noise. This process is shown to be resilient to impulse (or salt and pepper) noise even under high intensity levels. The proposed switching adaptive median and fixed weighted mean filter (SAMFWMF) is shown to yield optimal edge detection and edge detail preservation, an outcome we validate through high correlation, structural similarity index and peak signal to noise ratio measures. For comparative purposes, a comprehensive analysis of other denoising filters is provided based on these various validation metrics. The non-maximum suppression method and new edge following maximum-sequence are two techniques used to track the edges and overcome edge discontinuities and noisy pixels, especially in the presence of high-intensity noise levels. After applying predefined thresholds to the grayscale image and thus obtaining a binary image, several morphological operations are used to remove the unwanted edges and noisy pixels, perform edge thinning to ultimately provide the desired edge connectivity which results in an optimal edge detection method. The obtained results are compared to other existing state-of-the-art denoising filters and other edge detection methods in support of our assertion that the proposed method is resilient to impulse noise even under high-intensity levels.**

***Index Terms*— Impulse noise, denoising, median and mean filtering, edge detection.**

I. INTRODUCTION

Edge detection is a challenging nontrivial problem but is a task that remains essential for object identification, image segmentation, feature extraction, pattern recognition among other essential image processing tasks. There are several methods and well-known operators that are commonly used to detect edges in images, and their success is often weighted as a function of the amount of image detail that was preserved and the application at hand. When we deal with images, pertinent details can be useful when analyzing specific imaging data, but the concern has always been in delineating what really constitute actual edge data with a high degree of similarity to the original noise-free image in contrast to other background and noise data that burdens the edge detection process. The challenge is further amplified when the images are degraded by noise, affecting significantly their structural metrics.

Canny edge detection [1], perhaps one of the most useful and well-known method, is a multifaceted process that integrates Gaussian filtering for smoothing the image, intensity gradient, non-maximum suppression for edge thinning, thresholding and tracking of the edges to ensure edge connectivity and continuity. The holistically nested edge detection (HED) method [2], which is a robust edge detection method, uses convolutional neural networks and is based on image training and prediction through multi-scale and multi-level feature learning. Such edge detection methods and related edge operators extract quite successfully edge information and yield a good performance when dealing with clean images; however, their performance is degraded in the presence of impulse noise, especially when it is of high-intensity type. Such degradation could be overcome, but only with additional well thought out filtering steps. Neuro –fuzzy operator [3] is designed to detect edges in the presence of impulse noise, but its success is limited only for low intensity noise levels. A fast algorithm that detects edges in noisy images is proposed in [4], but preserving image details under different noise intensities was not its main focus.

Noise remains a ubiquitous and unwanted phenomenon that is inherent to many image acquisition and transmission sources. One such type of noise that degrades image quality is impulse (or salt and pepper) noise which appears as white and black pixels in the degraded image. In order to remove this type of noise, smoothing filters are often applied to the image to decrease the variance of the noise, while endeavoring to preserve as much as possible important details in the image. A standard course of action is to perform smoothing of the image first before some form of gradient is applied. With the knowledge that derivatives tend to amplify the presence of noise, a tradeoff needs to be negotiated between the objective of decreasing noise variance and the need for keeping all relevant image details.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

There are several image denoising and edge-preserving methods that have been proposed in the past, with in-depth surveys on them provided in [5] and [6]. The filters that are commonly used include the standard median filter [7], total variation (TV) filter [8, 9], anisotropic diffusion filter [10, 11], bilateral filter [12], guided filter [13] and non-local mean filter [14]. Empirical evaluations reveal that the median filter performs best in the presence of impulse noise. There are some studies [15, 16] that were proposed for improving the performance of the median filter in high-intensity impulse noise. A comprehensive survey on switching median filters [17] provides a comparative assessment of denoising filters such as the standard median filter [7], the center weighted median filter (CWMF) [18], the weighted median filter [19], the adaptive switching median filter (ASMF) [20] and the modified decision based unsymmetrical trimmed median filter (MDBUTMF) [21]. The results reported indicate that MDBUTMF [21] is better among them. Also, there are other switching based filters that were introduced in [22-25].

State-of-the-art filters such as truncated mean filters presented in [26-29] do indeed attenuate the presence of noise. The iterative trimmed and truncated mean algorithm (ITTM) [29] is seen as an improvement on the iterative truncated mean algorithm (ITM) and offers an estimation of the median by increasing the number of iterations without resorting to the time-consuming data sorting process. Also, ITTM has some properties such as fast convergence, symmetric distribution, unbiased estimates, convergence to median, impulsive noise suppression and edge preservation. It has a very good performance in the presence of Gaussian and α-stable noise, but its performance is highly degraded in the presence of high impulse noise.

In this study, a comparative assessment is provided contrasting the results obtained using the proposed approach with the results of the most recent and proven effective filters, which focus on the removal of impulse noise in images. These include: 1) improvement boundary discriminative noise detection (IBDND) [25] which is an improvement on BDND [24]. 2) Decision based unsymmetrical trimmed modified winsorized mean filter (DBUTMWMF) [30], which is based on two mean filtering steps. The authors report that the results they obtained were better than those achieved using AMF [31], progressive switching median filter (PSMF) [32], decision based median filter (DBMF) [33, 34], improved decision based filter (IDBA) [35], MDBUTMF [21], trimmed-global mean [36], adaptive cardinal B spline algorithm (ACBSA) [37] and the cascaded decision based median filter and unsymmetrical trimmed decision midpoint filter (CUDBMPF) [38]. 3) Two cascading algorithms were proposed in [39] with the first combining a decision based median filter and modified decision based partial trimmed global mean filter (DBPTMGF) [40] and the second combining DBMF and MDBUTMF [21]. The authors who proposed these cascading algorithms report a better performance than when using the AMF [31], the decision base unsymmetric median filter (DBUTMF) [34], the decision based partial trimmed global mean filter (DBPTGMF) [40] as well as when using other cascading algorithms such as DMF+UTMF and DMF+UTMP [33]. 4) Unbiased-based weighted mean filter (UWMF) as described in [41], which is a weighted mean filter, is based on the spatial bias, Minkowski distance and spatial distances in the x and y directions. The results using the UWMF show a better performance than when using AMF [31], the MDBUTMF [21], the improved boundary discriminative noise detection filter (IBDND) [25], cloud model filter (CMF) [42] as well as the interpolation-based impulse noise filter (IBINRF) [43]. Nonetheless, these denoising methods still encounter some challenges when faced with high impulse noise that include loss of image details, blurring of the image and unsmoothed edges, which make the edge detection process more difficult to attain.

The Motivation in this endeavor is driven by the following two goals: 1) Resolve the challenges faced with the use of denoising methods by keeping as much image details as possible, avoid blurring of the image, and preserve the sharper edges associated with boundaries. 2) Contend with these challenges even in the presence of high-intensity impulse noise. Combining these two steps highlights the novelty of this proposed method. Consequently, this study introduces a new denoising filter capable of preserving more edge details with high structural similarity to the original (noise-free) image even in the presence of high impulse noise. The results obtained, as will be discussed later, are contrasted to all other well-known denoising filters.

The proposed method, as illustrated in Figure 1, consist of four essential steps. First, adaptive median and fixed weighted mean smoothing filters in combination are applied to yield the highest structural metrics in comparison to current state-of-the-art filters. Second, edge detection using standard kernels is performed. Third, edge routes are detected based on the non-maximum suppression method. Fourth, discontinuities are filled and noisy pixels are removed according to the maximum-sequence method, especially in high predefined thresholds and high-intensity noise levels. Finally, after applying predefined thresholds and through the use of specific morphologic operations, the results are evaluated under different impulse noise intensities.

II. Proposed Method

With the proposed method, boundary edges of filtered images are assumed to have high correlation with the original images; as such edges should track the true routes even under high-intensity impulse noise. Most of the current leading filters ensure a good performance on impulse noise reduction, but they still do not perform well on boundaries, especially in the presence of impulse noise with high-intensity levels.

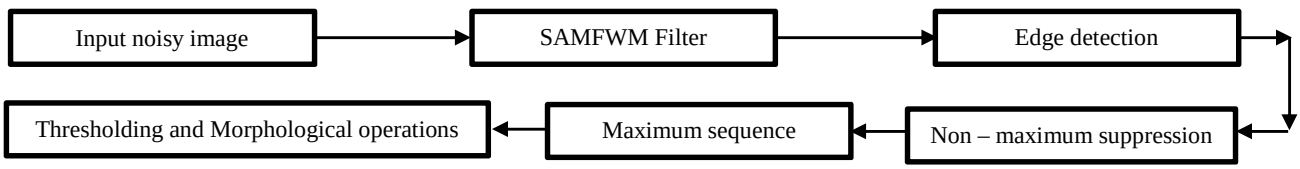


Fig. 1. Block diagram of the proposed method

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

We assume the noise model as given in (1), assuming normalization:

$$I_c = \begin{cases} 0 & Probability\ P_p \\ 1 & Probability\ P_s \\ 0 < c < 1 & Probability\ 1 - P_p - P_s \end{cases} \quad (1)$$

In this model, as expressed in [44], c denotes the uncorrupted pixels, and where the corrupted pixels are assigned probability P_s for salt and P_p for pepper. In this normalized representation of the image, 0 being the minimum intensity denoted by I_{min} and 1 being the maximum intensity denoted by I_{max}.

There are two choices that can be made when using the switching adaptive median filter:

- **In switch 1**, within an initial sliding window, all pixels with 0 and 1 values are removed, and the median value of the remaining pixels with probability of $1 - P_p - P_s$ as in (1) will replace the pixel being processed. If all of them are 0s, 1s or a combination of them, or if the variance of the pixels is much higher than the median value, then the size of the window is increased by 1 and the process is repeated until the window size reaches the predefined maximum window size. We assume the difference between pixel values is high when the variance is much higher than the median value (in this paper $\sigma > 2Median$), which could be an indication that an edge is present in that area. It thus checks the variance in bigger window sizes to validate whether such an edge does indeed exist or not. If there is an edge, the assumption is that the median value can detect it, otherwise the median value will be correlated to the texture found within the window.

- **In switch 2**, the 0s and 1s are not removed within the initial sliding window, and the median value of all pixels will replace the pixel being processed. In this switch, if all of the pixels are 0s, 1s or a combinations of 0s and 1s, the window is simply increased by 1, and in the same way as in switch 1, the process continues until the window size reaches the predefined maximum window size. This case is designed specifically for images which contain significant black and white regions with clear edges like checkerboards or mesh like images; but, for other types of images, switch 1 is expected to yield better structural metrics than switch 2.

By increasing the size of the adaptive median filter, the structural metrics will be somewhat decreased, resulting in an image that is slightly blurred. However, the edges still appear sharp. Therefore, it appears that there is a tradeoff to be made between the edges extracted and the quantitative values of the structural metrics. However, the pixel being processed will remain unchanged if all of the pixels in the selected window are 0s or 1s or a combination of them. There are special cases when a given texture would itself consist of 1s and 0s, for example a checkerboard. The problem for this latter case becomes more challenging in delineating such textures especially in the presence of impulse noise. When such combinations of 0s and 1s are found in several instances in the sliding window, the mean filter can be applied. This combination can smooth the image while maintaining high structural metrics and sharp edge boundaries even in the presence of high-intensity impulse noise. In order to avoid any lingering noise effects in the black and white regions (especially in relatively bigger regions) in which the mean filter could potentially change the intensities, an additional shrinkage window can be defined before applying the mean filter. This step, which removes 0s in white regions and 1s in black regions, can be very useful for textures that consist of a combination of black and white sections. The maximum size of the window would hence depend on the texture and noise level in the image being denoised.

Also, the structural metrics of fixed mean filter can be improved by assigning appropriate adaptive weights for the pixels in the selected window in accordance to the probabilities of noise occurrence. This window could contain all 0s (P_p), all 1s (P_s), or a combination of them together with the other pixels with probability of $1 - P_p - P_s$, as indicated earlier in (1). The mean filter tends to introduce more blur in the image, which in turn could lead to loss of details. To prevent theses side effects, the size of the mean filter should be kept small and fixed, as is done in the proposed method.

