

RESEARCH

Open Access



# A robust anisotropic diffusion filter with low arithmetic complexity for images

Resmi R. Nair<sup>1</sup>, Ebenezer David<sup>2\*</sup> and Sivakumar Rajagopal<sup>1</sup>

## Abstract

Image smoothing with edge preservation in the presence of outliers is a challenge in image processing. Anisotropic diffusion smoothing is a well-established paradigm in digital image smoothing with edge preservation. Anisotropic diffusion smoothing filters are not robust to impulse noise. They employ two or more stages for robustness in the presence of outliers, and, therefore, they demand significantly increased arithmetic operations for robustness; consequently, they are not power efficient. This paper introduces a low arithmetic complexity image smoothing model and proposes an intrinsically robust and power efficient algorithm for anisotropic diffusion smoothing of images. The algorithm outperforms the foundational robust smoothing algorithms in terms of the standard performance metrics and visual quality.

**Keywords:** Anisotropic filters, Computational complexity, Image denoising, Nonlinear filters, Robustness

## 1 Introduction

Image smoothing is an essential operation in digital image processing. One well-established approach to image smoothing is scale-space filtering [1]. The basic scale-space filter is an isotropic diffusion smoothing filter which produces image smoothing with poor edge retention. Perona-Malik (PM) introduced an edge-preserving anisotropic diffusion filter [2] which also formed the basis for subsequent developments in anisotropic diffusion smoothing. However, filters based on anisotropic diffusion failed to perform well in the presence of impulse noise. Black et al. [3] developed a statistical interpretation of anisotropic diffusion from the view point of robust statistics and proposed a diffusion coefficient based on Tukey's biweight robust error norm. Ling et al. [4] proposed a two-stage anisotropic median filtering in which the first stage is anisotropic diffusion; the diffusion coefficient is based on Tukey's biweight robust error norm proposed by Black et al., and the second stage is median filtering of the diffused image. Ham et al. [5] introduced an adaptive robust scale-space filter in which the robustness is provided

by the incorporation of a normalization term. Subsequent developments [6–13] have followed these foundational models [2–5].

Wu et al. [6] proposed a two-stage diffusion scheme for random valued impulse noise in which the ENI (edge pixels, noisy pixels, and image pixels) concept is employed in the first stage for the identification of noisy pixels. In the anisotropic diffusion stage, two diffusion functions, namely, controlling speed function and controlling fidelity function are employed for robustness. Wang et al. [7] introduced an adaptive switching anisotropic diffusion filter which operates in two modes. In one mode, the filter acts as a direction weighted median filter for impulse noise corrupted pixels, and in the other mode, the filter acts as an adaptive anisotropic diffusion filter for Gaussian noise-corrupted pixels. The filter selects the mode in accordance with a local difference factor of the pixel. Xia et al. [8] proposed a two-stage diffusion filter, which employs a switching function in the diffusion algorithm. The filter is designed to handle both salt and pepper noise-corrupted and random-valued impulse noise-corrupted images. In the first stage, noise detectors are employed to determine the noisy pixels. The value of the switch function in the diffusion algorithm depends on the noise detector output; therefore, the noisy pixels are replaced in the diffusion stage and the uncorrupted pixels are left unchanged. In the second stage, anisotropic

\* Correspondence: [auebenezer@yahoo.co.in](mailto:auebenezer@yahoo.co.in)

<sup>2</sup>Department of Electronics and Communication Engineering, Anna University, Chennai, Tamil Nadu 600025, India

