Scientia Iranica D (2017) 24(3), 1312–1324

SCIENTIA IRANICA

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
www.scientiairanica.com

Frequency-domain-based switching median filter for the restoration of images corrupted with high-density periodic noise

J. Varghese[*]

College of Computer Science, King Khalid University, Abha, P.O. No: 394, Zip: 61411, Kingdom of Saudi Arabia.

Received 8 April 2015; received in revised form 1 February 2016; accepted 30 May 2016

KEYWORDS
Periodic noise;
Quasi-periodic noise;
Median filter;
Adaptive filter;
Image restoration;
Non-linear filter.

Abstract. The paper proposes an adaptive Frequency-Domain-based Switching Median Filter (FDSMF) for the restoration of images corrupted by periodic noise. The proposed algorithm incorporates region-growing technique to effectively identify noisy peak areas of the Fourier transformed image in a binary noise map image. The restoration phase of the algorithm replaces the corrupted frequencies with the median of uncorrupted frequencies by recursive median filter. Experimental results from different naturally and artificially corrupted images at various noise levels/types reveal that the performance of the proposed algorithm in restoring images corrupted by periodic noise is better than other competing algorithms in terms of subjective and objective metrics.

1. Introduction

Noise reduction is an important task in image processing to restore images for facilitating pattern recognition and computer vision applications [1]. Among all types of noise reduction algorithms, periodic noise reduction algorithms are of extreme importance since these noises are very common in all types of imaging processes [1]. Periodic noise corrupts digital images by superimposing repetitive patterns in the true signal content of the image [2]. These noises occur during the image acquisition phase when images are sensed in the presence of thermal/electrical/electro-mechanical interferences. Moreover, they are very common in aerial images captured by sensors fitted in vibrating environments like helicopters, aircrafts, satellites, and moving cameras [3-5]. Medical imaging processes such as X-ray, Computer Tomography (CT) scan, and Magnetic Resonance Imaging (MRI) are also affected by periodic noise due to the errors in Complementary Metal Oxide Semi-conductor Active Pixel Sensor (CMOS APS) and Charged-Coupled-Device (CCD)-based sensors [6,7]. Periodic noise also affects digital images when images are received by TV receivers with poor signal/transmission strength [8]. Many spatial domain methods are proposed in the literature to restore periodic noise corrupted images by directly applying filtering operations to the input corrupted images, including mean, median [1-3], and soft morphological filters [9-13]; but, the performance of these filters is limited since it is very difficult to differentiate between noise and uncorrupted frequencies in spatial domain.

Frequency domain techniques carry out filtering operation in the transformed domain by using Discrete Fourier Transform (DFT), Discrete Wavelet Transform (DWT), Discrete Cosine Transform (DCT), etc., and inverse transforms provide the restored image in the spatial domain [1]. The Fourier-Transform-based frequency domain techniques [14-23] show much improved performance in denoising periodic noise since noisy frequencies are clearly differentiated in Fourier

[*]. *Tel.: 00966172418525; Fax: 00966172418185*
E-mail address: wrkus@kku.edu.sa

transformed domain by being made look like star/spike shaped peak areas [1-3]. The basic filtering techniques applied in frequency domain to attenuate periodic noise include Weiner filter, Ideal and Butterworth low-pass filters, band reject filters, and notch filter [1]. Ideal and Butterworth low-pass filters admit only the low frequencies clustered around the centre of low frequency in the Fourier transformed image and, thereby, cannot maintain image fidelity in the restored image, since these algorithms discard all high frequencies irrespective of their corruption status. Specific circular region around the centre of low frequency in Fourier domain is completely rejected by the band reject filters to dispose noisy frequencies, while regions centred at specific frequency are rejected by the notch filter [1]. The notch filters have advantage over band reject filters since they discard only small portions of the band. Among all notch filters, the Gaussian notch filter plays an important role in denoising of periodic noise corrupted images since it not only discards the corrupted centre frequency peak but also suppresses its neighbouring frequencies corresponding to the noisy peak areas. This advantage of Gaussian notch filters motivated later algorithms [19-22] to incorporate Gaussian notch-based correction algorithms in their filtering operation. Aizenberg and Butakoff proposed three filters by incorporating noisy peak identification and correction phases to restore images [17-19]. The Gaussian notch-based restoration filter proposed by Aizenberg and Butakoff [19] identifies a frequency in the Fourier domain to be corrupted when the ratio of the frequency value to the median frequency value from the prefixed neighbourhood exceeds a static threshold. The filter diffuses these corrupted frequencies by applying Gaussian notch filter with prefixed window size. Many variations of this filter have been developed by attempting at different peak detection and Gaussian notch filtering schemes [8,20,21], but these algorithms fail to effectively restore periodic noise due to the following difficulties:

1. Gaussian notch filters reject the noisy peak frequency without restoring it by uncorrupted neighbouring frequencies;
2. It is very difficult to make the filtering window size adaptive to varying noise levels;
3. The coefficients of Gaussian notch filter are designed in accordance with the distance without making them adaptive to the noisy frequency values; hence, these coefficients are not precise enough to diffuse the noisy peak areas.

Hudhud and Turner [22] proposed a semi-automated algorithm that manually identified the noisy peak positions by replacing the noisy frequencies with median of the neighbouring uncorrupted frequencies. Konstantinidis et al. [23] proposed an interpolation-based filter to restore images by using prefixed cross shaped window, but this filter could not perform effective restoration since the window size was not adaptive to the varying noise strength. The peak detection procedure employed by Grédiac, et al. [24], and Sur and Grédiac [25] algorithms uses static approximation of power law transformation, but it is not adaptive to the corrupting noise and image types.

The paper presents an adaptive Frequency-Domain-based Switching Median Filter (FDSMF) operator for the restoration of periodic noise corrupted images by incorporating region-growing-based noise detection and median-based noise correction phases. The paper is organized in four sections. Section 2 describes the proposed restoration filter. Section 3 provides the experimental result analysis, and conclusions are finally made in Section 4.

2. Proposed FDSMF algorithm

The proposed frequency-domain-based Switching Median Fillter (FDSMF) algorithm restores images corrupted with periodic noise by incorporating the distinct stages of noisy frequency identification and correction. The input corrupted image is Fourier transformed and its origin is shifted to the centre for isolating noisy frequencies. The origin shifted Fourier transformed image F of the input corrupted image A of size $M \times N$ is determined as:

$$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(-1)^{x+y}A(x,y)e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}. \tag{1}$$

Here, $u = 0, 1, 2, \cdots, M-1$, $v = 0, 1, 2, \cdots, N-1$, and $(-1)^{x+y}$ represent shifting operation. The algorithm uses '*fft2*' and '*fftshift*' functions of Matlab software to perform Eq. (1). The noise identification stage of the proposed algorithm detects the corrupted frequency positions in a binary flag image by finding the noisy peak areas in the origin shifted Fourier transformed image F. The noise correction phase of the proposed algorithm replaces the detected noisy frequencies in F by a valid restorer to diffuse the noise. Figure 1(b) shows the Fourier transformed image of Figure 1(a).

It can be observed that the noisy frequencies spreading throughout the corrupted image in spatial domain (Figure 1(a)) are concentrated in the Fourier transformed image, causing the star/spike-like appearance outside the low frequency region. The motivation and goal of the proposed algorithm is to effectively identify and restore these corrupted peaks in frequency domain.