In this study, noise reduction is performed with the intent to preserve edge points in an optimal fashion.

A. Structure of the Method

The procedural steps of this method embed the two main components of switching adaptive median (SAM) filtering and fixed weighted mean (FWM) filtering with additional shrinkage window to make up the proposed denoising method we refer to as SAMFWMF. The adaptive median component is so called in that the window size can be dynamically changed according to table I in section V. Increasing the window size from its original 3x3 size is warranted only if the SAM step did not yield optimal results. Another adaptive additional window is set to overcome any the remaining noise in white and black regions. By doing so, we avoided blurring the final SAMFWMF image by increasing the size of the window in the SAM component (steps 1-5) rather than in the FWM component (steps 6 -11) of the following process:

1. In the case of switch 1, if all of the pixels in the 3×3 window are 0s and 1s, or a combination of them, or if the variance (σ) of the window is much higher than the median value (in this case $\sigma > 2Median$) then, the size of the window is increased to a 4×4, then 5×5 and so on until it reaches the predefined maximum size. Otherwise, it leaves the window size unchanged. In the case of switch 2, it only checks if all of the pixels in the 3×3 window are either 0s or 1s, or a combination of them, and if so, the size of the window is increased by 1; else, it leaves the window size unchanged. Then set the normalized pixels of the 2-D selected window as a $1 \times N$ 1-D vector (N=3 to maximum value), and check if the pixel $I(i,j)$ being processed is a corrupted pixel; that is to check if $I(i,j) = 0$ or 1 (normalized value) in $W_{1\times N} = (\dots, I(i,j), \dots)$.
2. Detect all pixels with 0 and 1 values, and in the case of switch 1 eliminate them, so the size of the window $W_{1\times N}$ is now decreased to a new size $W_{1\times N-k} = (\dots, I(i,j), \dots)$, where k represents the number of corrupted pixels that were removed; and in the case of switch 2 where such pixels are not eliminated, the size of the window remains $W_{1\times N}$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

3. Switch 1 replaces the $I(i,j)$ pixel value with the median value of the remaining $N-k$ pixels in the vector window if at least one pixel remains in the reduced window, otherwise leaves $I(i,j)$ unchanged. Switch 2 replaces the $I(i,j)$ pixel value with the median value of the N pixels in the vector window. Figure 2 shows the algorithm of the switching adaptive median filter.

for $I_{i,j} \in$ Image **do**
$W_{min} \leftarrow$ Minimum window size
$W_{max} \leftarrow$ Maximum window size

Switch 1:
for $I_{x,y} \in W, \quad W_{min} \leq W \leq W_{max}$ **do**
$S \leftarrow$ remove all 0's and 1`s
$S' \leftarrow$ sort (W)
$M \leftarrow$ median (S')
$L \leftarrow$ length (S')
$v \leftarrow$ variance (S)
if $L \neq 0 \ \wedge \ v \leq 2 * M$ **then**
break;
end if
end for

Switch 2:
for $I_{x,y} \in W, \quad W_{min} \leq W \leq W_{max}$ **do**
$S \leftarrow$ sort (W)
$M \leftarrow$ median (S)
$L \leftarrow$ length (S)
if $L \neq 0$ **then**
break;
end if
end for

if $I_{i,j} \neq 1 \ \wedge \ I_{i,j} \neq 0$ **then**
$I_{i,j} \leftarrow I_{i,j}$, **continue**
else
$I_{i,j} \leftarrow M$
end if
end for

Fig. 2. Algorithm of the switching adaptive median filter

4. Leave uncorrupted pixels unchanged.
5. Slide the window by one pixel and repeat the process consisting of steps 1-4 throughout the entire image, establishing at this stage the SAM filtered image.
6. Starting from the predefined maximum size for the shrinkage window, we start by checking the boundary pixels of the selected window (filtering window). One of the following condition has to be met: If they are all 1, the interior pixels are changed to 1. If all the pixels on the boundary are 0, then the interior pixels are changed to 0. Otherwise (there is a combination of 0 and 1), the window is then shrunk by one and the process is repeated until the minimum size (3×3) is reached. Figure 3 shows the algorithm of the shrinkage window.
7. For the fixed mean filtered image, use a 2×2 window in a convolution manner, and check if the pixel being processed $(I(i,j)$ within the vector window $W_{1\times4} = (I(i,j),\ I(i,j+1),\ I(i+1,j), I(i+1,j+1))$ is found corrupted (i.e., $I(i,j) = 0$ or 1 (normalized value))
8. Using the weights selected on the basis of the two conditions described next, if salt or pepper (probability P_s or P_p) is detected, the new processed pixel would be

$L_{max} \leftarrow$ predefined maximum window size
for $I_{x,y} \in$ Image **do**
Northvector $\leftarrow I_{x,y} \in$ north boundary of $W_{L_{max}}$
Westvector $\leftarrow I_{x,y} \in$ west boundary of $W_{L_{max}}$
Eastvector $\leftarrow I_{x,y} \in$ east boundary of $W_{L_{max}}$
Southvector $\leftarrow I_{x,y} \in$ south boundary of $W_{L_{max}}$
NV1 $\leftarrow$ Northvector (all pixels) =1
WV1 $\leftarrow$ Westvector (all pixels) =1
EV1 $\leftarrow$ Eastvector (all pixels) =1
SV1 $\leftarrow$ Southvector (all pixels) =1
NV0 $\leftarrow$ Northvector (all pixels) =0
WV0 $\leftarrow$ Westvector (all pixels) =0
EV0 $\leftarrow$ Eastvector (all pixels) =0
SV0 $\leftarrow$ Southvector (all pixels) =0
LNV1 $\leftarrow$ length (NV1)
LWV1 $\leftarrow$ length (WV1)
LEV1 $\leftarrow$ length (EV1)
LSV1 $\leftarrow$ length (SV1)
LNV0 $\leftarrow$ length (NV0)
LWV0 $\leftarrow$ length (WV0)
LEV0 $\leftarrow$ length (EV0)
LSV0 $\leftarrow$ length (SV0)
if LNV1 = $L_{max} \vee$ LWV1 = $L_{max} \vee$ LEV1 = $L_{max} \vee$ LSV1=L_{max} **then**
$I_{x,y} \in W'_{L_{max}}$ (window without boundary) = 1
else if LNV0 = $L_{max} \vee$ LWV0 = $L_{max} \vee$ LEV0 = $L_{max} \vee$ LSV0=L_{max} **then**
$I_{x,y} \in W'_{L_{max}}$ (window without boundary) = 0
else
if L_{max} >=3
$L_{max} = L_{max} - 1$
end if
end if
end for

Fig. 3. Algorithm of the shrinkage window

assigned the new value as in (2). Otherwise, it leaves the pixels unchanged.

$$M_{new}(i,j) = \frac{\sum_{(x,y)\in S_{new(i,j)}} \omega_{x,y}\, I_{x,y}}{N-1} \qquad (2)$$

In this equations, N is 4, $S_{new(i,j)} = \{I(i,j+1), I(i+1,j), I(i+1,j+1)\}$, with indices (i,j) indicating the positions of the corrupted pixels, and (x,y) are the coordinates of the pixels around it. In this proposed method, when the detected corrupted pixel occurs as salt or pepper (with probabilities P_s or P_p), the weights $\omega_{x,y}$ are directly selected based on the probability of occurrence 1 or 0 for neighboring pixels, according to one of these conditions:

- Condition1: We assume the corrupted pixel with the probability of P_s or P_p occurs, and the probability of occurrence of 1 is more than that of 0 for the neighboring pixels (with the assumption that the window contains only 0 and 1). We will set $\omega_{x,y} = 1$ for all pixels. In this case, if all the neighboring pixels are equal to 1, the value of the corrupted pixel changes to 1, otherwise, changes to a value between 0 and 1 based on the assumption that the probability of changing neighboring pixels to a value between 0 and 1 is high.
- Condition 2: We assume the corrupted pixel with the probability of P_s or P_p occur, and the probability of occurrence of 0 is more than that of 1 in neighboring pixels (with the assumption that the window contains only 0 and 1). Then we will set $\omega_{x,y} = 2$ for the east and south pixels and $\omega_{x,y} = 1$ for the southeast pixel. In this case, if all the neighboring pixels are equal to 0, the value of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

the corrupted pixel changes to 0, otherwise, changes to a value between 0 and 1 based on the assumption that the probability of changing neighboring pixels to a value between 0 and 1 is high.

- Condition 3: We assume the corrupted pixel with the probability of P_s or P_p occurs, and there is a probability for neighboring pixels with value between 0 and 1 to exist. Then we will set $\omega_{x,y} = 2$ for the east and south pixels and $\omega_{x,y} = 1$ for the southeast pixel. If all of the neighboring pixels are equal or if the summation of the weighted neighboring pixels are greater or equal to the denominator (greater or equal to N-1) as $sum = \sum_{(x,y)\in S_{i,j}} \omega_{x,y}\ I_{x,y}\ \ if\ \ sum \geq N-1$, then we will set $\omega_{x,y} = 1$ for all pixels. In this case, the value of the corrupted pixel changes to a value between 0 and 1 with the assumption that the probability of changing neighboring 0 or 1 pixels (if they exist) to a value between 0 and 1 is high and with the rest of the pixels still assuming values between 0 and 1. Also, in the case of equal neighboring pixel values, the corrupted pixel would be equal to the value of these neighboring pixels.

Figure 4 shows the algorithm of the fixed weighted mean filter.

```
for  I_{i,j} ∈ Image do
w_{2×2} ← 2 × 2 window size
Mean = (∑_{(x,y)∈S_{i,j}} I_{x,y}) / 3
WeightedMean = (∑_{(x,y)∈S_{i,j}} ω_{x,y} I_{x,y}) / 3
Sum = ∑_{(x,y)∈S_{i,j}} ω_{x,y} I_{x,y}
S ← remove all 0's and 1`s
LS ← length (S)
   if I_{i,j} = 1 ∧ I_{i,j} = 0 then
      if LS=0 then
         if I_{x,y} = 1 then
            count1 = count1+1;
         end if
         if I_{x,y} = 0 then
            count0 = count0 + 1;
         end if
         if count1 > count0 then
            I_{i,j} ← Mean
         else
            I_{i,j} ← WeightedMean
         end if
      else
         if Sum >= 3 then
            I_{i,j} ← Mean
         else
            I_{i,j} ← WeightedMean
         end if
      end if
   else
      I_{i,j} ← I_{i,j}, continue
   end if
end for
```

Fig. 4. Algorithm of the fixed weighted mean filter

9. Replace the corrupted pixel with the mean value.
10. Leave uncorrupted pixels unchanged.
11. Repeat steps 7-9 for the entire filtered image, resulting in the SAMFWM filtered image.
12. Check the level of impulse noise present, and determine if the filter yields satisfactory results. If results are not satisfactory, increase the switching adaptive median 3×3 window into 5×5 and so on. Hence, as the intensity of the noise present is higher, set a new of dimension of the window as $W_{new} = W_{old} + 2$, and the process consisting of steps 1-11 is repeated until optimal results are obtained.

In this last step, optimization of the filtering results is reached when the evaluations measures, as described next, yield the highest values.

B. Evaluation measures

To measure the degree of edge preserving and image structural metrics, standard measures are computed in order to compare the performance of different filters including the proposed method to gauge the quality of image after the smoothing process is performed.

The following measures are used in this study:

- Correlation coefficient (β) [45] measures the amount of preserved details and edges after the denoising process.
- Structural similarity index (SSIM) [46] measures the difference between the original noise-free image and the denoised image after the denoising process.
- Peak signal to noise ratio (PSNR) which measures the level of noise in the denoised image after the denoising process.
- Figure of merit (FOM) [47] which measures the edge detection performance.

Equations (3) through (6) provide the different formulations used for Correlation (β), SSIM, PSNR and FOM, respectively. In all these formulations, $x(i,j)$ represent the pixels in the original noise-free image, $n(i,j)$ represent the pixels in the noisy image, and $y(i,j)$ represent the pixels in the denoised image after the filtering process has been applied.