Full list of author information is available at the end of the article

diffusion is carried out at two levels of filtering, namely, coarse filtering and fine filtering for the effective removal of impulse noise. Khan et al. [9] proposed a two-stage robust adaptive anisotropic diffusion filter. In the preprocessing stage, noisy pixels are identified and are replaced by the median value of the local neighborhood. In the second stage, an adaptive anisotropic diffusion is carried out. Haiying et al. [10] proposed a two-stage diffusion tensor-based robust diffusion filter. A clean pixel excluder is proposed to identify the noisy pixels in the first stage; in the second stage, an adaptive diffusion is carried out in accordance with the diffusion tensor to replace the noisy pixels. Sung et al. [11] proposed a single-pass dictionary-based anisotropic diffusion in which an offline dictionary and an online dictionary are employed. In the offline dictionary, diffusivity values and diffusion path kernels for different image patches are stored. A single-pass adaptive anisotropic diffusion is carried out in the online dictionary processing by choosing appropriate diffusivity values and diffusion path-based kernels from the offline dictionary for each image patch. Veerakumar et al. [12] introduced a two-stage adaptive anisotropic diffusion filter in which noisy pixel determination is the first stage and an adaptive anisotropic diffusion is the second stage. In the noisy pixel determination stage, each image pixel is compared against the maximum and the minimum pixel gray-level values for the determination of salt and pepper noise; a rank-ordered logarithmic difference approach is proposed for the identification of the pixels corrupted by the random-valued impulse noise. In the diffusion stage, an adaptive anisotropic diffusion is carried out for the replacement of noisy pixels in accordance with the local neighborhood characteristics.

Existing robust diffusion smoothing filters [4–13] employ a variety of additional nonlinear techniques for improving the performance in the presence of outliers. They are generally complex in terms of hardware (adders and multipliers) and are not power efficient. This paper proposes a simple first-order anisotropic diffusion model which exhibits a performance superior to the standard second-order parabolic partial differential equation (PDE)-based diffusion models. The proposed model is intrinsically robust, simple in hardware complexity, and power efficient and outperforms the foundational models in terms of performance metrics and visual quality.

The first-order robust diffusion smoothing filter which is intrinsically robust to impulse noise is derived in Section 2 based on Albert Einstein's stochastic argument for Brownian motion. Section 3 describes the performance of the proposed filter in detail in comparison with three foundational diffusion smoothing filters [2, 4, 5] and one extended model [12]. Section 4 concludes by highlighting the significance of the proposed filter in the context

of low-power digital signal processing (DSP) and embedded environments.

## 2 The proposed methodology

The methodology is based on the concept that the immediate future value of the intensity of a candidate pixel can be predicted in terms of the estimate of the increment or decrement of the pixel intensity in terms of the directional derivatives with respect to neighborhood pixels inside an open subset. The open subset includes four neighborhood pixels along the north, south, east, and west directions with respect to the center pixel of a  $3 \times 3$  window. The prediction equation is a novel mathematical model developed from Einstein's stochastic argument for Brownian motion. The mathematical model involves a candidate pixel, four neighborhood pixels, a weighting function, and a control parameter. The mathematical model is translated to an algorithm. For numerical computation of the algorithm, directional derivatives along north, south, east, and west directions of the candidate pixel are calculated; a median estimate of the directional derivatives is obtained. The candidate pixel intensity value is updated by adding a value proportional to the median estimate of the directional derivatives. This process is repeated for all the pixels in the image. The whole procedure is iterated  $n$  times until specified signal to noise ratio performance is achieved. Different test images corrupted with various noise types at different noise intensities are selected for testing the algorithm. Simulation is carried out on MATLAB R2014a installed in a system powered by 567u7 Intel (R) core (TM) i5-5200 U CPU @ 2.20 GHz, 64-bit Operating System with 4 GB RAM. The performance of the algorithm is validated in terms of visual results and the performance metrics peak signal to noise ratio (PSNR), structural similarity index metric (SSIM), and edge preservation index (EPI).