2.1. The noise detection phase

The noise detection phase of the proposed algorithm records the noisy frequency positions in a binary noise

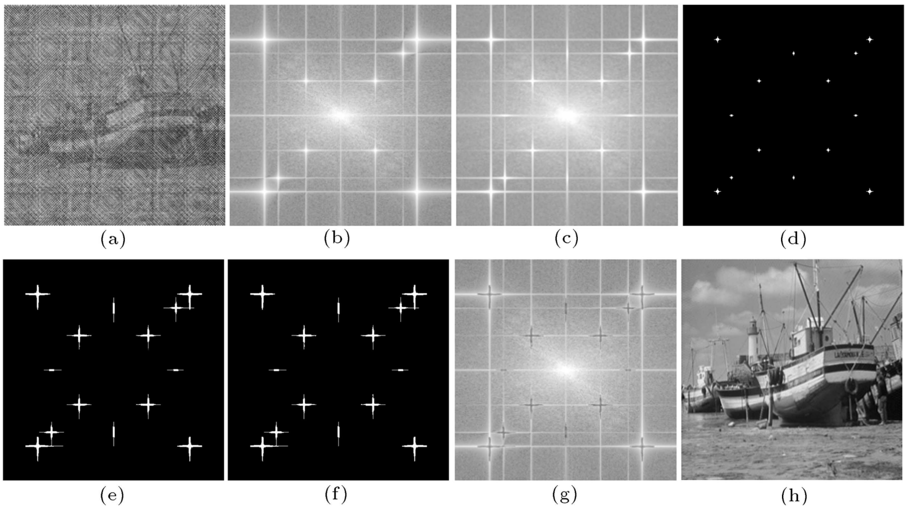

Figure 1. Different stages of FDSMF: (a) Corrupted Image, (b) Fourier transformed image, (c) difference image, (d) seed points identified, (e) initial noise map image, (f) final noise map image, (g) restored frequency domain image, and (h) final restored image.

map image f of the same size $M \times N$ as the input corrupted image A. The aim of the noise detection algorithm is to set '1' at the positions of the binary noise map image f, corresponding to the position of corrupted frequencies in the Fourier transformed image F. As an initial step, the algorithm resets all positions of binary noise map image f to '0', assuming that all the frequencies in F are uncorrupted; hence:

$$f=\{f(i_1,i_2)=0 : 0 \leq i_1 \leq M-1, 0 \leq i_2 \leq N-1\}. \tag{2}$$

The noise detection phase of the proposed algorithm starts by isolating the noisy frequency areas from the uncorrupted ones by finding the average difference of each pixel from its neighbors to better identify the noisy peak areas. The iterative peak detection stage of the noise detection algorithm finds the largest frequency peak outside the predefined low-frequency region. If the maximum peak value of the iteration is greater than the determined threshold, region-growing algorithm is applied to the frequency difference image by taking the largest peak point as the seed point. The positions of the corrupted areas identified by the region-growing algorithm are recorded in the binary flag image to indicate the noisy positions and, accordingly, the recorded peak areas of the difference frequency image are diffused to zero in order to prevent these peak positions from participating in the subsequent iterations to find the remaining noisy peaks. The iterations of the algorithm are continued until the current iteration of the algorithm cannot detect any new noisy peak areas. The algorithm is explained through the following steps:

- **Step i.** In order to highlight the noisy frequency peaks and to suppress other uncorrupted areas of the frequency domain image to make the iterative peak detection process easy, the algorithm finds the average difference image D as:

$$D(u,v)=\frac{1}{(W_1)^2}\sum_{k=-K_1}^{K_1}\sum_{l=-K_1}^{K_1}|F(u,v)-F(u+k,v+l)|. \tag{3}$$

Here, W_1 is the window size for finding the average difference and is equal to $(2K_1+1)$, where K_1 is a positive integer not less than 1. The value of $D(u,v)$ is the average difference of frequency value $F(u,v)$ from the neighbouring values. The difference image generated from Figure 1(b) is shown in Figure 1(c). The peak detection process of the proposed algorithm is iterative and it produces two sequences of binary noise map and frequency difference images in all iterations. The binary noise map image sequence is $f^0, f^1, f^2, \cdots$, where $f^0 = f$ is the binary noise map image in the 0th iteration and the frequency difference image sequence is $D^0, D^1, D^2, \cdots$, where $D^0 = D$ is the frequency difference image in the 0th iteration;

- **Step ii.** The algorithm identifies the position (i_1, i_2) corresponding to the maximum peak value among all the frequency difference values of the average difference image D^n of the nth iteration by avoiding the low frequency region RLF; it is determined by:

$$(i_1,i_2)=\underset{(k,l)}{\operatorname{argmax}}\{D^n(k,l):(k,l)\in\{\mathrm{Dom}(F)\backslash RLF\}\}, \tag{4}$$

where '\' is the set difference operator and the region of low frequency RLF is the set of low frequency positions defined by:

$$RLF=\{(k,l):d(k,l)\leq T_1\forall(k,l) \in \mathrm{Dom}(F)\}. \tag{5}$$

Here, T_1 is a threshold that differentiates between low- and high-frequency areas of frequency domain image and the domain of F; also, $\mathrm{Dom}(F)$ is the set of spatial positions of F defined by:

$$\mathrm{Dom}(F) = \{(k_1, l_1) : 0 \leq k_1 < M, 0 \leq l_1 < N\}. \tag{6}$$

Also, $d(k, l)$ is the Euclidian distance between the pixel positions (k, l) and the centre of the image, $(\frac{M}{2}, \frac{N}{2})$;

- **Step iii.** If the maximum peak value, $D^n(i_1, i_2)$, corresponding to the position (i_1, i_2) detected in the nth iteration is greater than a predefined threshold T_2, it is detected as a noisy peak. Accordingly, the peak position (i_1, i_2) is fed into region-growing algorithm as a seed point for identifying and recording of the peak areas associated with this peak in the noise map image of the nth iteration, f^n. If $\Omega^W(i_1, i_2)$ generalizes the set of pixel positions defined by neighborhood $W \times W$ centred at position (i_1, i_2), $\Omega^W(i_1, i_2)$ is mathematically expressed as:

$$\Omega^W(i_1, i_2) = \left\{ \begin{array}{r} (j_1, j_2) : i_1 - K \leq j_1 \leq i_1 + K, i_2 \\ - K \leq j_2 \leq i_2 + K \end{array} \right\}. \tag{7}$$

Here, $K = (W - 1)/2$, and W is an odd integer not less than 3. If $W_2 \times W_2$ denotes the window size used for the region-growing process, the region-growing algorithm is detailed through the following sub-steps;

- **Step iii.a.** The window $W_2 \times W_2$ and the maximum window size, $W_{\max} \times W_{\max}$, used in the region-growing algorithm are respectively initialized by 3×3 and 15×15;
- **Step iii.b.** The corresponding frequency positions of the peak area associated with the peak frequency value, $D^n(i_1, i_2)$, in the difference image, D^n, are identified in the set S by checking the absolute difference between the peak frequency value and the difference image value whose positions are defined by the set $\Omega^{W_2}(i_1, i_2)$ as:

$$S = \left\{ \begin{array}{c} (j_1, j_2) : (j_1, j_2) \in \Omega^{W_2}(i_1, i_2), \\ f^n(j_1, j_2) = 0 \quad \text{and} \\ |D^n(j_1, j_2) - D^n(i_1, i_2)| \leq T_3 \end{array} \right\}. \tag{8}$$

Here, T_3 is a threshold used for detecting the noisy peak areas. It is to be noted that the noisy peak positions that are not yet recorded in the flag image are collected in the set S;

- **Step iii.c.** If the set S is found nonempty, the binary map values of the noise map image in the $n + 1$th iteration, f^{n+1}, corresponding to the pixel positions caught in the set S are marked by '1' since these positions are detected as noisy frequency areas; hence:

$$f^{n+1}(j_1, j_2) = \begin{cases} 1 & \text{if } (j_1, j_2) \in S \\ f^n(j_1, j_2) & \text{Otherwise} \end{cases} \tag{9}$$

Accordingly, the corresponding difference values of $D^{n+1}(j_1, j_2)$ in the $n+1$th iteration corresponding to these noisy areas are diffused to '0' in order to prevent these areas from participating in the subsequent iterations while detecting new noisy peak positions, i.e.:

$$D^{n+1}(j_1, j_2) = \begin{cases} 0 & \text{if } (j_1, j_2) \in S \\ D^n(j_1, j_2) & \text{Otherwise} \end{cases} \tag{10}$$

The region-growing process is continued with the increased window size, $W_2 = W_2 + 2$, from Step iii.b if $W_2 < W_{\max}$;

- **Step iii.d.** Otherwise, if S is found empty, the iterative peak detection process is continued to the $n + 1$th iteration from Step ii since the region-growing algorithm in the iteration n stops adding new pixel positions in the noise position set S. Unlike normal region-growing algorithms that use average of already grouped pixels to check the similarity, the proposed region-growing algorithm uses the centre peak value to check the similarity and, thereby, the positions for adding noisy frequency to the noise map image.

