Journal of Imaging Science and Technology® 57(6): 060507-1–060507-9, 2013.

Edge Preserving Impulse Noise Reduction

Younghun Song and Yunsang Han
The Graduate School of Advanced Image Science, Multimedia and Film, Chung-Ang University, #221 Heuksuk-Dong, Dongjak-Gu, Seoul, 156-756, Korea

Jeong-Su Oh
Department of Image Science and Engineering, Pukyong National University, 599-1, Daeyeon-3Dong, Nam-Gu, Pusan, 608-737, Korea

Sangkeun Lee
The Graduate School of Advanced Image Science, Multimedia and Film, Chung-Ang University, #221 Heuksuk-Dong, Dongjak-Gu, Seoul, 156-756, Korea
E-mail: sangkny@cau.ac.kr

Abstract. *In this article, we propose an edge-directed switching median filter that considers the local correlation of pixels and edge directions for impulse noise reduction. The proposed algorithm consists of two main steps, detection and correction. In the first step, the impulse noise is detected using minimum and maximum values in a scalable mask. In the second step, a corrupted pixel is corrected using the local correlation between the uncorrupted pixels in the mask. This step specifically performs edge-directed filtering using principal component analysis to preserve the edge and detail information in the highly corrupted image. The experimental results showed that the proposed method can reduce impulse noise significantly and preserve more edge information than the existing state-of-the-art methods. In addition, the proposed method outperforms existing methods for highly corrupted images by an average of 6.47 dB. Therefore, we believe that the proposed method can be a useful tool for removing impulse noise in the field of infrared related devices and digital cameras.* © *2013 Society for Imaging Science and Technology.*
[DOI: 10.2352/J.ImagingSci.Technol.2013.57.6.060507]

INTRODUCTION

Impulse noise is generated by a defective camera sensor or by the acquisition of a digital image through a corrupted cable.[1] In particular, impulse noise frequently appears in infrared images that have a low contrast ratio in comparison with visible ray images, and thus it may appear in infrared imaging systems used with military weapons. If the imaging system generates severe impulse noise, the devices do not work well,[2] thus making the performance of noise reduction algorithms is critical. Impulse noise is generally classified into salt-and-pepper and random-valued noise. In particular, salt-and-pepper noise has the characteristics of minimum and maximum values in an image.[1,3]

Many approaches have been studied to reduce the impulse noise. One of these approaches is the standard median filter (SMF),[4] which is one of the non-linear filters used to remove impulse noise effectively in a grayscale image.[5] However, this method degrades the edge and detail information because it is applied uniformly to the entire image. To resolve this degradation problem, many algorithms that improve the SMF have been designed. The adaptive median filter (AMF)[6] employs an adaptive mask size to distinguish the corrupted and uncorrupted pixels. Weighted median filters[7] including the center-weighted median filter[8] give a high weight to the median value in a given mask.

The switching-based median filter has been reported to improve the methods listed in the previous paragraph, and is a state-of-the-art approach for removing impulse noise effectively.[9,23] Switching-based median filters generally consist of two steps. The first step is to detect the corrupted pixels, and the second step is to perform filtering. Consequently, the switching-based median filters reduce impulse noise and also preserve the image's edge and detail information effectively.[9–20,22–25] An example of a switching-based median filter is Luo's approach (LUO),[17] which calculates the noise ratio for each pixel during the detection step, and removes noise using a modified SMF with information about noise ratio. Another example is the simple adaptive median (SAM) filter[18] that detects noise using two peak values of the image's dynamic range, and performs filtering with an adaptive mask size. LUO and SAM effectively remove noise in images corrupted with low-density noise. However, they show very poor performance in images corrupted with more than 50% impulse noise, because the noise detection step used by these two methods simply uses the peak values in the corrupted image. The iterative adaptive switching median filter (IASMF)[15] improves the weaknesses shown by LUO and SAM by performing filtering that only considers the uncorrupted pixel values with an adaptive mask size. However, IASMF has shown weaknesses in preserving the image's edge and detail regions when the images are corrupted with high noise density. All the existing algorithms that only use a median value in a given mask generally do

Received May 27, 2012; accepted for publication Nov. 13, 2013; published online Jan. 3, 2014. Associate Editor: Miguel Lopez-Álvarez.
1062-3701/2014/57(6)/060507/9/$25.00

not preserve the image's edge and detail contents during the filtering step.

In our study, we improve the switching-based approach by applying the median and average filtering adaptively to preserve the image details according to local correlation in the given mask. Furthermore, we preserve edge components by employing additional edge-directed filtering around edge regions using principal component analysis (PCA) only for highly corrupted images that have a noise density of over 60%.

The proposed system consists of two main steps: detection and edge preserving correction for highly corrupted images. In the noise detection step, we obtain the corrupted pixel using the extreme values in a scalable mask. In the correction part, filtering is performed on the uncorrupted pixels in a mask with adaptive mask size. If there is no estimate for uncorrupted pixels in a given mask, the mask size is gradually increased up to a predefined size, and the uncorrupted pixels are sorted in ascending order. A current corrupted pixel is then substituted using the local correlation of uncorrupted pixels in the mask. Additional edge-directed filtering for highly corrupted images is performed using PCA.