The correlation coefficient is defined as follows:

$$\beta = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[x(i,j)-\overline{x(i,j)}]\times[y(i,j)-\overline{y(i,j)}]}{\sqrt{\sum_{I=0}^{M-1}\sum_{j=0}^{N-1}[x(i,j)-\overline{x(i,j)}]^2\times[y(i,j)-\overline{y(i,j)}]^2}} \quad (3)$$

Where $\overline{x(i,j)}$ and $\overline{y(i,j)}$ represent the mean values of the x and y images, respectively.

The structural similarity index (SSIM) is measured as follows:

$$\text{SSIM} = \frac{(2\bar{x}\bar{y}+C1)(2\sigma_{xy}+C2)}{(\bar{x}^2+\bar{y}^2+C1)(\sigma_x{}^2+\sigma_y{}^2+C2)} \quad (4)$$

Where σ_x and σ_y define the standard deviations in the x and y images, respectively, and σ_{xy} defines the standard deviation between the two images, while C_1 and C_2 are two variables which depend on the dynamic range of pixels often set in the as C_1=0.01L and C_2=0.03L, where L is the dynamic range (here it is assumed 1 since pixels are normalized). The values

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

of 0.01 and 0.03 are default values recommended by the inventors of the SSIM measure to stabilize the denominator and avoid a zero value in the denominator.

The peak signal to noise ratio (PSNR) measure is given as:

$$PSNR = 10log\frac{(\max(x))^2}{MSE} \quad (5)$$

Where MSE is the mean square error, and $\max(x)$ defines the maximum intensity of the pixels in image x.

Noteworthy comparisons and evaluations of different edge detection methods are provided in [47, 48]. It should be noted that the main point in this proposed method is in edge detection evaluation after the denoising process has been accomplished; therefore, the figure of merit (FOM) [47] of the algorithm is measured to assess the merits of the denoising process. FOM, which consists of several steps, is a metric that measures the distance between the detected edges with those of a reference image. The binary reference image is generated based on 3 steps: 1) white noise generator, 2) low pass filter (Gaussian PSF with width σ_g), and 3) a zero-crossing detector. The test image is generated based on 8 steps: 1) white noise generator, 2) low-pass filter (Gaussian PSF with width σ_g), 3) thresholding, 4) region labeling, 5) random grey level assignment (with standard deviation σ_h), 6) low-pass filter (Gaussian PSF with width σ_p), 7) Gaussian noise with variance σ_n^2), and 8) impulse noise (with noise level Im); all this before applying the denoising process which is an extra step in order to evaluate the edge detection performance in the presence of impulse noise. The FOM measure is thus given as:

$$FOM(\sigma_c, \sigma_g, \sigma_h, \sigma_p, \sigma_n, Im) = \frac{1}{NM}\sum_{n=1}^{N}\sum_{m=1}^{M} g^2(n,m) \quad (6)$$

Where (N, M) is the size of the image, $g(n,m)$ is the convolution between $f(n,m)$ and the Gaussian PSF with width σ_c, and where $f(n,m)$ is the difference between the binary image (with the detected edges) and the binary reference image.

C. *Experimental Evaluations in the Presence of Impulse Noise*

The cascading algorithm [39], IBDNDF [25], DBUTMWMF [30], UWMF [41], considered as most effective when dealing with impulse noise, are compared to the SAMFWMF under different impulse noise intensity levels, and the aforementioned metrics are used for evaluation. It should be noted that in order to optimize the denoising of the image, when the impulse noise is increased, the size of the filter may be changed.

Using images of Lena (512×512), Camera man (256×256), Coins (300×246) and checkerboard (256×256) as the standard examples used in the literature for comparative purposes, Figure 5 shows the correlation comparison between both switching methods on different types of images. The initial windows for both of them are equal to the maximum window size in the related noise intensity.

Figure 6 shows the edge boundaries of different images after applying the SAMFWMF with switch 1 in the presence of high noise intensities with different initial adaptive median windows and in contrast to the other well-known filters. The

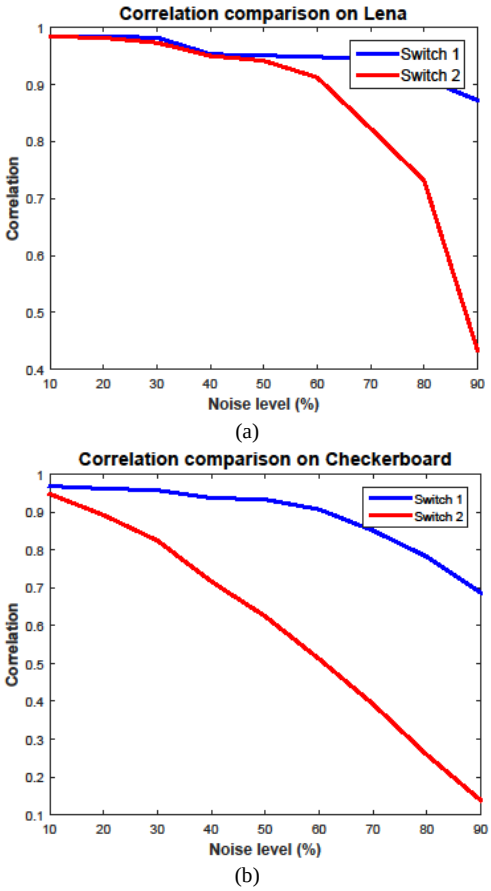


Fig. 5. Correlation comparison between switches a) on Lena b) on

adaptive properties of the proposed median filter, in which its initialization is dynamic, adapts well to the smoothness of the edges. In the lower initial window size, sharpness of the edges is not easily attained; therefore, as the initial window size is increased, the edges appear smoother and sharper, Figure 6 (rows 7 and 8) exemplifies these observations. As the results for the SAMFWMF reveal, the intensity variations on the edges are sharper, and the structural similarity measures are higher than with other filters even when the impulse noise intensity is high. Furthermore, since the mean filter introduces blur in the image with some details lost as a consequence, the proposed method maintains the size of the mean filter fixed but with specific weights given to the neighboring pixels during the smoothing process.

As indicated earlier, ITTM filters alone do not perform as well in the presence of high impulse noise intensity. They can instead be used as a post processing step to attenuate the effect of the high impulse noise, therefore, one can replace the proposed fixed weighted mean filter with ITTM filter (i.e., a combination of the proposed adaptive median filter with the ITTM filter) and then evaluate the results. Figure 7 shows the correlation comparison between the two filters: the proposed SAMFWMF which has fixed weighted mean filter, and a combination of the proposed adaptive median filter with the ITTM filter. The initial adaptive median window size for both of them are equal to the minimum window size (3 × 3). As the figure shows, SAMFWMF has performed better under the different impulse noise intensities. Also, in section II-A under conditions 1 and 2 of step 8, always there are 0`s and 1`s in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

proposed fixed weighted mean filter window, which meets the defined conditions in ITTM filters and makes it possible to measure the regular mean value of the selected window for the output. But, if ITTM filters are used under condition 3, no change appears in the results and it only adds more complexity to our proposed filter.

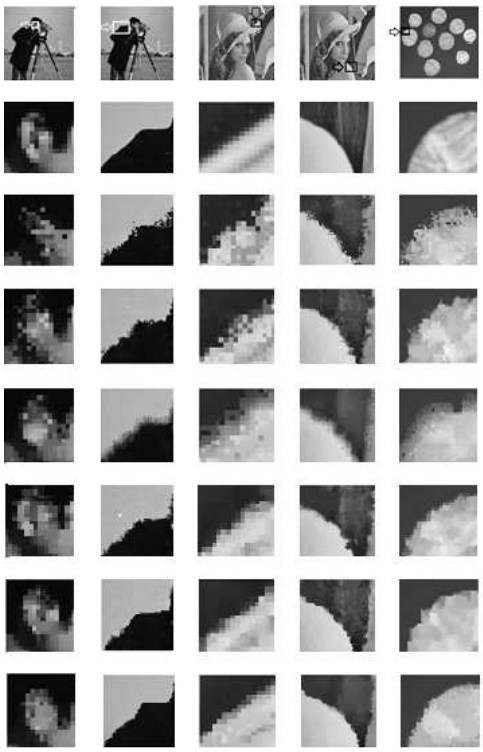


Fig. 6. Results of filtering Camera man with 70% impulse noise intensity, Camera man with 90% impulse noise intensity, Lena with 80% impulse noise intensity, Lena with 90% impulse noise intensity and Coins with 90% impulse noise intensity in column 1 through 5, respectively. After applying the filter: Rows 1 through 8 are: Original image showing the specified area under scrutiny, original specified area, denoised results using, cascading algorithm [39], IBDNDF [25], DBUTMWMF [30] and UWMF [41], and SAMFWMF with initial adaptive median window size=3 (minimum size), and SAMFWMF with initial adaptive median window size=maximum size in that noise level

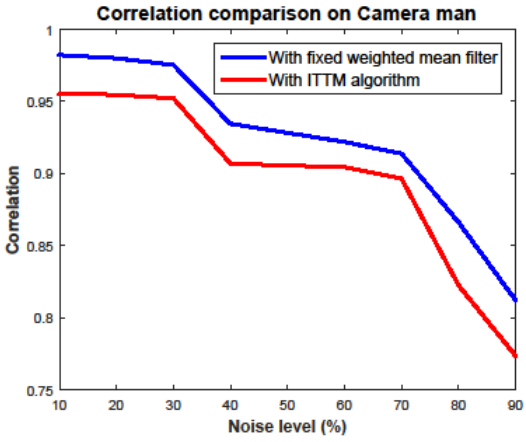


Fig. 7. Correlation comparison SAMFWMF and a combination of proposed adaptive median filter with ITTM filter on Camera man

Figure 8 shows the correlation comparison between the two states (with and without weights) of fixed mean filter (FM) with switch 1 in the presence of different noise intensities. The

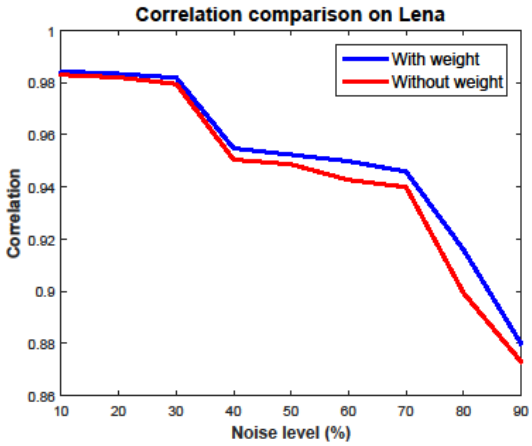


Fig. 8. Correlation comparison between two states of fixed mean filter on Lena

initial adaptive median window size for both of them are equal to the minimum window size (3×3). In the first approach, weights ($\omega_{x,y}$) are set for neighboring pixels, and in the second approach the method is run without setting these neighboring weights. As the figure shows, setting weights improves the structural metrics in the denoised images. Also, the proposed fixed mean filter can be used as post-processing for other denoising filters. Figure 9 shows the comparison

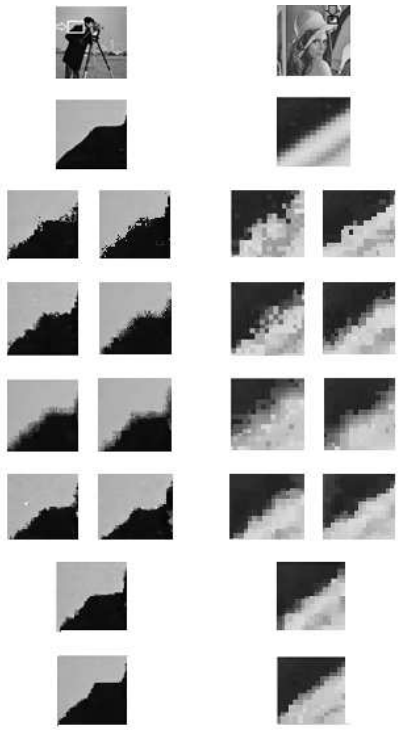


Fig. 9. Results of filtering Camera man with 90% impulse noise intensity, Lena with 80% impulse noise intensity in columns a and b, respectively. After applying the filter: Rows 1 through 8 are: Original image showing the specified area under scrutiny, original specified area, denoised results using, cascading algorithm [39] without (left) and with (right) fixed weighted mean filter, IBDNDF [25] without (left) and with (right) fixed weighted mean filter, DBUTMWMF [30] without (left) and with (right) fixed weighted mean filter and UWMF [41] without (left) and with (right) fixed weighted mean filter, SAMFWMF with initial adaptive median window size=3 (minimum size), and SAMFWMF with initial adaptive median window size=maximum size in that noise level

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

between other denoising methods with and without fixed weighted filter as a post-processing step in the presence of high impulse noise intensities.