- **Step iv.** If the maximum peak value, $D^n(i_1, i_2)$, detected in Step iii is found less than the predefined threshold, T_1, the algorithm stops the iterative noise map identification process since all the noisy areas in the difference image are caught in the noise map image, f^n. The detected peaks from the difference image (Figure 1(c)) are shown in Figure 1(d). The noise map generated from Figure 1(d) using the region-growing algorithm is provided in Figure 1(e);
- **Step v.** The binary noise map image generated by the proposed noise detection process normally contains breaks and gaps, and these gaps and breaks are filled by applying morphological closing operation to perform better identification of noisy areas in the frequency domain. The final noise map image f is given by:

$$f = f^n \bullet b. \tag{11}$$

Here, '$\bullet$' and b are the morphological closing operation and the structural element, respectively. The algorithm uses a flat 3×3 square structural element to perform morphological closing operation. The final noise map image determined from Figure 1(e) by applying the morphological closing operation is provided in Figure 1(f). Once the final noise map image f is generated, it is fed into the restoration algorithm for restoring the corrupted frequencies. The restoration process is explained in the following subsection.

2.2. *The noise correction phase*

The filtering algorithm of the proposed scheme is a variation of standard recursive median filter [1] and it restores the detected periodic noise corrupted frequencies in F by the median of uncorrupted frequencies from the static neighborhood. The algorithm uses the corrupted/purity status of the frequencies in the flag image f provided by the noise detection algorithm. The noise detection algorithm sets '1' at all positions of the binary noise map image f corresponding to the position of corrupted frequencies in the Fourier transformed image F. The median filter used in the filtering stage is recursive since it uses the previously restored frequencies to restore the current corrupted frequency. The proposed filtering scheme restores all corrupted frequencies in corrupted frequency image F for all positions $(j_1, j_2) \in \mathrm{Dom}(F)$ starting from position $(0, 0)$ and is tracked through the following steps. $\mathrm{Dom}(F)$ is defined in Eq. (6).

- **Step i.** If the binary noise map value, $f(j_1, j_2)$, corresponding to the frequency image value $F(j_1, j_2)$ at position (j_1, j_2) is '0', $F(j_1, j_2)$ is retained in the restored frequency image since $F(j_1, j_2)$ is uncorrupted and the algorithm is continued from Step iii;
- **Step ii.** Otherwise, if the binary noise map value, $f(j_1, j_2)$, corresponding to the frequency image value, $F(j_1, j_2)$, at position (j_1, j_2) is '1', the output frequency image value, $F(j_1, j_2)$, is restored with the median of uncorrupted frequencies of the frequency image F whose positions are defined by $\Omega^{W_3}(j_1, j_2)$ since $F(j_1, j_2)$ is a corrupted frequency. The proposed restoration algorithm is recursive and it re-uses some of its previous outputs as inputs to restore other pixels which are not yet restored. The algorithm moves from left to right and from top to bottom to replace the corrupted frequencies. For example, if $W_3 = 3$, the algorithm replaces $F(j_1, j_2)$ with the median of uncorrupted frequencies covered by $\Omega^{W_3}(j_1, j_2)$, among which $F(j_1-1, j_2-1)$, $F(j_1 - 1, j_2)$, $F(j_1 - 1, j_2 + 1)$, and $F(j_1, j_2 - 1)$ have already been restored. Hence, irrespective of the corrupted status of frequency-domain image, the proposed algorithm ensures the presence of sufficient uncorrupted pixels in the filtering window while replacing a corrupted pixel. Unlike iterative filtering, the computational complexity of recursive filtering will also be same as that of normal filtering. The output and input frequency images are given with the same notation, F, to indicate that the filter is recursive. The operation is mathematically expressed as:

$$F(j_1, j_2) = \mathrm{Median}(\psi). \tag{12}$$

Here, ψ is the set of uncorrupted frequencies of F defined by the positions $\Omega_j^{W_3}$ as:

$$\psi = \left\{ \begin{array}{l} F(l_1, l_2) : f(l_1, l_2) = 0 \quad \text{and} \\ \qquad\qquad (l_1, l_2) \in \Omega^{W_3}(j_1, j_2) \end{array} \right\}. \tag{13}$$

Accordingly, the algorithm resets the binary map value, $f(j_1, j_2)$, to zero since it replaces a corrupted frequency. This replacement of flag value enables the newly restored frequencies to participate while finding the restorer for replacing other neighboring frequencies;

- **Step iii.** The restoration process of the next pixel is continued from Step i and the restoration process is continued until all the frequencies in F are processed. The final restored frequency image from Figure 1(b) by the proposed algorithm is shown in Figure 1(g). It is to be noted that the corrupted frequency peak areas in Figure 1(b) are well patched by the proposed algorithm to produce Figure 1(g);
- **Step iv.** Once all the corrupted frequencies of the frequency image F are restored by replacing the corrupted frequencies with the recursive median of uncorrupted neighboring frequencies, the algorithm performs the inverses of centre shifting and Fourier transform to reconstruct the final restored image O as:

$$O(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} (-1)^{x+y} F(u, v) e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)}. \tag{14}$$

Here, $(-1)^{x+y}$ represents the shifting operation and the algorithm uses '*ifft2*' and '*ifftshift*' functions of Matlab software to perform Eq. (14). Since the algorithm effectively detects and restores the noisy frequency areas to generate the final restored image, it always has the ability to produce better restoration of images corrupted with high-density periodic noise as can be viewed from Figure 1(h).

3. Experimental results and analysis

The performance of the proposed FDSMF algorithm, while restoring artificially corrupted images is tested over 20 different 8-bit images, including Cameraman, Boats, Lena, Bridge, Man, and Barbara, of which the Boats, Lena, Barbara, and Cameraman images are used in this section for performance evaluation. The comparative filters used in the analysis are Standard Morphological Filter (SMF1)[1], Soft Morphological Filter (SMF2) [9], Ideal Low-Pass Filter (ILPF) [1], Aizenburg Mean Filter (AMF1) [17], Aizenberg Median Filter (AMF2) [18], Windowed Gaussian Notch Filter (WGNF) [19], Gaussian Star Shaped Filter (GSSF) [20], Adaptive Optimum Notch Filter (AONF) [8], Interpolation Notch Reject Filter (INRF) [23], Brickwall Notch Reject Filter (BNRF) [23], and Gaussian Notch Reject Filter (GNRF) [23]. Peak Signal to Noise Ratio (PSNR) and Mean Absolute Error (MAE) are the criteria used in this paper to analyze the objective performance of the algorithms. The formulations for PSNR and MAE are defined in [26]. The visual detail preservation capability of different algorithms is assessed with Mean Structural Similarity Index Measure (MSSIM) while the computational efficiency of the algorithms is analyzed with Computational Time (CT) in seconds.