The main difference between the existing and proposed schemes is that the proposed scheme considers the correlation of local neighborhoods, whereas the existing methods take the median value of the neighbors for a corrupted pixel. In addition, our algorithm performs additional filtering that considers the edge direction to preserve the image edge and details for highly corrupted images. The proposed technique outperforms the conventional methods, including SMF, AMF, LUO, SAM, and IASMF, across a wide range of salt-and-pepper noise densities, ranging from 10 to 90%. In particular, the local correlation-based correction and edge-directed filtering steps preserve detail information in terms of visual observation and peak signal-to-noise ratio (PSNR) scores.

The remainder of this article is structured in the following manner: the following section presents the proposed method. The experimental results are reported in the next section, and the fourth section concludes this article with some discussions.

PROPOSED ALGORITHM

The proposed algorithm consists of noise detection and edge preserving correction steps. Noise pixels are identified by the minimum and maximum values in a scalable mask in the detection step. Correction is then performed by considering the relationship with uncorrupted pixels in a given mask. Here, a corrupted pixel is replaced with the average value of the pixels, which have a strong correlation. Edge-directed filtering using PCA is performed only for highly corrupted images. It is noted that the image borders are padded symmetrically with neighboring pixel values to isolate the boundary problem. Figure 1 shows the overall block diagram of the proposed approach.

Noise detection

In this step, we detect the corrupted pixels using the local maximum and minimum values in the given mask. However, the noise is limited to salt-and-pepper noise if global maximum and minimum values are used to detect noise. First, the maximum $y_{i,j}^{\max}$ and minimum $y_{i,j}^{\min}$ values in a 5×5 mask w_n centered at (i, j) are checked using the following equations:[21]

$$y_{i,j}^{\min} = \min\{y_{i-s,j-t} | (s,t) \in w_n\} \tag{1}$$

$$y_{i,j}^{\max} = \max\{y_{i-s,j-t} | (s,t) \in w_n\}, \tag{2}$$

with

$$w_n = \{(s,t) : -2 \le s, t \le 2\}. \tag{3}$$

Next, we determine if a current pixel location (i, j) is corrupted only by minimum and maximum values. A flag image $\boldsymbol{f}$, where uncorrupted and corrupted pixels are denoted as 0 and 1, respectively, is generated (see Figure 2).

$$f_{i,j} = \begin{cases} 1, & \text{if } y_{i,j} = y_{i,j}^{\min} \text{ or } y_{i,j} = y_{i,j}^{\max} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Noise correction

To correct the corrupted pixels detected in the previous step, a supporting mask size is first determined. Unless there are at least two uncorrupted pixels in the mask, the mask size can be increased from 3×3 up to 21×21 until the condition is met. Otherwise, the uncorrupted pixels in the mask are sorted in ascending order as follows:

$$u_{(1)} \le u_{(2)} \le \cdots \le u_{(m)}, \quad u_{(k)} \in \boldsymbol{U} \text{ for } k = 1, 2, \ldots, m. \tag{5}$$

Here, $\boldsymbol{U}$ and $\boldsymbol{u}$ denote a set and its sorted elements of the uncorrupted pixel intensity values in a given mask, respectively. The differences are then computed to determine the correlation between the pixels as follows:

$$\boldsymbol{N}_{diff} = \{\boldsymbol{n}_{diff}(k) | k = 1, 2, \ldots, m-1\}, \tag{6}$$

where

$$\boldsymbol{n}_{diff}(k) = u_{(k)} - u_{(k+1)} \tag{7}$$

$\boldsymbol{N}_{diff}$ and $\boldsymbol{n}_{diff}$ denote a set and its elements of the differences between pixels. A set of difference average $\boldsymbol{N}_{avg}$ can be calculated as follows:

$$\boldsymbol{N}_{avg} = \{\boldsymbol{n}_{avg}(k) | k = 1, 2, \ldots, m-1\} \tag{8}$$

with

$$\boldsymbol{n}_{avg}(k) = \frac{1}{2} |\boldsymbol{n}_{diff}(k) + \boldsymbol{n}_{diff}(k+1)|. \tag{9}$$

The local correlation can be inferred from the sorted average $\boldsymbol{N}_{avg}$. The large value of $\boldsymbol{N}_{avg}$ indicates a weak correlation