D. Edge Gradients

Edge detection is a nontrivial process mainly due to the ambiguity associated with defining what constitutes an observable transition (differential thresholds or just-noticeable difference) between image intensities. For all practical purposes, first derivative operators are adequate in their use for edge detection and in determining local minima and maxima. Second derivative operators could be useful for localization purposes due to the zero crossings. To determine the strength of an edge point, the gradient should be measured perpendicular to the edge direction.

There are several kernels that can be used for edge detection. First derivative operators, although weak in terms of localization, are nonetheless less sensitive to noise than their second derivative counterparts and are also less complicated in their implementation. Accordingly, for this study, any edge detection kernel could have been used, but in the implementation of the proposed method, a 3×3 first order derivative kernel is used, and the results are satisfactory in terms of the evaluation metrics used in this study. Figure 11 (top part) describes the algorithm of the edge detection method.

III. CONTINUITY IN EDGES AND THRESHOLDING IN GRAY SCALE IMAGES

A. Non-maximum suppression

This technique [1] is used for edge thinning in the grayscale image. Edge strength is compared with the neighboring pixels according to gradient direction, the whole process can be summarized as follows:

- Calculate the vertical and horizontal gradient.
- Calculate the angle of the gradient, and
  - if the angle of gradient is 0 degree, the gradient magnitude is checked in the east and west directions, and if it is more than the magnitude of pixels in these directions, it is considered on the edge
  - if the angle of gradient is 45 degrees, the gradient magnitude is checked in the northeast and southwest directions, and if it is more than magnitude of pixels in these directions, it is considered on the edge
  - if the angle of gradient is 90 degrees, the gradient magnitude is checked in the north and south directions and if it is more than the magnitude of pixels in these directions, it is considered on the edge
  - if the angle of gradient is 135 degrees, the gradient magnitude is checked in the northwest and southeast directions and if it is more than the magnitude of pixels in these directions, it is considered on the edge

Figure 10 (bottom part) shows the algorithm of the non-maximum suppression.

$k_x \leftarrow$ x kernel
$k_y \leftarrow$ y kernel
for $I_{i,j} \in$ Image **do**
  $I_X \leftarrow$ Image conv. with k_x
  $I_y \leftarrow$ Image conv. with k_y
  $I_{i,j}$ = sqrt ($I_X^2 + I_y^2$)
  Theta = arctangent (I_y / I_x)
  Angle of direction = minimum of |[0;45;90;135;180] - $\text{Theta}_{i,j}$ |
  if Angle of direction = 0 ∧ Angle of direction = 180 **then**
    if $I_{i,j}$ is minimum in west and east direction **then**
      $I_{i,j} = 0$
    end if
  end if
  if Angle of direction = 45 **then**
    if $I_{i,j}$ is minimum in northeast and southwest direction **then**
      $I_{i,j} = 0$
    end if
  end if
  if Angle of direction = 135 **then**
    if $I_{i,j}$ is minimum in northwest and southeast direction **then**
      $I_{i,j} = 0$
    end if
  end if
  if Angle of direction = 135 **then**
    if $I_{i,j}$ is minimum in north and south direction **then**
      $I_{i,j} = 0$
    end if
  end if
end for

Fig. 10. Algorithm of the edge detection and non-maximum suppression

B. Maximum-sequence and thresholding

This technique is used to maintain edge continuity during the edge detection process and extract more edges at different threshold intensities while minimizing noise. The predefined threshold (T) and edge point factor (α) are set to any value within the normalized range such that 0 < T < 1, 0 < α < 1. The value α when used with T as ($\alpha * \mathrm{T}$) is assigned such as to resolve the dilemma of selecting too high or too low of a threshold initially. Then the following steps are considered:

- The process starts by setting a threshold value for a starting (first) edge point ($I_{startpoint}$) as ($\alpha * \mathrm{T}$) or (T)
- The next step is to check the value of $I_{startpoint}$ in all four edge directions in a 2×2 window $(W_{2\times2} = I(i,j), I(i,j+1), I(i+1,j), I(i+1,j+1))$, if it is higher than (T) or ($\alpha * \mathrm{T}$).
- Find the maximum value of the neighboring pixels (I_{max}) in direction of the edge.
- Increase the intensity of the maximum pixel (I_{max}) which was found within the 2×2 window to the value of (T).
- If $I_{startpoint} \geq \alpha * \mathrm{T}$ and $I_{max} \geq \alpha * \mathrm{T}$, then $I_{maxnew} = T$; or if $I_{startpoint} \geq \mathrm{T}$ and $I_{max} \geq \alpha * \mathrm{T}$ then $I_{maxnew} = T$
- This process scans the entire image.

It is possible to set $\alpha = 1$ and change the value of T to get the desired results, but this does not guarantee a noise-free outcome when a low value of T is chosen, especially when the probability of occurrence of the salt and pepper is high. By using the weights for the mean filter in order to affect the values of salt and pepper, the values of the noisy pixels may be changed to a value more consistent with their neighboring pixels, and if the value of T is less than that, the algorithm will assume the noisy pixels to be edge points. However, by using the maximum sequence method, this problem can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

overcome. Such an algorithm can thus detect edges at any threshold level simply by changing the value of α. If the pixel is considered part of the edge line, the algorithm will continue to track the line, but if it is a noisy point, the algorithm makes the pixel zero, creating a discontinuity.

There are two choices that can be made: in the first choice, we can assume $I_{startpoint} \geq$ T, while in the second choice, we assume $I_{startpoint} \geq \alpha *$ T. For both choices, the algorithm tracks the edge line and maintains the continuity. In the second choice, maybe some points in the edge line may not be detected, because the algorithm selects only the neighboring pixels which have maximum value in the selected window. This can be resolved by increasing the value of $I_{startpoint}$ in the related direction to the value of T if at least one of the neighboring pixel is $\geq \alpha *$ T. Empirical evidence showed that is better to use the first choice for low T and the second one for high T, on the basis of the histogram of the image. Figure 11 shows the performance of the algorithm to remove noisy pixels and track the edge lines. The symbol ($\times$) denotes any neighboring pixel around the $I_{startpoint}$ which is less than the value of ($\alpha *$ T). Figure 11-a shows how the maximum sequence method removes the noisy point $n(i,j)$. Recall that the value of $n(i,j)$ should be $n(i,j) \geq T$ in the first choice and $n(i,j) \geq \alpha *$ T in the second choice. If these conditions are met, the algorithm after checking all directions, changes the value of $n(i,j)$ to zero. Figure 11-b shows how the algorithm tracks the edges when more than one edge pixel (E) is found in a given direction. Likewise, the value of E should be $E \geq T$ in the first choice, and $E \geq \alpha *$ T in the second choice. The algorithm changes the value of both of the E pixels equal to the value of T and continues tracking the edge line. Figure 11-c shows the connectivity in the first choice when there is a discontinuity (D) between two edge points (E) where $E \geq T$ and $D \geq \alpha *$ T. The algorithm in this case increases the value of D to the value of T to maintain continuity. Figure 12 shows the algorithm of the maximum sequence.

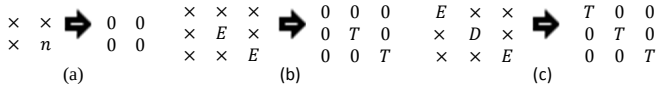


Fig. 11. Performance of the maximum sequence

Thresholding remains however a challenging problem in image processing, where a general value definition for all images is very difficult to attain, but it instead varies according to image characteristics and application at hand. There are known methods for thresholding such as the hysteresis and Otsu methods which are quite interesting and effectual. The hysteresis method relies on two thresholds (low and high) and pixels above the high threshold are assumed edges and those below are not edges and those pixels in between these two thresholds are edges only if they are adjacent to other edges. Also, the Otsu method sets its own general threshold depending on the minimized variance of the two regions that are separated by the threshold [49]. This last approach is akin to finding that threshold maximizing interclass variance in a bimodal histogram.

Figure 13 contrasts the results between different thresholding methods and the maximum-sequence method on the image. As the figure shows, the maximum-sequence

```
T ← predefined threshold
α ← edge point factor
T1 ← α.T
for I_{i,j} ∈ Image do
W_{2×2} ← 2 × 2 window size
for I_{x,y} ∈ W_{2×2} do
  if I_{nort hwest} ≥ T1 then
    h ← highest value of neighboring pixel
    if h ≥ T1 then
      h = T
      I_{nort hwest} = T
    end if
  end if
  if I_{nort heast} ≥ T1 then
    h ← highest value of neighboring pixel
    if h ≥ T1 then
      h = T
      I_{nort heast} = T
    end if
  end if
  if I_{sout hwest} ≥ T1 then
    h ← highest value of neighboring pixel
    if h ≥ T1 then
      h = T
      I_{sout hwest} = T
    end if
  end if
  if I_{sout heast} ≥ T1 then
    h ← highest value of neighboring pixel
    if h ≥ T1 then
      h = T
      I_{sout heast} = T
    end if
  end if
end for
end for
```

Fig. 12. Algorithm of the maximum sequence.

method can detect more edges, and the edges it detects are thinner in the different threshold intensities in contrast to the other methods.

IV. MORPHOLOGICAL OPERATIONS

After applying the threshold on the image, the output would be a binary image. So, in order to improve the binary image, some morphological operations as shown in figure 14 are performed on the image. The objectives for using such operations are to remove unwanted edge points and improve the tasks of edge thinning and edge continuity. They also help in determining the true edge boundaries, especially for curved regions, which remain a challenging task for many of the edge operators.

Figure 14 shows examples of binary formatted morphological operations which are applied to the binary image.

- Figure 14-a shows that diagonal pixel which is attached to the lines or curves, is removed, bit "1" in the top-left corner can be put in different corners.
- Figure 14-b shows that H pixel(s) which lie between two lines, are removed; it removes at most two pixels.
- Figure 14-c and 14-d shows that the gap between two pixels will be filled with a single pixel and connects two vertical and horizontal lines.
- Figure 14-e shows the unwanted pixels removal which are attached to each other in a region, this process is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

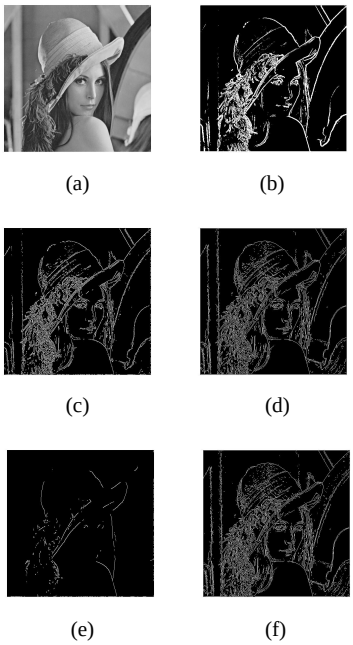


Fig. 13. Edges detection with different thresholding methods on Lena images a) Original image b) Otsu method c) Hysteresis method with predefined threshold value=0.01-0.1 (normalized) d) Proposed edge detection with predefined threshold value (T=0.1, normalized) and edge point factor (α = 0.6) e) Hysteresis method with predefined threshold value=0.1-0.4 (normalized) f) Proposed edge detection with predefined threshold value (T=0.4, normalized) and edge point factor (α = 0.15)

under control of the user, somehow, the user can determine the number of pixels that should be removed.