An Intel Core 2 Duo system with 2.6 GHz and 4 GB RAM is used for testing the algorithms. The Structural Similarity Index Measure (SSIM) [27] accounts for visual quality assessment similar to human visual system and is determined for each pixel defined by a window of an image. If Z_b and O_b are the image samples of the same structure from the bth window respectively from original uncorrupted image Z and restored image O, the formulation of SSIM is given by:

$$\mathrm{SSIM}(Z_b, O_b) = \frac{(2\mu_{Z_b}\mu_{O_b}+c_1)\,(2\sigma_{Z_bO_b}+c_2)}{(\mu^2_{Z_b}+\mu^2_{O_b}+c_1)\,(\sigma^2_{Z_b}+\sigma^2_{O_b}+c_2)}, \tag{15}$$

where μ_{Z_b}, μ_{O_b}, σ_{Z_b}, σ_{O_b}, and $\sigma_{Z_bO_b}$ are respectively the mean of original uncorrupted image, mean of restored image, standard deviation of original image, standard deviation of restored image, and joint standard deviation of original and restored images defined by the window size. Here, c_1 and c_2 are the constants to avoid zero denominators. Mean Structural Similarity Index Measure (MSSIM) is the average of all calculated SSIMs and is defined by:

$$\mathrm{MSSIM} = \frac{1}{m}\sum_{b=1}^{m}\mathrm{SSIM}(Z_b, O_b). \tag{16}$$

Here, m is the number of local windows used for determining SSIM. MSSIM is equal to 1 when both the images are identical. To artificially generate periodic noise for testing the performance of algorithms, the algorithm uses a linear combination of sinusoid functions with different amplitude and phase angles similar to the noise function used in [8,20]. These sinusoid functions are generated with the idea of Fourier series. In addition, these artificially generated periodic noise images are superimposed in the uncorrupted image through addition. The periodic noise images considered in this paper are:

$$N_1(x,y) = a \times 255(\sin(x+y)), \tag{17}$$

$$N_2(x,y) = a \times 255\begin{pmatrix}\sin(8y)+\sin(8x)\\ \quad +\sin(5.25x+5.25y)\\ +\sin(x+5.25y)\end{pmatrix}, \tag{18}$$

$$N_3(x,y) = a \times 255\begin{pmatrix}\sin(1.8x+1.8y)+\sin(x+y)\\ +\sin(2.2x+2.2y)\\ +\sin(1.8x-1.8y)\\ +\sin(x-y)\\ +\sin(2.2x-2.2y)\end{pmatrix}. \tag{19}$$

Here, a is the strength of the noise and (x,y) is the spatial position.

Figures 2 to 4 show the visual analysis of the restored outputs produced by ILPF, GSSF, WGNF, AONF, INRF, and the proposed FDSMF Filter, respectively, for images corrupted with N_1, N_2, and N_3 noises with the noise strength of $a = 1$. The restored outputs of different algorithms from corrupted Barbara image by $N_1 + N_2 + N_3$ with $a = 1$ for all types of noises are shown in Figure 5. In order to test the performance of the proposed algorithm in restoring real periodic/quasi periodic noise corrupted images, the algorithm is tested with 15 different non-synthetic/naturally corrupted images, of which Clown, Ariel Pompeii [28], Halftone Lady [29], Man, and football live match video frame [8] images are used in this paper for analysis.

Figure 6 shows the restored outputs of different algorithms for Clown image. Figure 7 produces the restored outputs of non-synthetic, naturally corrupted Ariel Pompeii, Halftone Lady, Man, and football live match video frame images. The visual analysis of restored outputs of different algorithms performed in Figures 2 to 7 shows that the restored outputs produced by the proposed algorithm from natural and artificially corrupted images are better than those produced by the competing algorithms.

Figures 2 to 4 also show the corrupted images and their corresponding Fourier transforms to provide a clear idea about the corrupting noise. Tables 1 to 3 show MAE, PSNR, MSSIM, and Computation Time (CT) analyses of the restored outputs produced by SMF1, SMF2, ILPF, AMF1, AMF2, WGNF, GSSF, AONF, INRF, BNRF, GNRF, and the proposed FDSMF filter, respectively, for Barbara, Cameraman, and Lena images corrupted with N_1, N_2, and N_3 noises with various noise strengths. The MAE, PSNR, MSSIM, and the Computation Time (CT) of different algorithms when restoring Barbara images corrupted

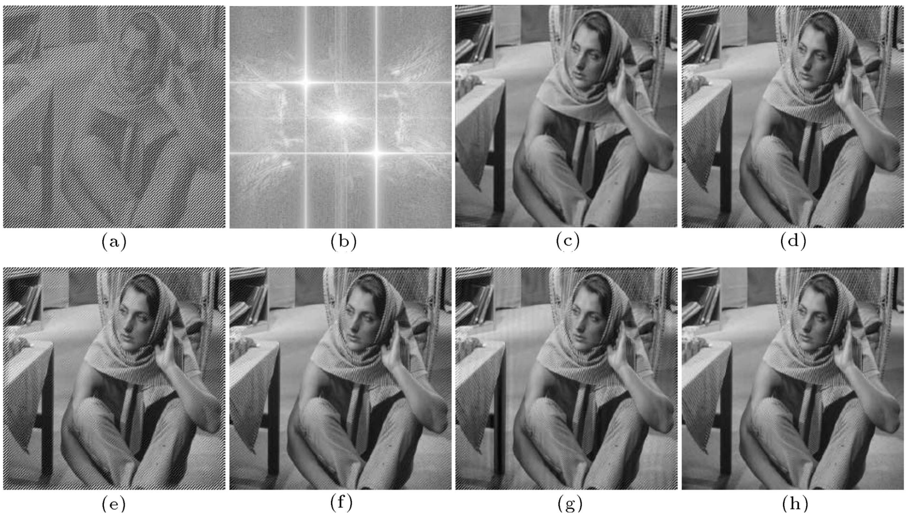

Figure 2. Outputs of algorithms in restoring the Barbara image corrupted with N_1-type noise of strength $a = 1.0$: (a) Corrupted image, (b) Fourier transform of the corrupted image, (c) ILPF, (d) GSSF, (e) WGNF, (f) AONF, (g) INRF, and (h) FDSMF.

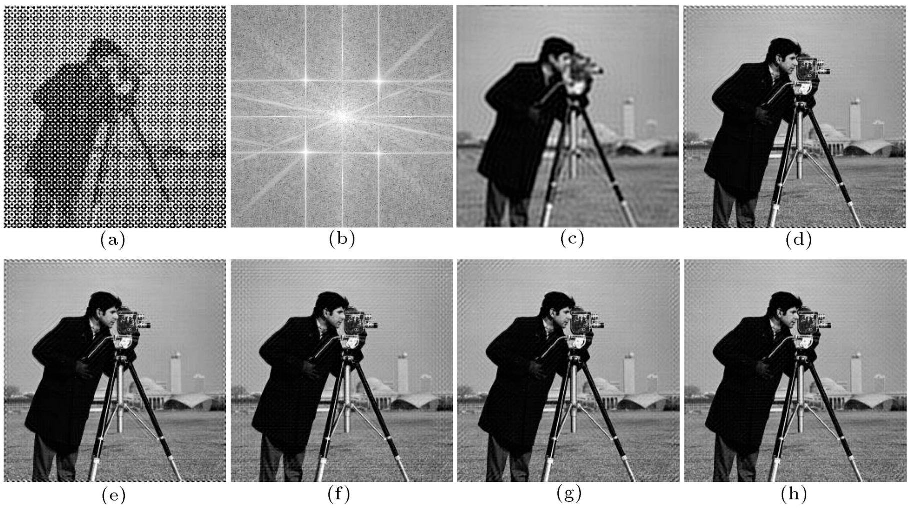

Figure 3. Outputs of algorithms in restoring the Cameraman image corrupted with N_2-type noise of strength $a = 1.0$: (a) Corrupted image, (b) Fourier transform of the corrupted image, (c) ILPF, (d) GSSF, (e) WGNF, (f) AONF, (g) INRF, and (h) FDSMF.

Table 1. MAE, PSNR, MSSIM, and CT analysis of different filters while restoring Barbara images corrupted with N_1 noise.