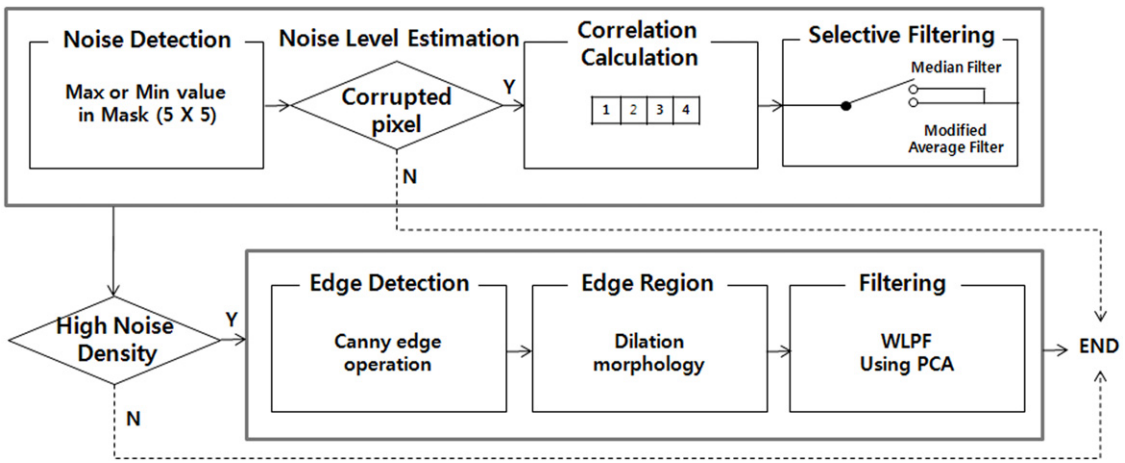


Figure 1. Block diagram of the proposed algorithm.

between the given pixels, whereas a small value denotes a strong correlation. From this observation, a threshold value T_c is defined by the standard deviation of uncontaminated pixels within the mask, and it represents the correlation of the pixels. The reason for using an adaptive threshold is to calculate the pixel correlation required to achieve the best performance for the images corrupted by various noise densities. Therefore, if a pixel value is lower than T_c, a current noise pixel is replaced with the average value of the pixels. Otherwise, a noise pixel is corrected by the median value of the pixels. MATLAB function median $(\cdot)$ is used to select the median value. The final restored pixel value $R_{i,j}$ is calculated by

$$R_{i,j} = \begin{cases} \dfrac{1}{l}\displaystyle\sum_{k=1}^{l} u_{(k)}, & \text{if } \boldsymbol{n}_{avg(k)} < T_c \\ \text{median } (U), & \text{otherwise} \end{cases} \tag{10}$$

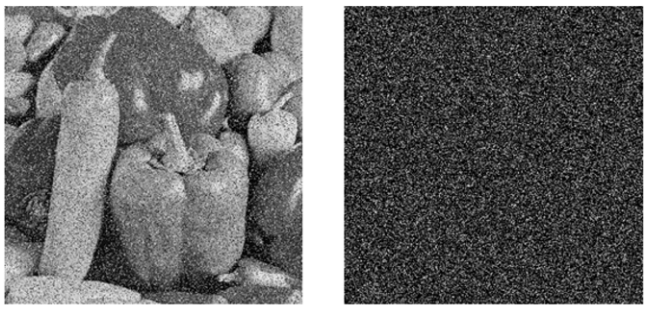
where l denotes the number of the uncorrupted pixels $\boldsymbol{U}$ in the given mask, and the threshold is defined as

$$T_c = \sqrt{\frac{1}{m-1}\sum_{k=1}^{m}(u_{(k)} - \bar{u})^2}.$$

Here, $\bar{u}$ indicates the mean value of $\boldsymbol{u}$.

Edge-directed filtering

To preserve the image edges for highly corrupted image, we perform additional filtering, namely, edge-directed filtering. Specifically, we estimate the noise density of the corrupted image in the noise detection step. If the estimated noise exceeds 60% which has been selected empirically, we detect the edge map and perform the edge-directed filtering. It should be noted that directional filtering is employed only around large edges, including curves, because most staircase artifacts in images filtered through the correction step are evident under these conditions. First, we determine whether the given image requires edge-directed filtering based on the number of edge pixels. If the number exceeds 10% of the pixels in the image, we proceed with directional filtering. Therefore, both conditions trigger the edge-directed filtering because it requires heavy computations. Additionally, trade-offs can be made between the percentage of edge pixels in the resulting image and computation time.



(a) (b)

Figure 2. Flag image by noise detection step: (a) original corrupted image (Peppers); and (b) noise detection image.

In this step, we obtain the edge information E^c using a canny edge operator.[1] Next, we apply morphological dilation on the edge pixels to isolate the edge region D as

$$D(i,j) = \{w_e \cap E^c(i,j) \neq \varphi | (s,t) \in w_e\} \tag{11}$$

where w_e indicates a structuring element; a 6×6 rectangular shape is used in practice. Figure 3 presents the procedures for generating the edge region.

In the edge regions, we apply directional low-pass filtering using PCA that can be summarized in the following five steps:

(1) To find the edge direction, we use a local autocorrelation metric $\boldsymbol{C}(i,j)$, considering the local variation of the patched area[26] as

$$\boldsymbol{C}(i,j) = \sum_{\Delta i, \Delta j \in \omega} (\boldsymbol{I}(i,j) - \boldsymbol{I}(i+\Delta i, j+\Delta j))^2 \tag{12}$$

where $\boldsymbol{I}$ is a function of the image, and ω is a window around the central pixel (i,j). In practice, a 5×5 rectangular window