### 2.1 A first-order robust diffusion smoothing model based on Einstein's stochastic argument for Brownian motion

Let  $I(x, t)$  be a space-time continuous function at position  $x$  and time  $t$ . Following Albert Einstein's stochastic argument for Brownian motion of suspended particles in a fluid medium [14], one can write

$$I(x, t + \tau) = I(x, t) + \tau(\partial I / \partial t) \quad (1)$$

where the right hand side (RHS) of (1) is expressed as a two term Taylor's series [14]

$$\begin{aligned}
 I(x, t) + \tau(\partial I/\partial t) &= I(x, t) \int_{-\infty}^{+\infty} \varnothing(\varepsilon) d\varepsilon \\
 &+ (\partial I/\partial x) \int_{-\infty}^{+\infty} \varepsilon \varnothing(\varepsilon) d\varepsilon \\
 &+ (\partial^2 I/\partial x^2) \int_{-\infty}^{+\infty} (\varepsilon^2/2) \varnothing(\varepsilon) d\varepsilon
 \end{aligned} \tag{2}$$

In (2),  $I(x, t)$  and  $I(x, t + \tau)$  represent the number of Brownian particles per unit volume at time  $t$  and  $t + \tau$  respectively,  $t + \tau$  is the next time instant,  $\varnothing(\varepsilon) = \varnothing(-\varepsilon)$ ,  $\int_{-\infty}^{+\infty} \varnothing(\varepsilon) d\varepsilon = 1$ ,  $\varepsilon$  and  $x$  are the spatial variables, and  $\varnothing(\varepsilon)$  is the probability density distribution function. Since  $\int_{-\infty}^{+\infty} \varnothing(\varepsilon) d\varepsilon = 1$  and  $\varnothing(\varepsilon) = \varnothing(-\varepsilon)$ , a simple manipulation of (2) eliminates the first term on the left hand side (LHS) and the first and second terms on the RHS of (2). Therefore, (2) can be written as.

$$\partial I/\partial t = (1/\tau)(\partial^2 I/\partial x^2) \int_{-\infty}^{+\infty} (\varepsilon^2/2) \varnothing(\varepsilon) d\varepsilon \tag{3}$$

By applying (1) to the LHS of (3)

$$I(x, t + \tau) = I(x, t) + (1/\tau) \int_{-\infty}^{+\infty} (\partial^2 I/\partial x^2) (\varepsilon^2/2) \varnothing(\varepsilon) d\varepsilon \tag{4}$$

Rearranging (4),

$$I(x, t + \tau) = I(x, t) + \left[ (1/\tau) \int_{-\infty}^{+\infty} (\varepsilon^2/2) \varnothing(\varepsilon) d\varepsilon \right] (\partial^2 I/\partial x^2) \tag{5}$$

Let the second term on the RHS of (5) be interpreted as a statistical expectation of the directional derivatives  $\partial I/\partial_{Ex}$ ,  $\partial I/\partial_{Wx}$  resulting from  $(\partial/\partial x)$ .  $(\partial I/\partial x) = (\partial^2 I/\partial x^2)$ , where  $I = I(x, t)$  is the number density,  $E$  and  $W$  are the east and west directions respectively, the dot symbol represents the divergence, and the time  $t$  is a parameter. Let  $(1/\tau) \int (\varepsilon^2/2) \varnothing(\varepsilon) d\varepsilon$  be defined as spreading function  $D(x)$ . Equation (5) says that the function  $D(x)$  diffuses the concentrated mass  $I(x, t)$  throughout the length  $x$  from  $x = x_0$  to  $x = x_L$  ( $x_0$  and  $x_L$  are boundaries) with Gaussian probability density distribution at equilibrium. In the context of DSP, let this be interpreted as system response to white noise excitation whose autocorrelation is a concentrated impulse [15]. Function  $D(x)$  becomes the convolution kernel. Extending (5) to two dimensions

$$I(x, y, t + \tau) = I(x, y, t) + D(x, y) [(\partial^2 I/\partial x^2) + (\partial^2 I/\partial y^2)] \tag{6}$$