Filters	Noise strength $a = 0.5$				Noise strength $a = 0.9$				Noise strength $a = 1.5$			
	MAE	PSNR	MSSIM	CT	MAE	PSNR	MSSIM	CT	MAE	PSNR	MSSIM	CT
SMF1	16.37	22.38	0.78	0.85	18.79	22.14	0.75	0.81	23.32	22.06	0.67	0.84
SMF2	10.32	23.27	0.81	10.34	10.79	23.14	0.79	10.30	11.89	22.84	0.74	10.28
ILPF	9.70	23.99	0.85	16.73	10.63	23.15	0.84	16.71	12.19	21.52	0.82	16.72
AMF1	8.03	24.09	0.87	27.82	9.44	23.99	0.85	27.72	13.07	20.55	0.76	27.44
AMF2	7.82	25.00	0.89	22.40	9.21	24.10	0.87	22.62	12.13	21.74	0.82	22.35
WGNF	7.41	25.40	0.91	22.38	9.01	24.30	0.89	22.29	11.98	21.86	0.85	22.19
GSSF	3.87	26.33	0.95	2.93	5.68	22.43	0.92	4.07	7.43	23.33	0.91	5.65
AONF	2.33	34.11	0.95	4.31	3.21	33.69	0.93	9.87	4.38	30.96	0.91	12.07
INRF	2.52	34.64	0.96	6.48	3.62	32.73	0.93	6.53	5.97	30.91	0.92	6.47
BNRF	2.87	34.28	0.95	5.07	3.78	32.64	0.91	5.13	6.08	30.86	0.90	5.12
GNRF	1.92	36.56	0.97	5.00	2.30	34.42	0.94	5.99	5.38	32.31	0.92	5.83
FDSMF	**1.23**	**4 1.79**	**0.98**	**4.15**	**1.41**	**40.01**	**0.96**	**5.41**	**1.82**	**38.16**	**0.97**	**5.57**

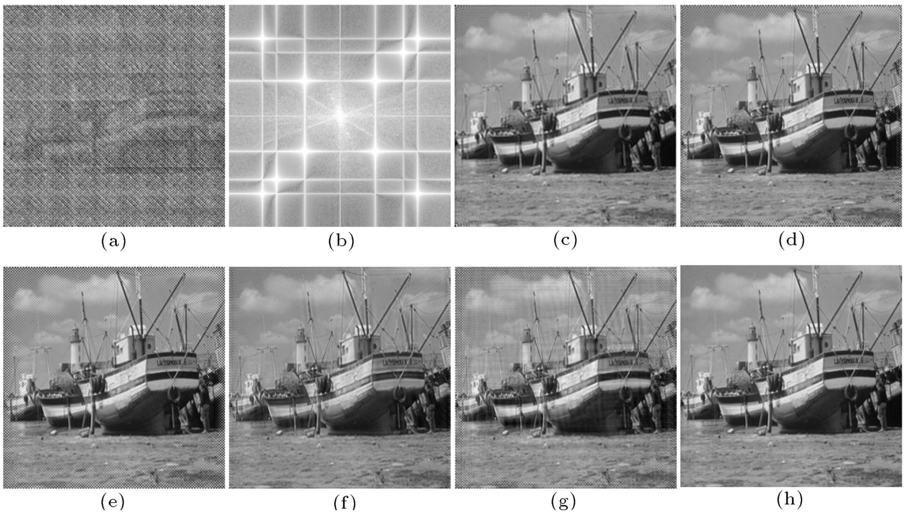

Figure 4. Outputs of algorithms in restoring the Lena image corrupted with N_3-type noise of strength $a = 1.0$: (a) Corrupted image, (b) Fourier transform of the corrupted image, (c) ILPF, (d) GSSF, (e) WGNF, (f) AONF, (g) INRF, and (h) FDSMF.

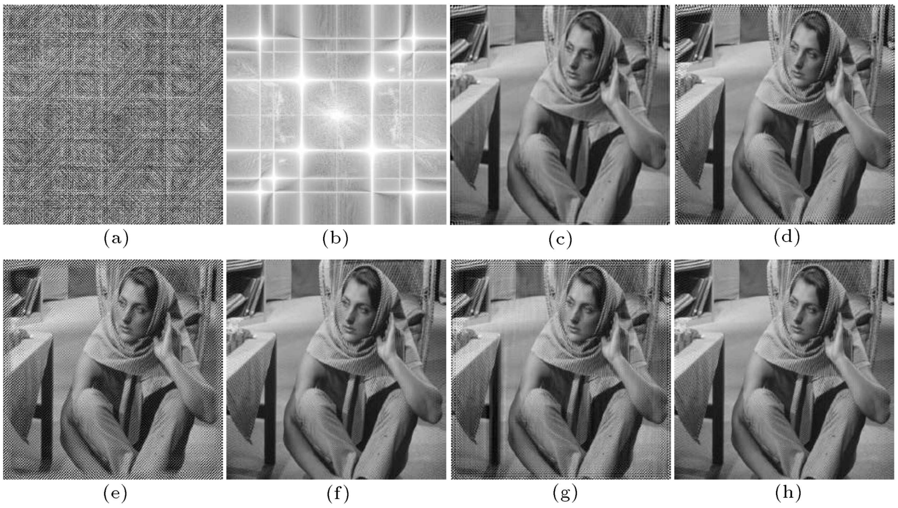

Figure 5. Outputs of algorithms in restoring the Barbara image corrupted with $N_1 + N_2 + N_3$-type noises each with strength $a = 1$: (a) Corrupted image, (b) Fourier transform of the corrupted image, (c) ILPF, (d) GSSF, (e) WGNF, (f) AONF, (g) INRF, and (h) FDSMF.

Table 2. MAE, PSNR, MSSIM, and CT analyses of different filters when restoring Cameraman images corrupted with N_2 noise.

Filters	**Noise strength $a = 0.5$**				**Noise strength $a = 0.9$**				**Noise strength $a = 1.5$**			
	MAE	**PSNR**	**MSSIM**	**CT**	**MAE**	**PSNR**	**MSSIM**	**CT**	**MAE**	**PSNR**	**MSSIM**	**CT**
SMF1	18.89	19.59	0.67	0.23	20.09	16.93	0.63	0.26	28.83	14.58	0.51	0.25
SMF2	17.25	19.81	0.71	2.82	16.80	17.96	0.64	2.77	30.32	15.60	0.52	2.77
ILPF	8.76	24.41	0.76	2.14	9.78	23.69	0.73	2.16	11.58	22.25	0.68	2.17
AMF1	6.87	28.68	0.77	7.12	8.72	26.23	0.70	7.07	11.80	23.39	0.60	7.08
AMF2	6.29	28.59	0.79	5.70	9.55	24.95	0.76	5.66	14.59	23.69	0.71	5.59
WGNF	5.07	29.95	0.89	5.78	6.01	28.01	0.86	5.78	7.45	26.16	0.82	5.85
GSSF	4.78	28.25	0.92	0.95	6.74	24.08	0.88	1.19	9.01	20.74	0.85	1.61
AONF	8.56	27.54	0.85	5.73	12.69	22.57	0.81	9.71	19.26	19.15	0.76	17.05
INRF	5.13	31.54	0.80	2.38	8.43	27.06	0.64	2.70	19.39	18.78	0.43	3.02
BNRF	4.11	33.38	0.86	2.61	6.36	29.28	0.74	3.29	17.61	19.04	0.48	3.31
GNRF	2.43	36.86	0.95	2.41	3.24	34.40	0.92	2.40	7.82	26.24	0.84	2.67
FDSMF	**2.30**	**39.28**	**0.96**	**2.68**	**3.16**	**36.46**	**0.93**	**4.25**	**6.48**	**29.01**	**0.89**	**3.05**

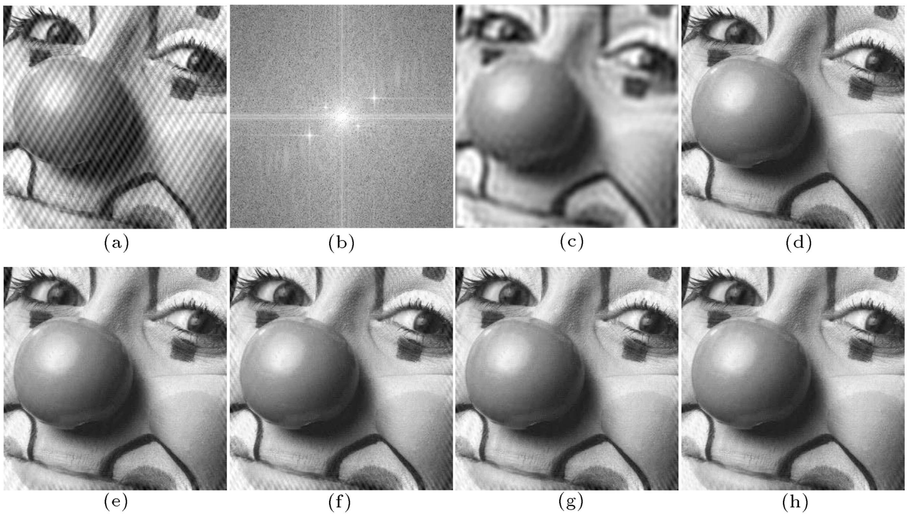

Figure 6. Outputs of algorithms in restoring non-synthetic clown image: (a) Real periodic noisy image, (b) Fourier transform of the corrupted image, (c) ILPF, (d) GSSF, (e) WGNF, (f) AONF, (g) INRF, and (h) FDSMF.