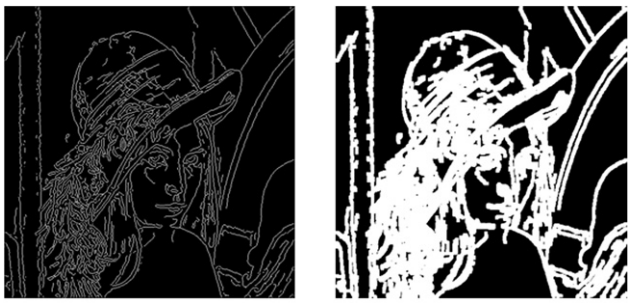


Figure 3. An example of generating the edge region: (a) edge pixels of restored image; and (b) edge region after applying dilation operator on the edge pixels.

was used. Next, the shifted image, approximated by Taylor expansion, is given by

$$\boldsymbol{I}(i+\Delta i, j+\Delta j) \approx \boldsymbol{I}(i,j) + [\boldsymbol{I}_i(i,j)\boldsymbol{I}_j(i,j)] \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix} \quad (13)$$

where $\boldsymbol{I}_i$ and $\boldsymbol{I}_j$ are derived functions of the i and j directions, respectively. Equation (13) can be rewritten as

$$C(i,j) = [\Delta i \Delta j]\boldsymbol{H} \begin{bmatrix} \Delta i \\ \Delta j \end{bmatrix} \quad (14)$$

where $\boldsymbol{H}$ is a 2×2 structure tensor, a form of Hessian matrix, defined as

$$\boldsymbol{H} = \begin{bmatrix} I_i^2 & I_{ji} \\ I_{ji} & I_j^2 \end{bmatrix}. \quad (15)$$

By analyzing $\boldsymbol{H}$, we can compute the direction and strength of an edge, which play an important role in this work.

(2) Through PCA of the Hessian matrix, edge strength was calculated,[26] using two Eigenvalues ($\lambda_1 \gg \lambda_2$), as

$$E_s = \frac{\lambda_1}{\lambda_2}. \quad (16)$$

(3) Between the two corresponding Eigenvectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, the minor axis $\boldsymbol{v}_2$ is chosen since this indicates the edge direction of a given pixel.

(4) The average value I_{Ed} is computed along the edge direction by

$$I_{Ed} = \frac{1}{N}\sum_{i=1}^{N} z_i \quad (17)$$

where z_i indicates N pixel values along the minor axis in a 5×5 window.

(5) Then, the final output $\hat{I}$ is defined by combining the input I and the edge-directed averaged value I_{Ed} as

$$\hat{I} = \frac{((\tau_1 - E_s) \times I + E_s \times I_{Ed} + \tau_2)}{\tau_1}. \quad (18)$$

Here, τ_1 and τ_2 are constant values. In practice, E_s values were quantized into 0, 16, 20, 24, 28, and 32 for hardware implementation. The values τ_1 and τ_2 were set to 32 and 16, respectively. It should be noted that τ_2 was determined as $\tau_1 \times \frac{1}{2}$ for rounding-off calculations. It is note that the quantization and parameter selections can make the equation be realized easily and fast by bit-shift operations (for example $32 \gg 5$ gives 1). The above filtering scheme can correct the corrupted large edges, especially curved ones.

EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed scheme, 512×512 sized test images that included Peppers, Lena, Barbara and Bridge were used. All the compared algorithms were implemented using MATLAB 2009b with the MS Windows XP operating system running on an Intel® Core™ i5 2.67 GHz CPU and 2 GB RAM. In this study, noise was modeled as a salt-and-pepper impulse noise, even though the proposed method is not limited to this model. Let pixels $x(i,j)$ be corrupted randomly by two fixed extreme values, 0 and 255. The probability density function for each pixel intensity value $s(i,j)$ located at (i,j) is defined by the following:

$$f(x) = \begin{cases} \frac{p}{2}, & \text{for } x = 0 \\ 1-p, & \text{for } x = s(i,j) \\ \frac{p}{2}, & \text{for } x = 255. \end{cases}$$

A set of Peppers, Lena, Barbara and Bridge images, as shown in Figure 4, was contaminated with varying noise densities from 10 to 90%. In particular, high noise images around 90% are not common in practice, but they were tested to show the robustness of the proposed algorithm.

The performance of the proposed scheme was evaluated by comparing the proposed algorithm with the existing algorithms SMF,[1] AMF,[6] IASMF,[15] LUO,[17] and SAM[18] in terms of PSNR, mean absolute error (MAE) and processing time. The PSNR is calculated as follows:

$$PSNR = 20 \cdot \log_{10}\left(\frac{MAX}{\sqrt{MSE}}\right)$$

where MAX and MSE represent the maximum pixel value (255) for an eight bit input image and the MAE, respectively. The MAE is defined by

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |f_i - y_i|.$$

Here, f_i is the prediction and y_i the true value. Different mask sizes and numbers of iterations were applied to obtain the best PSNR values for the compared algorithms. In AMF, the maximum mask size $w_{max} = 35$ was obtained as the optimal value for the best performance through various tests for the given images. In the IASMF, the maximum window size $w_{max} = 11$ was originally selected from the article. However, we set the window size to $w_{max} = 21$ for best performance.

Figure 4. Original images for the evaluation: (a) Peppers; (b) Lena; (c) Barbara; and (d) Bridge.

Table I. Parameter settings for the proposed scheme.