Some new developments were performed on the following morphological operations:

- Figure 14-f shows that vertical and horizontal pixel(s) which are attached to the lines or curves, are removed, also, it removes the pixels from left side, upside and downside.
- Figure 14-g shows how a single pixel which is attached to the lines or curves vertically or horizontally, are removed. The symbol '×' indicates that the pixel can be zero or one and the pixel can be attached left, right, up and down.
- Figure 14-h and 14-i shows that the gap between two pixels will be filled with a single pixel and connects the curves; the figures show examples of binary formats for horizontal up-right side curve connection and vertical down-left side curve connection, respectively. Also, it can connect the curve horizontally or vertically in different sides.
- Figure 14-j shows that the pixel(s) which lie in front of each other as parallel (double edges), are removed, they can be up to 3 pixels.
- Figure 14-k shows a pixel on the corner is removed to make the edge thinner. The symbol "×" indicates that the pixel can be zero or one and such a pixel can be situated in the different corners.

Figure 15 shows the algorithm of the thresholding and morphological operations.

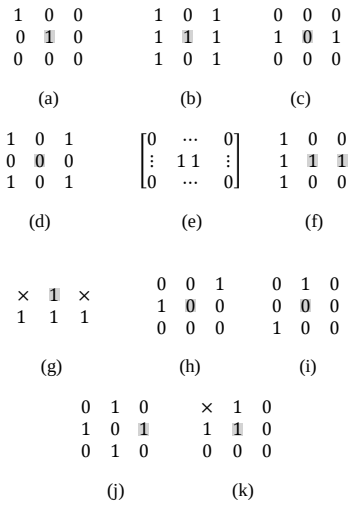


Fig. 14. Morphological operations

```
T ← predefined threshold
for I_{i,j} ∈ Image do
    if I_{i,j} ≥ T then
        I_{i,j} = 1
    end if
    if I_{i,j} ≤ T then
        I_{i,j} = 0
    end if
end for
for I_{i,j} ∈ Image do
  "Apply morphological operations from figure 8"
end for
```

Fig. 15. Algorithm of the thresholding and morphological operations.

V. Results

In this section, the results obtained using the proposed method after applying the steps of denoising (SAMFWMF) followed by edge detection is presented. The results of SAMFWMF are compared with the cascading algorithm [39], IBDNDF [25], DBUTMWMF [30] and UWMF [41] filters on different images and under different impulse noise intensity levels. Images of Lena (512×512), Camera man (256×256), Coins (300×246) and checkerboard (256×256) are standard examples used in the literature for comparative purposes.

Table II shows the results obtained on the execution time (in seconds) after the denoising process, comparing different denoising filters to the SAMFWMF. All the experiment were run on a PC with Intel(R) core (TM) 2 Quad CPU 2.67GHZ and 8G RAM. All filters except for UWMF [41] yielded high execution time in the presence of high-intensity impulse noise. As the results show, SAMFWMF has a high execution time because of the complicated nesting blocks. Most of the execution time of SAMFWMF is dedicated to switching adaptive median filter (more than 70%) and the rest of the time is dedicated to the shrinkage window and weighted fixed mean filters. By decreasing the initial adaptive median window size, the execution time is increased. Also, by decreasing the shrinkage window size, the execution time is decreased.

Tables III-IV, show respectively the results obtained on the correlation (β) and the peak signal to noise ratio (PSNR) measured in comparison to the different initial adaptive

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

median window sizes for the SAMFWMF. Note that in tables III-IV, higher numbers are associated with better results. All these metrics/measures are computed in the presence of 10 up to 90 percent impulse noise and switch 1 is used for SAMFWMF. Also, Table V shows the results for the computed structural metrics using the checkerboard as a challenging example for different initial adaptive median window sizes for the SAMFWMF. In this case, switch 2 is used for the SAMFWMF, given the nature of the checkerboard image used here as a challenge for denoising such type of input images. As Tables III-V show, by increasing the initial adaptive median window size (approaching to maximum predefined window size), the values of the structural metrics are decreased, but, the edges are sharper and smoother; therefore, there is a tradeoff between better image similarity with less noise and the need for sharper edges.

Tables VI-VIII show respectively the results obtained on the correlation (β), the peak signal to noise ratio (PSNR) and the structural similarity (SSIM) measures, comparing different denoising filters to the SAMFWMF based on the minimum initial adaptive median window size to the maximum predefined window size. All these metrics/measures are computed in the presence of 10 to 90 percent impulse noise and switch 1 is used for the proposed filter. In tables VI-VIII, higher numbers are again associated with better results. As the results show, SAMFWMF yielded better structural metrics. Also, table IX shows the results for the computed structural metrics using the checkerboard example for comparing the results obtained using different denoising filters to the SAMFWMF. In this case, switch 2 is used for the SAMFWMF, with higher numbers indicating better results. As the results show, SAMFWMF has better structural metrics. Also, table X shows the results obtained on the correlation (β) and the peak signal to noise ratio (PSNR) measures, comparing other denoising filters with and without fixed weighted mean filter as a post-processing step. All these metrics/measures are computed in the presence of 10 to 90 percent impulse noise. As the results indicate, the structural metrics are increased when the fixed weighted mean filter is used as a post-processing step for other denoising filters.

The results in table XI show the FOM comparison (with different input parameters) between the proposed edge detection (without SAMFWMF) and the Canny edge detection algorithm. Table XII shows the FOM comparison (with different input parameters) between proposed edge detection algorithm after SAMFWMF denoising process with Canny edge detection algorithm after the same SAMFWMF denoising process, and the proposed edge detection algorithm after UWMF [41] denoising process with and without fixed weighted mean filter as a post processing step. We selected UWMF [41], because visually it has shown better results among the other filters used in the comparison. In table XII, in order to evaluate the edge detection after the denoising processes, we insert an extra block in FOM process which injects impulse noise before the denoising process is applied. In tables XI and XII, lower numbers in this case show improvement on the performance as FOM is monotonically increasing the noise variance and image blurring. The size of input for FOM is 64×64. A 3×3 first-order derivative kernel is used for the proposed edge detection algorithm. In tables XI and XII, there is an unexpected tendency in the results of the proposed edge detection which shows decreasing FOM values by increasing noise intensity. The study in [47] indicates that when operators are used (like Sobel) which can cause thickening of edges or missed edges, noise can improve the detector quality by decorrelation of the quantization error. However, the proposed edge detection algorithm (with and without applying SAMFWMF) has resulted in better structural metrics.

Figures 16 and 17 show a comparison of the denoising filters in the presence of 90% impulse noise on the images of Lena and the coins. The proposed filter with switch 1 is used in both figures. As the results show, SAMFWMF has better structural metrics, and by increasing the initial adaptive median window size (approaching to maximum predefined window size), the similarity is decreased, but the edges become sharper and smoother.

To evaluate the performance of the proposed edge detection step after SAMFWMF process with switch 1, the results obtained are compared with the other denoising filters. Also, the performance of the SAMFWMF with switch 1 after applying the Canny edge detector, which is one of the most powerful and most reliable edge detectors [47] [48], is evaluated. Figures 18 and 19 show these comparisons on the camera man and coins in the presence of 90% impulse noise intensities. For both figures, the initial adaptive median window size for SAMFWMF is equal to the maximum predefined window size. In both figures, part (c) shows the results when applying the Canny edge detection step after SAMFWMF process, part (d) shows the results when applying the proposed edge detection step after the cascading algorithm [39] process, part (e) shows the results when applying the proposed edge detection step after IBDNDF [25] process, part (f) shows the results when applying the proposed edge detection step after DBUTMWMF [30] process, part (g) shows the results when applying the proposed edge detection step after the UWMF [41] process, and part (h) shows the result when applying the proposed edge detection step after the SAMFWMF process. Figure 20 shows the results when applying the SAMFWMF (using switch 2) and the proposed edge detection algorithm on the checkerboard image.

As the results clearly demonstrate, the proposed algorithm has better performance amongst all other methods in terms of keeping relevant detail and in obtaining the highest similarity, least noise, preserving edges, and better edge tracking, especially in the presence of impulse noise under high-intensity levels. Table XIII summarizes the acronyms and the corresponding methodologies.

TABLE I
MAXIMUM WINDOW SIZE OF ADAPTIVE MEDIAN FILTER IN DIFFERENT NOISE LEVELS A) ON LENA, CAMERA MAN AND COINS B) CHECKER BOARD

Window size	***3×3***	***5×5***	***7×7***	***9×9***	***> 9×9***
Noise level	$< 40\%$	$\geq 40\%$ $\leq 70\%$	$> 70\%$ $\leq 80\%$	$> 80\%$ $\leq 90\%$	> 90

(a)

Window size	***3×3***	***5×5***	***≥7×7***
Noise level	$< 40\%$	$\geq 40\%$ $< 70\%$	$\geq 70\%$

(b)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

VI. CONCLUSION

In this study, a new combination of median and mean filter, we refer to as the switching adaptive median and mean filter (SAMFWMF), with additional shrinkage window, was introduced as a new smoothing filter to remove or minimize in an optimal fashion the presence of impulse noise even at high intensity levels. The adaptive properties of the median filter can control the similarity and edge smoothing as an option to adjust the smoothness and sharpness of the edges. Also, a shrinkage window is introduced in order to improve the denoising process, and the entire process is completed by applying a 2×2 fixed weighted mean filter. The properties of the mean filter as set provide a considerable improvement on the denoising process and circumvent image blurring, especially under high impulse noise intensity levels. Also, the weights are set for the fixed mean filter based on probabilities of noise occurrence with the ability to remove the remaining noise in the image with the least effect on non-noisy pixels. Also, the switching property of the denoising filter introduced a new option which is able to denoise the images like the challenging case of the checkerboard even in the presence of high-intensity impulse noise. This combination of filters is shown to yield the best (i.e., highest) structural metrics than any other well-known denoising filter in the presence of different impulse noise intensities. Denoising under this method is shown to preserve edge details with good edge preserving capability as reflected through the highest structural similarity measure between the denoised image and the original noise-free image. The comparative results that were presented indicate that the proposed method outperformed state-of-the art methods and filters which were developed to remove this type of noise. In the edge detection phase, and after the smoothing process attained with SAMFWMF, we observe that our method preserved edge continuity and tracked well the boundaries, especially in high predefined thresholds in relation to the use of maximum-sequence, whose intent was to detect more edges at different threshold intensities while minimizing the effect of noise. This new approach led to a better performance in contrast to other common thresholding methods. For visual appreciation of the optimal outcome, several morphological operations were used on the final image. The results obtained proved that the proposed method yielded a better performance after edge detection even in the presence of high intensity impulse noise.