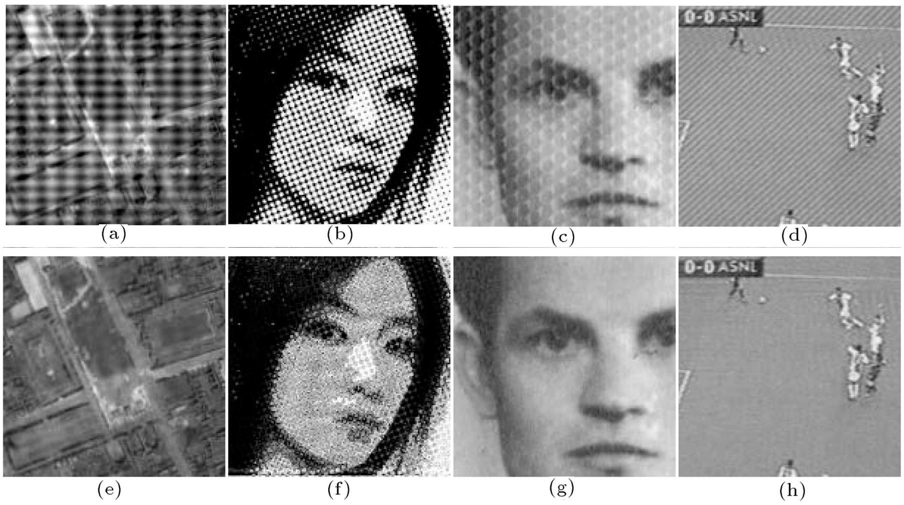

Figure 7. Performance of FDSMF in restoring naturally corrupted images: (a) Corrupted satellite image, (b) halftone newspaper image, (c) corrupted scanned photograph, (d) corrupted football live match video frame, and its corresponding restored images in (e)-(h).

Table 3. MAE, PSNR, MSSIM, and CT analyses of different filters when restoring Lena images corrupted with N_3 noise.

Filters	Noise strength $a = 0.5$				Noise strength $a = 0.9$				Noise strength $a = 1.5$			
	MAE	**PSNR**	**MSSIM**	**CT**	**MAE**	**PSNR**	**MSSIM**	**CT**	**MAE**	**PSNR**	**MSSIM**	**CT**
SMF1	18.88	14.71	0.53	0.86	24.32	10.46	0.45	0.87	28.61	8.77	0.32	0.81
SMF2	16.71	17.51	0.60	10.01	22.62	12.63	0.51	10.17	25.85	11.20	0.41	10.22
ILPF	5.72	28.04	0.94	17.07	7.29	24.67	0.91	16.80	9.82	20.89	0.86	16.77
AMF1	15.51	19.42	0.67	27.25	17.80	16.33	0.59	27.16	21.31	15.90	0.51	27.05
AMF2	14.30	20.03	0.79	22.43	16.07	18.83	0.68	22.26	19.07	17.38	0.55	21.84
WGNF	9.00	24.24	0.84	22.88	12.85	22.23	0.73	22.84	15.07	19.82	0.60	22.82
GSSF	7.82	25.71	0.93	7.01	11.75	24.70	0.89	11.79	13.55	20.82	0.84	20.23
AONF	6.82	25.97	0.94	16.20	7.62	24.87	0.91	27.36	13.79	20.61	0.76	45.17
INRF	6.80	27.18	0.90	6.73	13.25	24.37	0.87	6.84	17.66	20.02	0.74	6.91
BNRF	6.40	28.53	0.92	4.40	10.49	24.83	0.83	4.59	16.83	20.52	0.79	4.72
GNRF	3.58	33.60	0.94	4.23	5.32	29.58	0.89	4.22	8.12	25.52	0.82	4.18
FDSMF	**2.76**	**36.42**	**0.98**	**4.29**	**4.26**	**32.67**	**0.95**	**4.34**	**6.31**	**29.16**	**0.91**	**4.53**

Table 4. MAE, PSNR, MSSIM, and CT analyses of different filters when restoring Barbara images corrupted with $N_1 + N_2 + N_3$ noises.

Filters	Noise strength $a = 0.5$				Noise strength $a = 0.9$				Noise strength $a = 1.5$			
	MAE	**PSNR**	**MSSIM**	**CT**	**MAE**	**PSNR**	**MSSIM**	**CT**	**MAE**	**PSNR**	**MSSIM**	**CT**
SMF1	26.11	6.43	0.37	0.82	34.85	4.48	0.22	0.84	49.91	2.54	0.11	0.91
SMF2	21.47	12.89	0.49	10.34	25.19	9.87	0.37	10.25	31.24	6.54	0.21	10.29
ILPF	11.07	22.59	0.83	16.70	13.41	20.10	0.80	16.70	17.18	16.84	0.75	16.72
AMF1	20.91	15.61	0.66	27.44	24.14	12.51	0.53	27.43	28.13	9.92	0.33	27.38
AMF2	19.18	18.21	0.73	22.33	20.76	16.15	0.66	22.18	26.29	11.72	0.48	21.88
WGNF	16.52	20.14	0.76	22.93	18.23	18.06	0.69	22.73	28.42	13.65	0.53	22.87
GSSF	10.40	24.55	0.92	9.97	14.54	19.56	0.85	18.29	18.50	14.48	0.76	31.68
AONF	8.81	25.43	0.91	27.20	11.05	22.53	0.86	42.56	15.28	19.75	0.81	84.92
INRF	21.51	18.45	0.62	6.76	33.51	14.04	0.47	6.92	28.20	12.10	0.67	4.97
BNRF	15.14	20.52	0.73	5.28	26.81	15.29	0.58	5.21	44.10	10.90	0.44	5.20
GNRF	10.35	23.68	0.86	4.99	17.46	20.40	0.78	4.99	28.20	18.10	0.67	4.97
FDSMF	**3.73**	**33.41**	**0.97**	**5.14**	**5.23**	**30.77**	**0.94**	**5.97**	**6.52**	**28.92**	**0.93**	**7.51**

with $N_1 + N_2 + N_3$ noises with equal strengths are tabulated in Table 4. Quantitative values tabulated in Tables 1 to 4 clearly indicate that the proposed algorithm is capable of producing better outputs than the outputs of competing algorithms.

4. Discussion, analysis, and comparison

The section provides the analysis, discussion, and comparison of different algorithms, results, and parameters of different methods to demarcate the improved performance of the proposed algorithm. The Standard Morphological Filter (SMF1) replaces all the pixels of the image by the average of morphological closing and opening operations with a static structural element while Soft Morphological Filter (SMF2) replaces all the pixels of the image by the average of soft morphological closing and opening operations with a static structural element. These filters produce extensive blurring in the restored images and are not effective for restoring images corrupted by periodic noise due to their difficulties in differentiating periodic noise frequencies from the uncorrupted ones in the spatial domain. The Ideal Low-Pass Filter (ILPF) admits only specific low-pass region clustered around the Direct Current (DC) coefficient of Fourier transformed image by discarding all high-frequency components irrespective of their corrupted status and, hence, cannot maintain fine image details in the restored image. Aizenburg Mean (AMF1), Median (AMF1), and Windowed Gaussian Notch Filter (WGNF) algorithms stop the filtering process in the low-frequency region RLF defined by a radius distance. Aizenburg Mean (AMF1) reduces the impact of noisy peak position by dividing it by a constant when the ratio of that frequency to the mean of surrounding frequency exceeds a threshold. Aizenburg Median Filter (AMF1) and Windowed Gaussian Notch Filter (WGNF) detect a frequency as corrupted when the ratio of that frequency to the median of surrounding frequencies exceeds a threshold. AMF1 replaces only the corrupted frequency by the median of surrounding frequencies while WGNF reduces the impact of noise not only of the corrupted peak but also of the neighboring frequencies by applying a static sized notch reject filter. These filters fail to provide ample restoration of images corrupted with periodic noises due to the non-adaptive nature of detecting and denoising noisy peak areas. In addition, the low-frequency areas that should be avoided in the filtering process of these filters are statically fixed and do not vary according to the position of corruption.