Maximum mask size w_{max}	21
Conducting PCA	≥60% noise density and 10% edge pixel
E_s	32
τ_1	32
τ_2	16

Table I summarizes the parameter settings, and their values are empirically assigned and fixed for all the test images.

It is worth noting that all the compared methods are non-iterative algorithms even though we carried out iterations to obtain the best performance for schemes, SMF, AMF, and LUO.[21]

Table II shows the comparative PSNR results for each algorithm. The proposed scheme outperforms SMF, AMF, IASMF, LUO, and SAM by 9.03, 3.0, 0.59, 4.81, and 1.43 dB on average, respectively. In particular, in the case of 80% corrupted images, the proposed algorithm was approximately 6.47 dB better than the other reference algorithms. Additionally, we measured averaged MAE scores for each algorithm. As expected, the proposed algorithm produced the lowest MAE values for various noise densities as shown in Table III.

To show the detail preservation performance for each algorithm, various contamination levels were applied to the image Barbara. An example is shown in Figure 5. The first row in the figure indicates the images corrupted by salt-and-pepper noise with various noise densities ranging from 10 to 90% in 20% increments, respectively. The second, third, fourth, fifth, sixth, and seventh rows show the restored images from SMF, AMF, IASMF, LUO, SAM, and the proposed algorithm, respectively, for the corresponding corrupted images. The proposed algorithm produces higher restored performance in the given images, particularly in highly corrupted ones of over 60% noise density.

Table IV shows the processing time comparison for each algorithm. All of the approaches take longer time, since the noise density increases. However, the overall processing speed of the proposed scheme is slower than the other baseline systems because it considers more steps when correcting the noise and preserving the edge and detail information as the noise density increases. In practice, the proposed scheme can be used adaptively according to the degree of noise strength of the corrupted image, and its processing time can be reduced considerably for a system that has similar noise levels for long periods because there is no need to estimate noise strength at those times. Note that the IASMF also provides good performance, but it does not preserve the edge and detail information well in highly corrupted images, since it replaces the corrupted pixels with median values that do not consider the local correlation and direction of salient edges. However, the proposed algorithm could improve the restoration performance using the local pixel correlation and edge-directed filtering for highly corrupted images. Figure 6 illustrates the geometric restoration performance of the proposed scheme for the Peppers image degraded with 90% noise. As we can see, the proposed scheme preserves edge and detail information as close as possible to the original structure.

Additionally, to see the performance in depth, Figure 7 shows graphically comparative results for the sequential 15 pixels in the Barbara that was corrupted with 90% noise density. It is easily seen that the proposed scheme can restore the original structure better than the other existing methods even though the given pixel values are highly contaminated. The main reason for quality degradation in the existing algorithms is that they simply correct the noisy candidate pixel with the median value of the given mask without any content or noise strength analysis. Nevertheless, the proposed algorithm may degrade original information by replacing the noisy pixel with an average value instead of the median value in corrupted images with low noise density. However, it was verified that the degree of degradation was small enough to be ignored.

CONCLUSION

This article proposed an edge-directed switching filter that can preserve image edge and details even in images with more than 90% noise. The proposed approach is composed of a detection step that uses the extreme values in a scalable mask and a correction step that considers the local correlation of neighboring pixels in a mask. In the correction step, we performed additional edge-directed filtering using PCA for highly corrupted images to preserve edge information. The experimental results showed that the proposed method

Figure 5. Comparable results: (a) the corrupted Barbara images by salt-and-pepper noise from 10 to 90% with increments by 20 from left to right; (b) corresponding SMF (second row); (c) AMF (third row); (d) IASMF (fourth row); (e) LUO (fifth row); (f) SAM (sixth row); and (g) the proposed approach (seventh row). As the noise density increases, the corresponding results from the compared algorithms give different observations.

Table II. Resultant comparison in terms of the PSNR (dB) for the existing and proposed algorithms.

Noise ratio (%)	Peppers						Lena					
	SMF	AMF	IASMF	LUO	SAM	Proposed	SMF	AMF	IASMF	LUO	SAM	Proposed
10	33.49	37.75	42.08	40.82	41.42	42.32	35.59	40.27	44.44	43.3	42.88	44.77
20	32.52	35.87	38.58	36.24	38.15	38.86	34.32	37.97	40.91	39.14	39.35	41.35
30	30.84	33.67	36.48	33.59	35.16	36.96	32.81	35.94	38.73	36.37	36.36	39.23
40	29.61	31.72	34.41	31.49	33.17	35.00	31.45	34.12	36.96	34.24	34.37	37.51
50	27.97	30.55	33.11	29.83	31.93	33.72	29.25	32.6	35.27	32.2	32.99	35.82
60	23.66	29.34	31.59	27.50	30.01	32.46	24.61	31.21	33.84	30.01	31.16	34.33
70	18.11	27.88	30.11	24.11	28.78	31.16	18.34	29.85	32.35	25.51	29.74	33.03
80	12.26	25.89	28.32	18.22	26.96	29.38	12.49	28.09	30.56	18.76	28.18	31.23
90	8.13	23.67	26.08	11.23	24.14	27.13	8.04	25.93	28.14	11.20	26.17	28.98
Noise ratio (%)	Barbara						Bridge					
	SMF	AMF	IASMF	LUO	SAM	Proposed	SMF	AMF	IASMF	LUO	SAM	Proposed
10	24.89	28.62	33.66	33.19	32.78	34.19	25.97	29.84	34.47	34.14	32.70	34.83
20	24.39	27.57	30.54	29.73	30.41	31.14	25.14	28.62	31.38	30.42	29.51	31.72
30	23.79	26.44	28.58	27.46	28.53	29.15	24.24	27.13	29.35	27.94	27.88	29.74
40	23.36	25.28	27.09	25.74	27.19	27.66	23.29	25.74	27.84	26.05	26.16	28.28
50	22.68	24.18	25.79	24.27	25.91	26.38	22.32	24.39	26.49	24.34	25.14	26.97
60	20.88	23.17	24.64	22.78	25.08	25.27	20.28	23.25	25.22	22.61	23.62	25.71
70	16.89	22.77	23.68	21.08	24.02	24.31	16.4	22.22	23.97	20.43	22.76	24.45
80	12.12	22.17	22.6	17.26	23.03	23.41	11.94	21.2	22.67	16.78	21.62	23.14
90	8.18	21.14	21.31	11.08	21.60	22.49	7.93	19.6	20.75	10.82	20.00	21.26