TABLE II
EXECUTION TIME OF DIFFERENT DENOISNG FILTERS ON CAMERA MAN

	10%	*40%*	*80%*	*90%*
UWMF [41]	0.309078	0.465255	1.002563	1.652859
IBDNDF [25]	0.622326	1.580908	5.242255	11.045418
DBUTMWMF [30]	1.839567	4.002603	3.621499	4.868026
Cascading algorithm [39]	5.268536	6.429431	8.824390	11.972124
Proposed Algorithm	**8.481578**	**8.564322-8.659768**	**8.576215-10.387456**	**8.553084-12.779870**

TABLE III
CORRELATION (β) COMPARISION ON LENA, CAMERA MAN AND COINS FOR DIFFERENT INITIAL ADAPTIVE MEDIAN WINDOW SIZES OF SAMFWMF WITH SWITCH 1

	Initial window=3×3			*Initial window=5×5*			*Initial window=7×7*			*Initial window=9×9*		
	Lena	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*
10%	0.9843	0.9821	0.9933	-	-	-	-	-	-	-	-	-
20%	0.9834	0.9790	0.9921	-	-	-	-	-	-	-	-	-
30%	0.9819	0.9753	0.9906	-	-	-	-	-	-		-	-
40%	0.9548	0.9345	0.9799	0.9528	0.9301	0.9785	-	-	-	-	-	-
50%	0.9524	0.9282	0.9770	0.9508	0.9231	0.9755	-	-	-	-	-	-
60%	0.9499	0.9219	0.9734	0.9478	0.9165	0.9720	-	-	-	-	-	-
70%	0.9459	0.9138	0.9690	0.9432	0.9083	0.9666	-	-	-	-	-	-
80%	0.9160	0.8663	0.9544	0.9136	0.8624	0.9501	0.9106	0.8571	0.9504		-	-
90%	0.8800	0.8122	0.9300	0.8773	0.8077	0.9287	0.8754	0.8071	0.9277	0.8725	0.8011	0.9274

TABLE IV
PSNR COMPARISION ON LENA, CAMERA MAN AND COINS FOR DIFFERENT INITIAL ADAPTIVE MEDIAN WINDOW SIZES OF SAMFWMF WITH SWITCH 1

	Initial window=3×3			*Initial window=5×5*			*Initial window=7×7*			*Initial window=9×9*		
	Lena	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*
10%	29.4967	26.5927	31.7914	-	-	-	-	-	-	-	-	-
20%	29.2478	25.9157	31.1160	-	-	-	-	-	-	-	-	-
30%	28.8934	25.2241	30.3653	-	-	-	-	-	-	-	-	-
40%	24.9708	20.8165	26.8780	24.5841	20.5567	26.6311	-	-	-	-	-	-
50%	24.5506	20.4417	26.3245	24.4105	20.1538	26.1080	-	-	-	-	-	-
60%	24.3336	20.0870	25.7222	24.1757	19.8203	25.5401	-	-	-	-	-	-
70%	24.0080	19.6702	25.0923	23.8142	19.4124	24.7978	-	-	-	-	-	-
80%	21.9598	17.5974	23.2398	21.8666	17.5196	22.9917	21.7218	17.3890	22.9759	-	-	-
90%	20.2855	16.0121	21.2778	20.2264	15.9205	21.2165	20.1514	15.9198	21.2199	20.0942	15.8457	21.2177

TABLE V
STRUCTURAL METRICS COMPARISION ON CHECKERBOARD FOR DIFFERENT INITIAL ADAPTIVE MEDIAN WINDOW SIZES OF SAMFWMF WITH SWITCH 2

	Initial window=3			*Initial window=5*			*Initial window=13*		
	30%	*50%*	*80%*	*30%*	*50%*	*80%*	*30%*	*50%*	*80%*
$\boldsymbol{\beta}$	0.9595	0.9307	0.8076	-	0.9295	0.8054	-	-	0.8007
PSNR	17.0082	14.6801	10.2579	-	14.6175	10.1606	-	-	10.0947
SSIM	0.8417	0.7953	0.6599	-	0.7915	0.6497	-	-	0.6492

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

TABLE VI
CORRELATION (β) COMPARISION ON LENA, CAMERA MAN AND COINS FOR DIFFERENT DENOISNG FILTERS AND SAMFWMF (SWITCH 1)

	Cascading algorithm[39]			*IBDNDF [25]*			*DBUTMWMF [30]*			*UWMF [41]*			**SAMFWMF**		
	Lena	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	***Lena***	***Camera man***	***Coins***
10%	0.9722	0.9580	0.9893	0.9720	0.9567	0.9885	0.9722	0.9581	0.9892	0.9725	0.9586	0.9897	**0.9843**	**0.9821**	**0.9933**
20%	0.9704	0.9529	0.9873	0.9704	0.9518	0.9861	0.9712	0.9543	0.9880	0.9717	0.9566	0.9886	**0.9834**	**0.9790**	**0.9921**
30%	0.9681	0.9472	0.9842	0.9682	0.9464	0.9836	0.9701	0.9508	0.9863	0.9704	0.9533	0.9869	**0.9819**	**0.9753**	**0.9906**
40%	0.9396	0.9024	0.9716	0.9404	0.9031	0.9724	0.9420	0.9080	0.9751	0.9441	0.9128	0.9763	**0.9528-0.9548**	**0.9301-0.9345**	**0.9785-0.9799**
50%	0.9348	0.8905	0.9646	0.9376	0.8950	0.9692	0.9397	0.9011	0.9722	0.9424	0.9085	0.9738	**0.9508-0.9524**	**0.9231-0.9282**	**0.9755-0.9770**
60%	0.9286	0.8777	0.9574	0.9339	0.8865	0.9637	0.9366	0.8951	0.9683	0.9396	0.9021	0.9698	**0.9478-0.9499**	**0.9165-0.9219**	**0.9720-0.9730**
70%	0.9199	0.8616	0.9481	0.9303	0.8799	0.9591	0.9323	0.8852	0.9639	0.9355	0.8927	0.9647	**0.9432-0.9459**	**0.9083-0.9138**	**0.9663-0.9690**
80%	0.8781	0.8067	0.9205	0.9018	0.8340	0.9442	0.8948	0.8295	0.9388	0.9063	0.8479	0.9497	**0.9106-0.9160**	**0.8571-0.8663**	**0.9504-0.9544**
90%	0.8324	0.7413	0.8846	0.8677	0.7859	0.9216	0.8547	0.7726	0.9117	0.8709	0.7956	0.9248	**0.8725-0.8800**	**0.8011-0.8122**	**0.9274-0.9300**

TABLE VII
PSNR COMPARISION ON LENA, CAMERA MAN AND COINS FOR DIFFERENT DENOISNG FILTERS AND SAMFWMF (SWITCH 1)

	Cascading algorithm[39]			*IBDNDF [25]*			*DBUTMWMF [30]*			*UWMF [41]*			**SAMFWMF**		
	Lena	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	***Lena***	***Camera man***	***Coins***
10%	26.918	22.805	29.666	26.884	22.695	29.379	26.940	22.836	29.656	26.981	22.889	29.862	**29.496**	**26.592**	**31.791**
20%	26.675	22.320	28.970	26.661	22.259	28.613	26.802	22.492	29.233	26.861	22.697	29.443	**29.247**	**25.915**	**31.116**
30%	26.345	21.834	28.034	26.382	21.821	27.929	26.656	22.206	28.700	26.681	22.403	28.870	**28.893**	**25.224**	**30.365**
40%	23.434	19.012	25.335	23.514	19.088	25.460	23.611	19.272	25.905	23.771	19.533	26.198	**24.584-24.758**	**20.556-20.816**	**26.631-26.878**
50%	23.109	18.515	24.406	23.318	18.759	25.006	23.448	18.986	25.449	23.646	19.332	25.734	**24.410-24.550**	**20.153-20.441**	**26.108-26.324**
60%	22.719	18.069	23.625	23.071	18.429	24.318	23.244	18.729	24.914	23.456	19.062	25.261	**24.175-24.333**	**19.820-20.087**	**25.540-25.722**
70%	22.242	17.555	22.790	22.847	18.191	23.803	22.981	18.372	24.385	23.181	18.677	24.451	**23.814-24.008**	**19.412-19.670**	**24.797-25.092**
80%	20.296	15.941	20.806	21.220	16.595	22.322	21.131	16.784	22.185	21.453	17.033	22.790	**21.721-21.959**	**17.389-17.597**	**22.975-23.239**
90%	18.814	14.566	19.107	19.814	15.389	20.719	19.669	15.465	20.563	19.943	15.653	20.980	**20.094-20.285**	**15.845-16.012**	**21.217-21.277**

TABLE VIII
SSIM COMPARISION ON LENA, CAMERA MAN AND COINS FOR DIFFERENT DENOISNG FILTERS AND SAMFWMF (SWITCH 1)

	Cascading algorithm[39]			*IBDNDF [25]*			*DBUTMWMF [30]*			*UWMF [41]*			**SAMFWMF**		
	Lena	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	*Lena*	*Camera man*	*Coins*	***Lena***	***Camera man***	***Coins***
10%	0.9559	0.9220	0.9342	0.9542	0.9174	0.9312	0.9574	0.9227	0.9348	0.9580	0.9251	0.9373	**0.9744**	**0.9576**	**0.9634**
20%	0.9401	0.8997	0.9225	0.9384	0.8947	0.9180	0.9468	0.9052	0.9270	0.9489	0.9128	0.9307	**0.9635**	**0.9426**	**0.9558**
30%	0.9209	0.8741	0.9072	0.9199	0.8671	0.9046	0.9354	0.8885	0.9174	0.9361	0.8955	0.9202	**0.9496**	**0.9225**	**0.9463**
40%	0.8893	0.8281	0.8695	0.8902	0.8227	0.8716	0.9016	0.8438	0.8815	0.9043	0.8497	0.8892	**0.9048-0.9156**	**0.8508-0.8630**	**0.8898-0.9022**
50%	0.8585	0.7867	0.8396	0.8662	0.7886	0.8515	0.8825	0.8141	0.8658	0.8848	0.8204	0.8706	**0.8861-0.8962**	**0.8221-0.8370**	**0.8725-0.8875**
60%	0.8190	0.7418	0.8046	0.8369	0.7553	0.8280	0.8599	0.7851	0.8442	0.8633	0.7929	0.8509	**0.8641-0.8744**	**0.7940-0.8080**	**0.8521-0.8670**
70%	0.7724	0.6875	0.7659	0.8099	0.7245	0.8039	0.8312	0.7454	0.8187	0.8331	0.7588	0.8225	**0.8343-0.8456**	**0.7604-0.7736**	**0.8240-0.8427**
80%	0.6895	0.6103	0.6829	0.7693	0.6699	0.7614	0.7614	0.6508	0.7151	0.7791	0.6921	0.7727	**0.7772-0.7995**	**0.6901-0.7129**	**0.7714-0.7958**
90%	0.5987	0.5234	0.5981	0.6976	0.6074	0.6939	0.6817	0.5640	0.6353	0.7039	0.6136	0.7018	**0.7032-0.7253**	**0.6116-0.6315**	**0.7011-0.7169**

TABLE IX
STRUCTURAL METRICS COMPARISION ON CHECKERBOARD FOR DIFFERENT DENOISNG FILTERS AND SAMFWMF (SWITCH 2)

	Cascading algorithm[39]			*IBDNDF [25]*			*DBUTMWMF [30]*			*UWMF [41]*			**SAMFWMF**		
	30%	*50%*	*80%*	*30%*	*50%*	*80%*	*30%*	*50%*	*80%*	*30%*	*50%*	*80%*	***30%***	***50%***	***80%***
β	0.9587	0.9270	0.7921	NaN	NaN	NaN	0.9076	0.8470	0.4251	0.9520	0.9202	0.7860	**0.9595**	**0.9295-0.9270**	**0.8002-0.8076**
PSNR	16.9202	14.5194	10.0525	NaN	NaN	NaN	12.7868	10.1671	6.8285	16.2001	13.9772	9.6609	**17.0082**	**14.6175-14.6801**	**10.0947-10.2579**
SSIM	0.7457	0.7027	0.4793	NaN	NaN	NaN	0.1626	0.1427	0.2089	0.7391	0.7467	0.6202	**0.8417**	**0.7915-0.7953**	**0.6492-0.6599**

TABLE X
CORRELEATION (β) AND PSNR COMPARISON OF OTHER DENOISNG FILTERS WITH AND WITHOUT FIXED WEIGHTED MEAN FILTER ON LENA