The Adaptive Optimum Notch Filter (AONF) adaptively determines the low-frequency region to avoid the low frequency components in the filtering process. The optimum low-frequency region is determined by partitioning the frequency image F into non-overlapping arc shaped portions using concentric rings of 5-pixel width. These rings are further divided into 12 equal slices and, subsequently, the average of individual slices in the Fourier transformed image F is found out. The optimum radius of low frequency is the distance of that slice from the inner ring closest to the centre of the image which produces less average value than its corresponding outer ring slice. These low-frequency regions are avoided in the filtering process. AONF uses the hot point detection process of [30] to identify the noisy peaks. Once the noisy peaks are identified, the algorithm uses the traditional region-growing algorithm to quantify the noisy region to design the Gaussian notch filter for denoising the noisy regions. The traditional region-growing algorithm can produce misclassifications in peak detection since it uses the dynamically changing mean value of the already grouped frequency values for checking the similarity of neighbouring frequencies. This region-growing technique is different from the proposed method, in which the algorithm uses the centre peak value to check the similarity for adding the noisy frequency positions to the noise map image. Gaussian Star Shaped Filter (GSSF) uses the traditional region-growing approach to identify the low-frequency regions to avoid this area in the filtering process. The algorithm detects a position exposed to corruption if the frequency value exceeds a certain threshold value and noisy positions are applied with Gaussian star shaped notch filter to diffuse the noisy areas. Interpolation Notch Reject Filter (INRF), Brick-wall Notch Reject Filter (BNRF), and Gaussian Notch Reject Filter (GNRF) algorithms detect a frequency as noisy peak when either the difference between central frequency and the maximum of outer surrounding frequencies of 5 × 5 neighbourhood or the difference between the maximum of frequencies covered by 3 × 3 neighbourhood and the maximum of outer surrounding frequencies of 5×5 neighbourhood exceeds a predefined threshold. However, these algorithms could not provide effective restoration since they do not quantify the area of corruption associated with the peaks. Unlike other filters in the comparative study [1,8,9,17-20,23], the proposed filter first isolates the noisy frequencies from the uncorrupted ones by determining the difference image to make the threshold-based noise detection process easy. A region-growing algorithm that depends on the centre frequency value to group similar pixels is applied for quantifying the noisy regions associated with individual peaks. The proposed filter replaces the detected noisy frequencies by applying recursive switching median filter to generate the restored frequency domain image. The algorithm outperforms other algorithms in terms of subjective and objective metrics due its better capability in identifying and denoising noisy frequencies.

4.1. Parameter analysis

The subsection analyses various parameters used in the proposed algorithm. The window $W_1 \times W_1$ is used for finding the difference image D. The window size W_1 is set to 5 after analyzing different noisy situations and image statistics. T_1 is the radius distance threshold used for fixing the low-frequency region RLF that should be avoided in the noise detection process. Table 5 shows the average PSNR values obtained from restored Lena, Cameraman, and Barbara images with varying noise strengths and T_1 with N_1-type noise. It can be observed from Figures 1(b), 2(b), 3(b), 4(b), 5(b), and 6(b) that the optimum value of T_1 is fully dependent on the corrupting noise type. For adaptively fixing T_1, algorithms need the information of the nearest noisy peak position from the centre of Fourier transformed image F. The proposed algorithm uses. Payman et al. algorithm [8] to fix the radius threshold, T_1. This approach first segments the entire Fourier transformed image to arc shaped portions using concentric rings of 5-pixel width. These rings are further divided into 12 equal slices and, subsequently, the average of individual slices in the Fourier transformed image F is found out. The optimum T_1 value is the distance of the origin of that slice from the inner ring closest to the centre of the image that produces less average value than its corresponding outer ring slice. T_2 is the threshold used for detecting noisy peak positions of the image. Obviously, the optimum T_2 values depend on the individual peak values and are very much heuristic to obtain. By making experiments on different types of images with varying noises and strengths, T_2 is formulated as a function of the central frequency by:

Table 5. Average PSNR values obtained for N_1 noise by varying T_1 values.

Noise	**T_1 values**				
strength, *a*	**10**	**50**	**90**	**110**	**150**
0.1	19.698	37.986	**39.24**	39.24	19.947
0.3	19.655	35.845	**37.72**	37.72	10.449
0.5	19.648	35.523	36.52	**36.59**	6.017
0.7	19.607	34.141	**34.23**	34.23	3.095
0.9	19.551	32.806	**32.44**	32.44	0.913

$$T_2 = \lambda_1 \times D\left(\frac{M}{2}, \frac{N}{2}\right). \tag{20}$$

Table 6 shows the average PSNR values obtained from Lena, Cameraman, and Barbara by varying noise strengths and λ_1 values with N_1-type noise. Accordingly, λ_1 is set to 0.1 to provide better results. T_3 is formulated as:

$$T_3(x, y) = \lambda_2 \times D(x, y). \tag{21}$$

Here, (x, y) is the spatial position of the peak value under consideration. $T_3(x, y)$ makes the region-growing

Table 6. Average PSNR values obtained for N_1 noise by varying λ_1 values.

Noise	**λ_1 values**					
strength, *a*	**0.08**	**0.10**	**0.12**	**0.14**	**0.16**	**0.18**
0.1	37.50	**39.24**	34.16	32.29	31.69	29.09
0.3	35.36	**37.72**	34.45	28.91	27.55	26.50
0.5	31.52	**36.59**	30.62	29.18	26.44	25.73
0.7	32.18	**34.23**	32.01	28.19	26.66	25.11
0.9	32.68	**32.44**	31.33	29.56	26.36	24.56

Table 7. Average PSNR values obtained for N_1 noise by varying λ_2 values.

Noise	**λ_2 values**					
strength, a	**0.7**	**0.75**	**0.8**	**0.85**	**0.9**	0.95
0.1	33.12	34.24	36.11	**39.24**	39.21	31.28
0.3	30.50	31.98	33.95	**37.72**	37.64	31.05
0.5	29.77	30.71	32.97	**36.59**	36.40	30.77
0.7	29.76	31.18	32.36	**34.23**	34.01	30.33
0.9	28.89	30.07	31.48	**32.44**	32.25	29.86

Table 8. Average PSNR values obtained for N_1 noise by varying W_3 values.

Noise	**W_3 values**				
strength, a	**3**	**5**	**7**	**9**	**11**
0.1	39.15	**39.24**	**39.24**	39.23	39.22
0.3	37.43	**37.72**	37.71	37.67	37.64
0.5	36.37	**36.59**	36.50	36.45	36.40
0.7	34.02	**34.23**	34.13	34.06	34.01
0.9	32.35	**32.44**	32.35	32.30	32.25

algorithm add all the pixels surrounding the noisy peak at position (x, y) to the noise map image whose values lie in the interval $[D(x, y), T_3(x, y)]$. The algorithm yielded better results when $\lambda_2 = 0.85$, as can be seen in Table 7. Although the current formulation of parameters provides reasonable results for most of the images, optimization of these parameters can be regarded as a direction for future work by performing heuristic training with artificial neural networks/evolutionary algorithms.

The restoration window, $W_3 \times W_3$, used in the recursive restoration phase needs to ensure that there is sufficient number of uncorrupted frequencies caught in the noise-free frequency set ψ and, hence, the algorithm set $W_3 = 5$. PSNR analysis for various values of W_3 with varying noise levels is shown in Table 8. For Standard Morphological Filter (SMF2), the best quantitative values are selected by varying window sizes. For other algorithms used in the comparative study, the parameters are set as suggested in their respective papers.