Table III. Resultant comparison in terms of the MAE for the existing and proposed algorithms.

	Peppers						Lena					
	SMF	AMF	IASMF	LUO	SAM	Proposed	SMF	AMF	IASMF	LUO	SAM	Proposed
Average value	7.515	1.941	1.365	3.717	1.532	1.321	5.270	1.561	1.305	2.608	1.377	1.010
	Barbara						Bridge					
	SMF	AMF	IASMF	LUO	SAM	Proposed	SMF	AMF	IASMF	LUO	SAM	Proposed
Average value	9.543	3.766	2.773	5.027	2.910	2.661	10.388	4.380	3.217	6.785	3.912	3.144

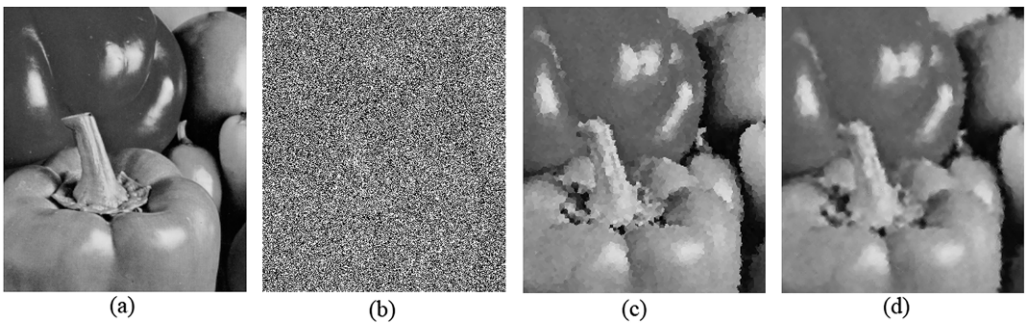


Figure 6. Comparison of preserving edges: (a) enlarged partial region of original Peppers image; (b) degraded image by 90% noise density; (c) the result of IASMF; and (d) the result of the proposed method.