	Cascading Algorithm [39]				*IBDNDF [25]*				*DBUTMWMF [30]*				*UWMF [41]*			
	Without mean filter	*With mean filter*	*Without mean filter*	*With mean filter*	*Without mean filter*	*With mean filter*	*Without mean filter*	*With mean filter*	*Without mean filter*	*With mean filter*	*Without mean filter*	*With mean filter*	*Without mean filter*	*With mean filter*	*Without mean filter*	*With mean filter*
	β	*β*	*PSNR*	*PSNR*	*β*	*β*	*PSNR*	*PSNR*	*β*	*β*	*PSNR*	*PSNR*	*β*	*β*	*PSNR*	*PSNR*
10%	0.9722	0.9840	26.918	29.406	0.9720	0.9838	26.884	29.364	0.9722	0.9841	26.940	29.452	0.9725	0.9843	26.981	29.431
20%	0.9704	0.9823	26.675	28.971	0.9704	0.9823	26.661	28.989	0.9712	0.9832	26.802	29.222	0.9717	0.9834	26.861	29.232
30%	0.9681	0.9799	26.345	28.432	0.9682	0.9805	26.382	28.634	0.9701	0.9820	26.656	28.745	0.9704	0.9821	26.681	29.018
40%	0.9396	0.9505	23.434	24.376	0.9404	0.9516	23.514	24.493	0.9420	0.9521	23.611	24.495	0.9441	0.9548	23.771	24.526
50%	0.9348	0.9455	23.109	23.968	0.9376	0.9484	23.318	24.222	0.9397	0.9487	23.448	24.398	0.9424	0.9511	23.646	24.326
60%	0.9286	0.9381	22.719	23.423	0.9339	0.9423	23.071	23.965	0.9366	0.9427	23.244	24.162	0.9396	0.9451	23.456	24.387
70%	0.9199	0.9306	22.242	22.930	0.9303	0.9401	22.847	22.543	0.9323	0.9436	22.981	23.843	0.9355	0.9455	23.181	24.005
80%	0.8781	0.9104	20.296	21.855	0.9018	0.9103	21.220	21.675	0.8948	0.9048	21.131	21.643	0.9063	0.9167	21.453	21.934
90%	0.8324	0.8375	18.814	19.049	0.8677	0.8612	19.814	20.065	0.8547	0.8654	19.669	20.088	0.8709	0.8792	19.943	20.339

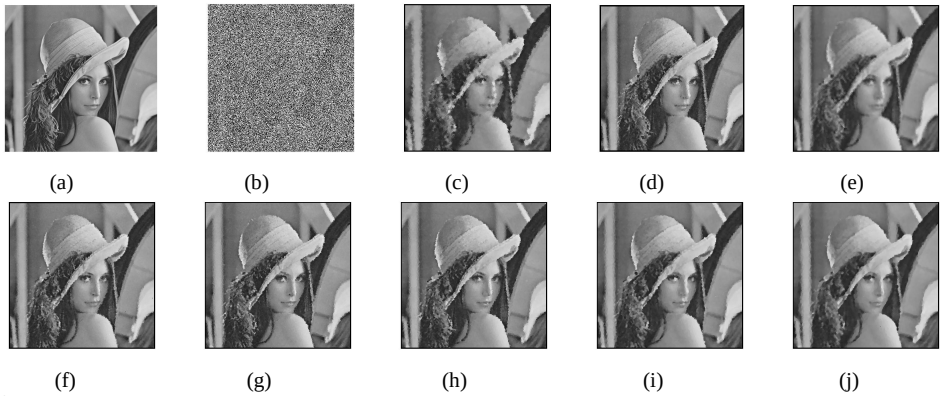
TABLE XI
FOM CAMPARISON BETWEEN PROPOSED EDGE DETETION AND CANNY

σ_g	1		2	
σ_h	5	15	5	15
σ_p	0.5	1	0.5	1
σ_n	1	2	1	2
Canny	1.1627	1.4009	1.1223	1.3539
Proposed Algorithm	**0.0607**	**0.0377**	**0.0783**	**0.0237**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

TABLE XII
FOM CAMPARISON BETWEEN PROPOSED EDGE DETECTION ALGORITHM AFTER APPLYING SAMFWMF PROCESS, PROPOSED EDGE DETECTION ALGORITHM AFTER APPLYING UWMF PROCESS (WITH AND WITHOUT FIXED WEGHTED MEAN FILTER AS A POST PROCESSING), CANNY EDGE DETECTION AFTER APPLYING SAMFWMF PROCESS

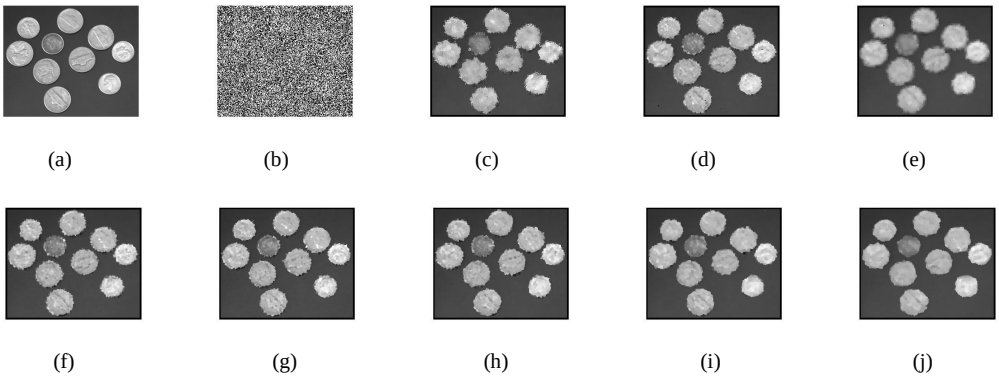
σ_g	1				2			
σ_h	5		15		5		15	
σ_p	0.5		1		0.5		1	
I_m	10%	30%	10%	30%	10%	30%	10%	30%
Canny	6.3127	7.8594	6.4761	7.8751	6.3561	7.9511	7.0012	7.9724
UWMF – With post processing	5.5767	4.4949	5.5123	4.6514	5.4337	4.8386	5.2711	4.2581
UWMF – Without post processing	4.8765	1.8743	4.6754	1.9876	4.1132	1.3241	3.9854	1.2190
Proposed Algorithm	**4.3306**	**0.6555**	**4.1241**	**0.3072**	**3.9849**	**0.3318**	**3.6782**	**0.3058**



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

(f) (g) (h) (i) (j)

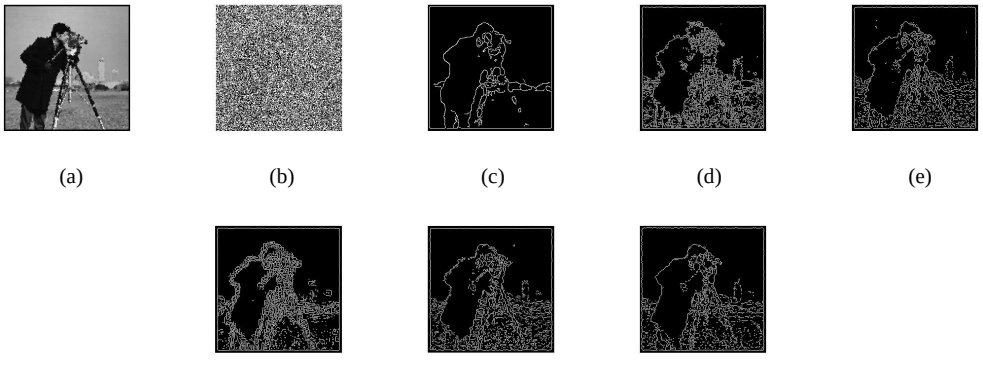
Fig. 16. After apply the filter with 90% impulse noise intensity on Lena, a) Original image b) Noisy image c) Cascading algorithm [39] d) IBDNDF [25] e) DBUTMWMF [30] f) UWMF [41] g) SAMFWMF(initial window size=3) h) SAMFWMF (initial adaptive median window size=5) i)) SAMFWMF (initial adaptive median window size=7) j)) SAMFWMF (initial adaptive median window size=9)



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

(f) (g) (h) (i) (j)

Fig. 17. After apply the filter with 90% impulse noise intensity on Coins, a) Original image b) Noisy image c) Cascading algorithm [39] d) IBDNDF [25] e) DBUTMWMF [30] f) UWMF [41] g) SAMFWMF(initial adaptive median window size=3) h) SAMFWMF (initial adaptive median window size=5) i) SAMFWMF (initial adaptive median window size=7) j)) SAMFWMF (initial window size=9)



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

(f) (g) (h)

Fig. 18. After edge detection with 90% impulse noise on Camera man, T=0.2 (normalized) a) Original image b) Noisy image c) Edge detection with canny (with σ=1, by increasing the σ, more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [39] process e) Proposed edge detection algorithm after IBDNDF [25] process f) Proposed edge detection algorithm after DBUTMWMF [30] process g) Proposed edge detection algorithm after UWMF [41] process h) Proposed edge detection algorithm after SAMFWMF process

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2018.2857448, IEEE Transactions on Image Processing

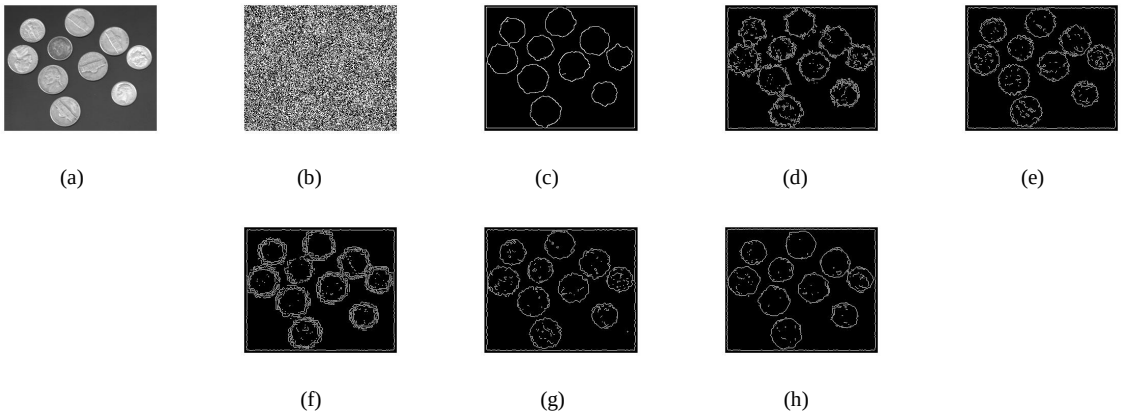


Fig. 19. After edge detection with 90% impulse noise on Coins, T=0.3 (normalized) a) Original image b) Noisy image c) Edge detection with canny (with σ=1, by increasing the σ, more details will be lost) after SAMFWMF process d) Proposed edge detection algorithm after cascading algorithm [39] process e) Proposed edge detection algorithm after IBDNDF [25] process f) Proposed edge detection algorithm after DBUTMWMF [30] process g) Proposed edge detection algorithm after UWMF [41] process h) Proposed edge detection algorithm after SAMFWMF process

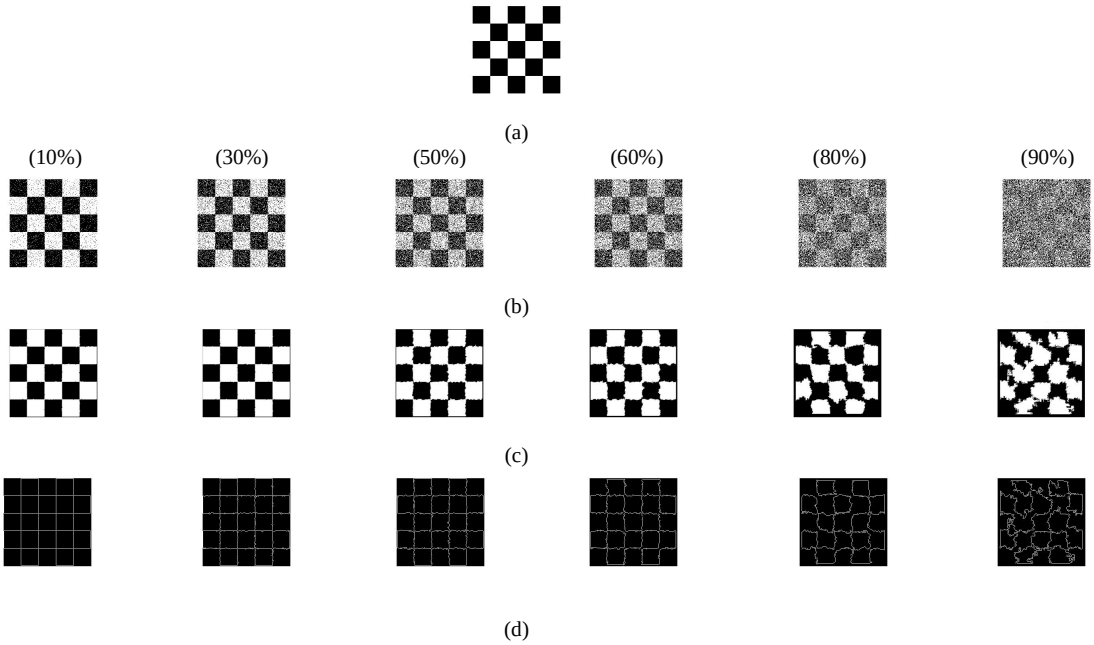


Fig. 20. After apply the SAMFWMF and proposed edge detection algorithm on checker board with switch 2 a) Original image b) Noisy images c) After denoising d) After edge detection

TABLE XIII
SUMMARY OF ACRONYMS AND THEIR CORRESPONDING METHODOLOGIES

Acronym	Corresponding methodology
Switching Adaptive Median (SAM)	This is a technique for denoising and it switches between two states of adaptive median filter in which adaptive median filter is flexible and adapts itself to the predefined conditions
Fixed Weighted Mean (FWM)	This technique for denoising, calculates the averaging weighted mean of neighboring pixels in which the size of the selected window is fixed
Shrinkage window	This technique is used to improve the denoising in which the size of the window is shrunk according to predefined condition
Gradient based edge detection	This technique is used to detect the image edges in which a kernel obtain based on the gradient of the image and the kernel convolves with the image in order to edge detecting
Non-maximum suppression	This technique is used to track the edges based on the angle of gradient
Maximum sequence	This technique is used to keep the connectivity of the edges and remove the noisy pixels after edge detection
Thresholding	This technique is used to obtain a binary image from grayscale one
Morphological operation	This technique is used for trimming the binary image in order to better visualization

ACKNOWLEDGMENT

We are grateful for the continued support from the National Science Foundation (NSF) under NSF grants CNS-1532061, CNS-1551221, CNS-1042341, CNS-1429345, and CNS 1338922. We also greatly appreciate the support of the Ware Foundation.