5. Conclusion

The paper presented an adaptive frequency-domain-based switching median filter for the restoration of digital images corrupted by periodic noise. The region-growing algorithm incorporated in the binary noise map image generation process improves detection of noisy peak areas of the Fourier transformed image. The recursive restoration algorithm always ensures the existence of uncorrupted frequencies while restoring the corrupted frequencies. Experimental results at various noise levels for naturally/artificially corrupted images show that the proposed algorithm provides better restoration than other competing algorithms do in terms of the quantitative and qualitative metrics used in the analysis. Although the current formulation of thresholds works reasonably well for the majority of images, it can be made adaptive to different images and types of periodic noises by attempting at adaptive training schemes by neural networks/genetic algorithms in the future work.

Acknowledgment

The authors would like to express their gratitude to King Khalid University, Saudi Arabia, for providing administrative and technical support.

References

1. Gonzalez, R.C. and Woods, R.E., *Digital Image Processing*, 3rd Ed., Prentice Hall, Upper Saddle River, NJ (2008).
2. Catrysse, P.B., *Optics for Digital Imaging*, John Wiley & Sons Ltd., Hoboken, NJ (2015).
3. Smolka, B., *Nonlinear Techniques of Noise Reduction in Digital Images*, Silesian University Press, Poland (2004).
4. Arrasmith, W.W., *Systems Engineering and Analysis of Electro-optical and Infrared Systems*, CRC Press, Boca Raton, FL (2015).
5. Lillesand, T., Kiefer, R.W. and Chipman, J., *Remote Sensing and Image Interpretation*, John Wiley & Sons Ltd., Hoboken, NJ (2014).
6. Allinson, N., Anaxagoras, T., Aveyard, J., Arvanitis, C., Bates, R., Blue, A. and Zin, H. "The multidimensional integrated intelligent imaging project (M-I3)", *Nucl. Instrum. Methods Phys. Res. A.*, **604**(1), pp. 196-198 (2009).
7. Jing, W. and Liu, D.C. "2-D FFT for periodic noise removal on strain image", *4th Int. Conf. on Bioinformatics and Biomedical Engrg.*, Chengdu, China, pp. 1-4 (2010).
8. Payman, M., Masoumzadeh, M. and Habibi, M. "A novel adaptive Gaussian restoration filter for reducing periodic noises in digital image", *Signal Image Video Process.*, **7**(5), pp. 1-13 (2013).
9. Zhen, J., Liao, H., Zhang, X. and Wu, Q.H. "Simple and efficient soft morphological filter in periodic noise reduction", *IEEE Region 10 Conf. TENCON*, Hong Kong, pp. 1-4 (2006).
10. Ji, T.Y., Lu, Z. and Wu, Q.H. "Optimal soft morphological filter for periodic noise removal using a particle swarm optimiser with passive congregation", *Signal Process.*, **87**(11), pp. 2799-2809 (2007).
11. Ji, T.Y., Lu, Z. and Wu, Q.H. "A particle swarm optimizer applied to soft morphological filters for periodic noise reduction", In *Applications of Evolutionary*

Computing, pp. 367-374, Springer, Berlin Heidelberg (2007).

12. Zhen, J., Ming, Z., Li, Q. and Wu, Q. "Reducing periodic noise using soft morphology filter", *Int. J. Electron.*, **21**(2), pp. 159-162 (2004).
13. Yun-peng, L., Fang-cheng, L., Li, C. and Li, Y. "Study of the mathematical morphological filter in suppressing periodic narrow bandwidth noise of pd", *Proc. of the Csee* (March 2004).
14. Guttman, N. and Julesz, B. "Lower limits of auditory periodicity analysis", *J. Acoust. Soc. Am.*, **35**(4), pp. 610-610 (1963).
15. Xiaohua, D., Zheng, G. and Chao, H. "An adaptive real-time descreening method based on SVM and improved SUSAN filter", In *Proc. ICASSP*, Dallas, TX, pp. 1462-1465 (2010).
16. Zhouping, W., Wang, J., Nichol, H., Wiebe, S. and Chapman, D. "A median-Gaussian filtering framework for Moiré pattern noise removal from X-ray microscopy image", *Micron*, **43**(2), pp. 170-176 (2012).
17. Aizenberg, I. and Butakoff, C. "Frequency domain median like filter for periodic and quasi-periodic noise removal", In *Image Processing: Algorithms and Systems*, R.D. Edward, T.A. Jaakko, and O.E. Karen, pp. 181-191, Proc. SPIE 4667, San Jose, CA (2002).
18. Aizenberg, I. and Butakoff, C. "Nonlinear frequency domain filter for quasi periodic noise removal", *Int. TICSP Workshop on Spectra Meth. and Multirate Signal Process.*, Tolouse, France, pp. 147-153 (2002).
19. Aizenberg, I. and Butakoff, C. "A windowed Gaussian notch filter for quasi-periodic noise removal", *Image Vis. Comput.*, **26**(10), pp. 1347-1353 (2008).
20. Ketenci, S. and Gangal, A. "Design of Gaussian star filter for reduction of periodic noise and quasi-periodic noise in gray level images", *Int. Symp. on Innovations in Intelligent Systems and Appl.*, Trabzon, Turkey, pp. 1-5 (2012).
21. Payman, M. and Behnampour, M. "Adaptive optimum notch filter for periodic noise reduction in digital images", *Amirkabir Int. J. of Elect. and Electronic. Eng.*, **42**(1), pp. 1-7 (2010). Available: http://www.sid.ir/en/VEWSSID/J_pdf/5064920100101. pdf
22. Hudhud, G.A. and Turner, M.J. "Digital removal of power frequency artifacts using a Fourier space median filter", *IEEE Signal Process. Lett.*, **12**(8), pp. 573-576 (2005).
23. Konstantinidis, A.C., Olivo, A., Munro, P.R., Bohndiek, S.E. and Speller, R.D. "Optical characterisation of a CMOS active pixel sensor using periodic noise reduction techniques", *Nucl. Instrum. Methods Phys. Res. Sect. A-Accel. Spectrom. Dect. Assoc. Equip.*, **620**(2), pp. 549-556 (2010).
24. Grédiac, M., Sur, F. and Blaysat, B. "Removing quasi-periodic noise in strain maps by filtering in the Fourier domain", *Exp. Techniques* (2015).
25. Sur, F. and Grédiac, M. "Automated removal of quasiperiodic noise using frequency domain statistics", *J. Electron. Imaging*, **24**(1), pp. 013003-013003 (2015).
26. Varghese, J., Ghouse, M., Subash, S., Siddappa, M., Khan, M.S. and Bin Hussain, O. "Efficient adaptive fuzzy-based switching weighted average filter for the restoration of impulse corrupted digital images", *IET Image Process.*, **8**(4), pp. 199-206 (2014).
27. Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P. "Image quality assessment: From error measurement to structural similarity", *IEEE Trans. Image Process*, **13**(4), pp. 600-612 (2004).
28. Source of satellite sensed Pompeii image: https://miac.unibas.ch/SIP/06-Restoration.html
29. Source of halftone image: http://2011writersfestival.blogspot.com/2011/01/great-poster-adventure-part-2.html
30. Venkateswarlu, R., Sujata, K.V. and Venkateswara Rao, B. "Centroid tracker and point selection", In *SPIE Proceedings*, **1697**, pp. 520-529 (1992).

Biography

Justin Varghese received the PhD degree in Computer and Information Technology from Manonmaniam Sundaranar University, Tirunelveli, India. He is currently an associate professor with the College of Computer Science, King Khalid University, Abha, Saudi Arabia. He has authored 52 research papers in refereed international journals/proceedings/books, including the IEEE, IET, and Springer journals. His current research interests include signal and image processing, visual perception, mathematical morphology, fuzzy logic, and pattern recognition. Dr. Varghese received the best paper award at the IEEE International Conference on Advanced Computing, NITK, Surathkal, India, in 2006. He is a Senior Member of the IEEE.