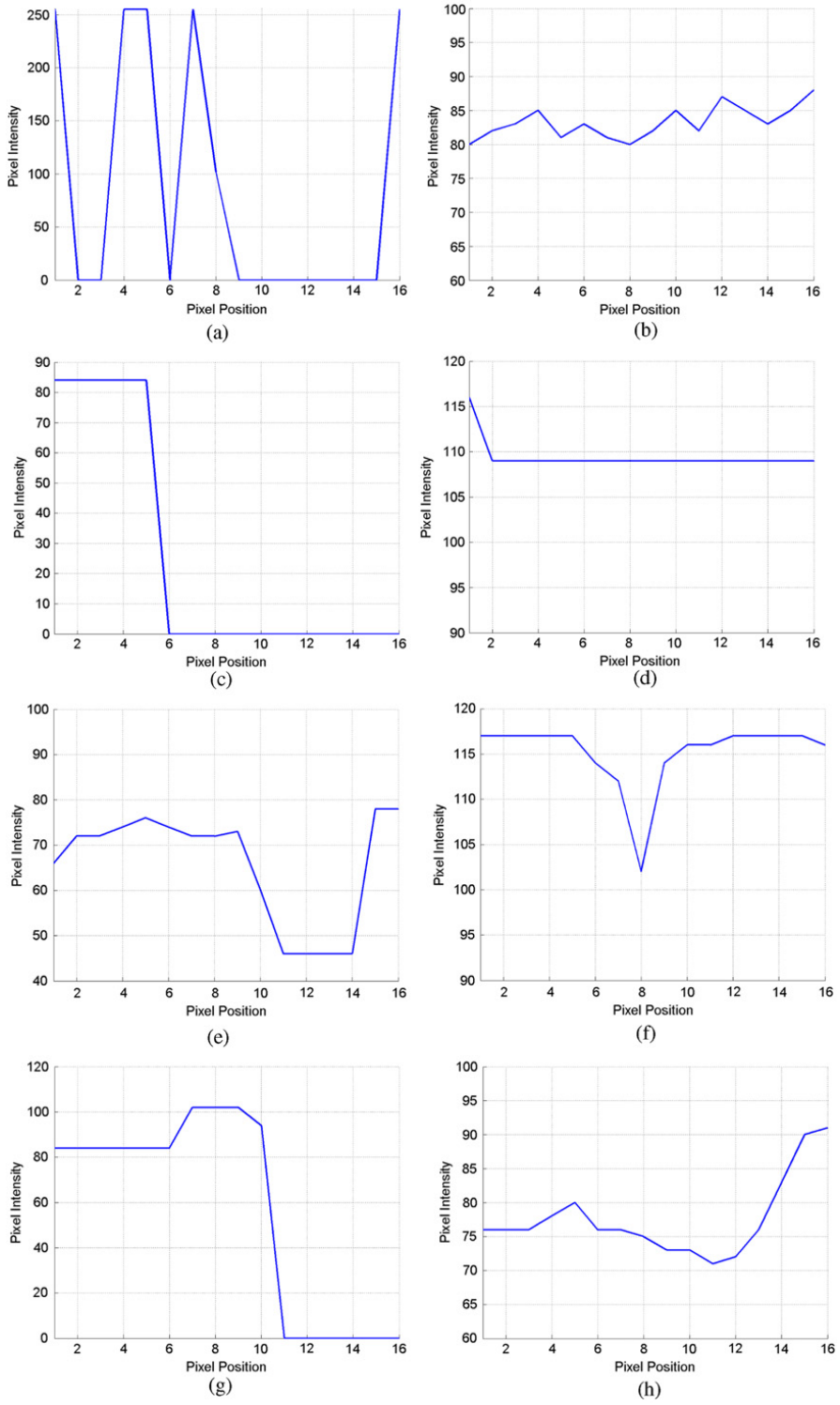


Figure 7. Comparative restoration results for existing algorithms: (a) original sequential 15 pixel values in the Barbara image. No contamination exists; (b) corresponding corrupted values by 90% salt-and-pepper noise; (c) SMF; (d) AMF; (e) IASMF; (f) LUO; (g) SAM; and (h) the proposed algorithm. It is noted that the restored outputs from compared algorithms were obtained from the corrupted image (b).

outperformed the existing approaches in terms of the PSNR over a wide range of noise densities. Particularly, image details and edges were preserved much better than the other algorithms in a highly corrupted image. Therefore, the proposed approach can be widely used as a good tool for eliminating impulse noise occurring in electronic devices.

ACKNOWLEDGMENT

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the MEST Korea government (No. 2013-029824, 2011-0003496), by Brain Korea 21+, and by SK Telecom Ltd. (No. 20121048).

Table IV. Processing time comparison (s) for the existing and proposed algorithms.

Noise density (%)	Peppers						Lena					
	SMF	AMF	IASMF	LUO	SAM	Proposed	SMF	AMF	IASMF	LUO	SAM	Proposed
10	0.170	0.429	2.322	5.366	7.167	3.507	0.134	0.661	2.320	3.693	7.161	3.531
20	0.174	0.423	2.795	5.549	7.203	5.096	0.172	0.493	2.784	5.586	7.210	5.090
30	0.226	0.514	3.250	7.401	8.160	6.793	0.201	0.508	3.228	7.368	7.572	7.030
40	0.255	0.611	3.721	9.238	8.481	8.391	0.251	0.605	3.700	9.291	8.152	8.739
50	0.367	1.722	4.117	11.114	8.199	10.126	0.346	0.725	4.152	23.343	8.063	10.394
60	0.631	2.486	4.594	23.622	8.790	15.266	0.554	1.178	4.681	23.551	8.971	15.375
70	0.684	3.882	5.114	34.393	8.864	17.260	0.722	4.160	5.119	29.178	9.504	17.910
80	0.748	6.444	5.739	52.733	9.928	19.671	0.762	9.374	5.752	57.368	10.590	20.184
90	0.769	17.294	7.078	71.634	13.936	24.265	0.811	16.399	6.876	80.071	13.500	24.474
Noise density (%)	Barbara						Bridge					
	SMF	AMF	IASMF	LUO	SAM	Proposed	SMF	AMF	IASMF	LUO	SAM	Proposed
10	0.171	0.409	2.346	5.282	7.137	3.429	0.135	1.356	2.358	5.331	7.871	3.552
20	0.173	0.504	2.780	5.430	7.210	5.028	0.176	1.400	2.776	5.436	8.021	5.071
30	0.207	0.512	3.231	7.230	7.737	6.576	0.174	1.654	3.243	5.549	7.828	6.612
40	0.352	0.519	3.657	9.126	8.090	8.175	0.233	1.708	3.669	9.190	8.167	8.187
50	0.395	0.741	4.154	10.986	8.095	9.913	0.334	1.553	4.151	11.023	8.081	9.950
60	0.522	10.298	4.656	21.377	8.988	11.842	0.479	4.843	4.608	22.610	9.174	12.007
70	0.663	15.482	5.039	27.190	8.915	19.175	0.783	7.844	5.171	20.430	9.176	14.488
80	0.728	18.830	5.669	53.798	10.354	21.882	0.793	9.542	5.850	47.841	10.292	18.164
90	0.901	27.242	6.847	63.340	14.559	26.101	0.802	22.700	6.999	59.561	14.650	24.764

REFERENCES

[1] R. C. Gonzalez and R. E. Woods, *Digital Image Processing: Third Edition* (Pearson Education, Inc., Press, 2008).