REFERENCE

[1] J. Canny, “A Computational Approach to Edge Detection,” IEEE Trans. Pattern Anal. Mach. Intell, vol.PAMI-8, no.6, Nov. 1986.

[2] S. Xie, Z. Tu, “Holistically-nested edge detection,” In Proc. IEEE Int. Conf. Comput. Vision, pp. 1395–1403 (2015).

[3] M. E. Yukse, “Edge detection in noisy images by neuro-fuzzy processing,” Elsevier, Int. J. Electron. Commun. vol.61 pp.82 – 89, 2007.

[4] Y. Shi, Q. Zhao, F. Guo, Y. Zhu, “A Fast Edge Detection Model In Presence of Impulse Noise,” in Proc. Springer 8th Int. Conf. Image Graph., Tianjin, China, vol.9217, pp. 64-74, Aug. 2015.

[5] P. Jain, V. Tyagi, "A survey of edge-preserving image denoising methods," Springer, Inform. Syst. Front. , vol.18, no.1, pp. 159–170, Feb. 2016.

[6] C. Pal, Amlan Chakrabarti, Ranjan Ghosh, "A Brief Survey of Recent Edge-Preserving Smoothing Algorithms on Digital Images," Elsevier, Procedia Comput. Sci., pp.1-40, Mar. 2015.

[7] R. C. Gonzalez, R.E. Woods, "Digital image processing," 3th ed., Upper Saddle River, NJ, Prentice-Hall, 2006.

[8] L. I. Rudin, S. Osher, E. Fatemi, "Nonlinear total variation based noise removal algorithms," Elsevier, Physica D J., vol.60, no.1-4, pp.259–268, 1992.

[9] A. Chambolle, "An Algorithm for Total Variation Minimization and Applications," J. Math. imaging Vis., vol. 20, no.1, pp.89-97, Jan. 2004.

[10] P. Perona, J. Malik, "Scale-Space and Edge Detection Using Anisotropic Diffusion," IEEE Trans. Pattern Anal. Mach. Intell., vol.12, no.7, .Jul. 1990.

[11] G. Gerig, O. Kubler, R. Kikinis, F. A. Jolesz, "Nonlinear Anisotropic Filtering of MRI Data," IEEE Trans. Med. Imag., vol.1, no.2, Jun. 1992.

[12] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images", in Proc. IEEE, 6th Int. Conf. comput. Vis., Bombay, India, Jan. 1998.

[13] K. He, J. Sun, X. Tang, "Guided Image Filtering," IEEE Trans. Pattern Anal. Mach. Intell., vol.35, June. 2013.

[14] A. Buades, Bartomeu Coll, Jean-Michel Morel, "A non-local algorithm for image denoising," in Proc. IEEE Computer society Conf. Comput. Vis. Pattern Recog., vol.2, pp.60- 65, Jun. 2005.

[15] R. Yang, L. Yin, M. Gabbouj, J. Astola, T. Neuvo, "Optimal weighted median filtering under structural constraints" IEEE Trans. Signal Process., vol.43, no.3, pp. 591–604, 1995.

[16] G. Qiu, An improved recursive median filtering scheme for image processing", IEEE Trans. Image Process., vol.5, no.4, pp. 646–648, 1996.

[17] S. Vishaga, S. L. Das, "A Survey on Switching Median Filters for Impulse Noise Removal," in Proc. IEEE Int. Conf. Circuit, Power Comput. Tech., Mar. 2015.

[18] S. J. Ko and Y. H. Lee, "Center Weighted Median Filter," IEEE Trans. Circuits Systs., vol. 3, no.9, pp 984-993, Sep. 1991.

[19] Y. Dong, S. Xu, "A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise," IEEE Signal Process. Lett., vol. 14, no. 3, Mar. 2007.

[20] V. V. Khryaschev, A. L. Priorov, I. V. Apalkov, P. S. Zvonarev, "Impulse Denoising Using Adaptive Switching Median Filter," in Proc. IEEE 2th Int. Conf. Image process., Sep. 2005.

[21] S. Esakkirajan, T. Veerakumar, A. N. Subramanyam, C. H PremChand, "Removal of High Dendity Salt and Pepper noise Through Modified Decision based Unsymmetrical Trimmed Median Filter," IEEE Signal Process. Lett., vol.18, no.5, May 2011.

[22] H. L. Eng, K. K. Ma, "Noise adaptive soft-switching median filter," IEEE Trans. Image Process., vol.10, no.2, pp. 242–251, 2001.

[23] R. Yang, L. Yin, M. Gabbouj, J. Astola, T. Neuvo, "A New impulse detector for switching median filters," IEEE Signal Process. Lett., vol.9, no.11, , pp. 360–363, 2002.

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

[25] I. F. Jafar, R. A. AlNa'mneh, K. A. Darabkh, "Efficient Improvements on the BDND Filtering Algorithm for the Removal of High-Density Impulse Noise," IEEE Trans. Image Process., Vol. 22, No. 3, Mar. 2013.

[26] X. D. Jiang, "Iterative truncated arithmetic mean filter and its properties," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1537–1547, Apr. 2012.

[27] Z. Miao, X. Jiang, "Further properties and a fast realization of the iterative truncated arithmetic mean filter," *IEEE Trans. Circuits Systems II: Exp. Briefs*, vol. 59, no. 11, pp. 810–814, 2012.

[28] Z. Miao, X. Jiang, "Weighted iterative truncated mean filter," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4149–4160, 2013.

[29] Z. Miao, X. Jiang, "Additive and exclusive noise suppression by iterative trimmed and truncated mean algorithm," *Signal Process.*, vol. 99, pp. 147-158, 2014.

[30] K. Vasanth, T. G. Manjunath, N. Raj "A Decision Based Unsymmetrical Trimmed Modified Winsorized Mean Filter for the Removal of High Density Salt and Pepper Noise in images and Videos," Elsevier, Procedia Comput. Sci., No. 48, pp.29-36, Aug. 2015.

[31] H. Hwang, R.A. Haddad, "Adaptive Median Filters: New Algorithms and Results," IEEE Trans. Image Process., vol.4, pp.499-502, Apr. 1995.

[32] Z. Wang, D. Zhang, "Progressive Switching Median Filter for Removal of Impulse Noise from highly Corrupted Images," IEEE Trans. Circuits Syst. I, Fundam. Theory Appl., vol.46, no.1, Jan. 1999.

[33] S. Balasubramanian, S. Kalishwaran, R. Muthuraj, D. Ebenezer, and V. Jayaraj, "An efficient non-linear cascade filtering algorithm for removal of high density salt and pepper noise in image and video sequence," in Proc. IEEE Int. Conf. Control Autom. Commun. Energy Conservation., pp. 1-6., Jun. 2009.

[34] K. Aiswarya, V. Jayaraj and D. Ebenezer, "A New and Efficient Algorithm for the Removal of High Density Salt and Pepper Noise in Images and Videos," in Proc. IEEE 2th Int. Conf. Comput. Mod. Simulation., pp. 409-413, 2010.

[35] M. S. Nair, K. Revathy, R. Tatavarti, "An Improved Decision Based Algorithm for Impulse Noise Removal," in Proc. IEEE Congress on Image and Signal Process., pp. 426–431, May 2008.

[36] T. Veerakumar, S. Esakkirajan, I. Vennila, "An Approach to Minimize Very High Density Salt and Pepper Noise through Trimmed Global Mean," Int. J. Comput. Appl., vol. 39, no. 12, Feb. 2012.

[37] P. S. Jayasree, P. Raj, P. Kumar, R. Siddavatam, S. P. Ghrera, "A Fast Novel Algorithm for Salt and Pepper Image Noise Cancellation using Cardinal B-Splines," Springer, Signal Image and Video Process. J, vol.7, no. 6, pp.1145-1157, Nov. 2013.

[38] S. Balasubramanian, S. Kalishwaran, R. Muthuraj, D. Ebenezer, V. Jayaraj, "An Efficient Non Linear Cascade Filtering Algorithm for Removal of high Density Salt and Pepper Noise in Image and Video Sequence," in Proc. IEEE Int. Conf. Control Autom. Commun. Energy Conservation, pp. 1–6, Jun. 2009.

[39] A. Dash, S. K. Sathua, "High Density Noise Removal By Using Cascading Algorithms," in Proc. IEEE 5th Int. Conf. Advanced Comput. Commun. Tech., pp.96-101, Feb. 2015.

[40] M. T. Raza, S. Sawant, "High density salt and pepper noise removal through decision based partial trimmed global mean filter," in Proc. IEEE Int. Conf. Engineering (NUiCONE), Nirma University, pp.1-5, Dec. 2012.

[41] C. Kandemir, C. Kalyoncu, Ö. Toygar, "A weighted mean filter with spatial-bias elimination for impulse noise removal," Elsevier, Digital Signal Process., vol. 46, pp. 164–174, Nov. 2015.

[42] Z. Zhou, "Cognition and Removal of Impulse Noise With Uncertainty," IEEE Trans. Image Process., vol. 21, no. 7, Jul. 2012.

[43] C. Kalyoncu1, Ö. Toygar, H. Demirel, "Interpolation-based impulse noise removal," IET Image Process., vol.7, no.8, pp.777-785, Nov.2013.

[44] K. M Moon, M. D. Patil, B. Parmar, "Image Restoration Using Adaptive Switching Median Filter," in Proc. IEEE Int. Conf. Comput. Intell. Comput. Research, pp.439, Dec.2010.

[45] J. L. Mateo, A. Fernandez-Caballero, "Finding out general tendencies in speckle noise reduction in ultrasound images," Elsevier, Expert Syst. Appl., vol. 36, no.4, pp.7786-7797, May 2009.

[46] Z. Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, Eero P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Trans. Image Process., vol. 13, no. 4, Apr. 2004.

[47] V. D. Heyden, "Evaluation of edge detection algorithms," IET 3th Conf. Image Process. its Appl., pp.618-622, Jul. 1989.

[48] M. C. Shin, D. B. Goldgof, K. W. Bowyer, S. Nikiforou, "Comparison of Edge Detection Algorithms Using a Structure from Motion Task," IEEE Trans. System, Man, and Cybernetics—Part B: Cybernetics, vol. 31, pp. 589-601, Aug. 2001.

[49] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," IEEE Trans. Syst. Man Cybern, vol.SMC vol.SMC-9, no.1, Jan. 1979.