[2] J. M. Lopez-Alonso, J. Alda, and E. Bernabeu, "Principal component characterization of noise for infrared images," Appl. Opt. **41**, 320–331 (2002).

[3] R. H. Chan, C. W. Ho, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," IEEE Trans. Image Process. **14**, 1479–1485 (2005).

[4] J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering* (CRC, Boca Raton, FL, 1997).

[5] G. P. Deepti, M. V. Borker, and J. Sivaswamy, "Impulse noise removal from color images with hopfield neural network and improved vector median filter," *IEEE Sixth Indian Conf. on Computer Vision, Graphics and Image Processing* (2008), pp. 17–24.

[6] H. Hwang and R. A. Haddad, "Adaptive median filters: new algorithms and results," IEEE Trans. Image Process. **4**, 499–502 (1995).

[7] D. R. K. Brownrigg, "The weighted median filter," Commun. ACM **27**, 807–818 (1984).

[8] S. J. Ko and Y. H. Lee, "Center weighted median filters and their applications to image enhancement," IEEE Trans. Circuits Syst. **38**, 984–993 (1991).

[9] D. Zhang and Z. Wang, "Impulse noise detection and removal using fuzzy techniques," Electron. Lett. **33**, 378–379 (1997).

[10] T. Chen, K. K. Ma, and L. H. Chen, "Tri-state median filter for image denoising," IEEE Trans. Image Process. **8**, 1834–1838 (1999).

[11] Z. Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," IEEE Trans. Circuits Syst. Part II: Analog and Digital Signal Process. **46**, 78–80 (1999).

[12] H. L. Eng and K. K. Ma, "Noise adaptive soft-switching median filter," IEEE Trans. Image Process. **10**, (2001).

[13] G. Pok, J. C. Liu, and A. S. Nair, "Selective removal of impulse noise based on homogeneity level information," IEEE Trans. Image Process. **12**, 85–92 (2003).

[14] K. Nallaperumal, J. Varghese, and S. Saudia, "Adaptive threshold based switching median filter for highly corrupted images," *Proc. of CSI-IEEE First Int'l. Conf. EAIT 2006* (Elsevier, Calcutta, India, 2006), pp. 103–106.

[15] K. Nallaperumal, J. Varghese, S. Saudia, S. P. Mathew, K. Krishnaveni, and P. Kumar, "A new adaptive class of filter operators for salt & pepper impulse corrupted images," Int. J. Imaging Sci. Eng. **1**, 44–50 (2007).

[16] P. E. Ng and K. K. Ma, "A switching median filter with boundary discriminative noise detection for extremely corrupted images," IEEE Trans. Image Process. **15**, 1506–1516 (2006).

[17] W. Luo, "Efficient removal of impulse noise from digital images," IEEE Trans. Consum. Electron. **52**, 523–527 (2006).

[18] H. Ibrahim, N. S. P. Kong, and F. Ng, "Simple adaptive median filter for the removal of impulse noise from highly corrupted images," IEEE Trans. Consum. Electron. **54**, 1920–1927 (2008).

[19] K. K. V. Toh and N. A. M. Isa, "Cluster-based adaptive fuzzy switching median filter for universal impulse noise reduction," IEEE Trans. Consum. Electron. **56**, 2560–2568 (2010).

[20] M. Juneja and P. S. Sandhu, "Design and development of an improved adaptive median filtering method for impulse noise detection," Int. J. Comput. Electrical Eng. **1**, 627–630 (2009).

[21] Y. Song, Y. Han, and S. Lee, "Pixel correlation-based impulse noise reduction", *The 17th Korea–Japan Joint Workshop on Frontiers of Computer Vision* (Ulsan, Feb. 2011), pp. 1–4.

[22] C. Y. Chen, C. H. Chen, Y. L. Kuo, and C. H. Chen, "A two-pass filter for impulse noise reduction based on edge characteristics," *IEEE Conf. on Intelligent Information on Hiding and Multimedia Signal Processing 2009* (Sep. 2009), pp. 5–8.

[23] E. H. Kundra, E. M. Verma, and E. Aashima, "Filter for removal of impulse noise by using fuzzy logic," Int. J. Image Process. **3**, 195–202 (2011).

[24] B. Smolka, "Adaptive edge enhancing technique of impulsive noise removal in color digital images," *Computational Color Imaging* (Springer, Berlin, Heidelberg, 2011), pp. 60–74.

[25] K. S. Srinivasan and D. Ebenezer, "A new fast and efficient decision-based algorithm for removal of high-density impulse noises," IEEE Signal Process. Lett. **14**, 189–192 (2007).

[26] C. Harris and M. Stephens, "A combined corner and edge detector," *Proc. 4th Alvey Vision Conference* (1988), pp. 147–151.