Expression (6) in terms of the inner product becomes

$$I(x, y, t + \tau) = I(x, y, t) + (\nabla \cdot D(x, y) \nabla I), \tag{7}$$

where  $\nabla$  is the “del” operator and the second term on the RHS of (7) is the divergence of the gradient of  $I(x, y, t)$ . Assuming that an image is a volume with one dimension vanishingly small and invoking the divergence theorem

$$I(x, y, t + \tau) = I(x, y, t) + [D_N \nabla_N I + D_S \nabla_S I + D_E \nabla_E I + D_W \nabla_W I], \tag{8}$$

where  $\nabla_N I$ ,  $\nabla_S I$ ,  $\nabla_E I$ , and  $\nabla_W I$  represent the directional derivatives along north, south, east, and west directions, and  $D_N$ ,  $D_S$ ,  $D_E$ , and  $D_W$  are diffusion functions. Equation (8) describes a spatio-temporal spreading of the number density  $I(x, y, t)$  in which the current value of the number density is updated by a statistical estimate of flux density which tends to reach an equilibrium state such that  $I(x, y, p + 1) = I(x, y, p)$  after some  $t = p + 1$  when the probability density distribution of the process settles to Gaussian [16]. Equations (4)–(8) are based on the continuum hypothesis; for sampled time and space variables, the last equation becomes

$$\begin{aligned}
 I(u\Delta x, v\Delta y, (n + 1)\Delta t) &= I(u\Delta x, v\Delta y, n\Delta t) \\
 &+ (\Delta t/(\Delta x)^2) \cdot [D_N \nabla_N I + D_S \nabla_S I + D_E \nabla_E I + D_W \nabla_W I],
 \end{aligned} \tag{9}$$

where  $\Delta t$  is the time sampling interval;  $\Delta x$  and  $\Delta y$  are the space sampling intervals;  $u$ ,  $v$ , and  $n$  are integers including zero;  $\Delta x = \Delta y$ ; and  $I$  in the second term on the RHS is  $I(u\Delta x, v\Delta y, n\Delta t)$ . If the integers  $u$  and  $v$  take a value equal to zero, the corresponding image is the initial or the original image. It becomes an initial value problem which is a necessary condition for a causal Markov process. Consistent with (5), the diffusion of  $I(u\Delta x, v\Delta y, n\Delta t)$  is interpreted as the accumulation of the expectation of the directional derivatives with respect to iteration number  $n$ , starting with an initial number density as a delta function at  $n = 0$ . At equilibrium,  $I(u\Delta x, v\Delta y, (m + 1)\Delta t) = I(u\Delta x, v\Delta y, m\Delta t)$  for some  $n = m$ , and is a Gaussian distributed function. Since mean is the posterior estimate in the least square sense for Gaussian distribution [16], (9) can be written as

$$\begin{aligned}
 I(u\Delta x, v\Delta y, (n + 1)\Delta t) &= I(u\Delta x, v\Delta y, n\Delta t) \\
 &+ (\Delta t/(\Delta x)^2) \cdot (4) [(1/4)(D_N \nabla_N I + D_S \nabla_S I + D_E \nabla_E I + D_W \nabla_W I)]
 \end{aligned} \tag{10}$$

$$\begin{aligned}
 I(u\Delta x, v\Delta y, (n + 1)\Delta t) &= I(u\Delta x, v\Delta y, n\Delta t) \\
 &+ \frac{4\Delta t}{(\Delta x)^2} \\
 &\times \left[ \frac{1}{4} \sum_{k=1}^4 D_k \nabla_k I \right],
 \end{aligned}
 \tag{11}$$

where  $k = 1, 2, 3, 4$  represent  $N, S, E,$  and  $W$  directions, respectively;  $D_k$  are associated with the Gaussian kernel; and the second term on the RHS of (11) is the expectation of  $\nabla_k I$  such that the estimate is the maximum likelihood (ML) estimate in the  $L_2$  sense [17]. Let  $D_k$  be chosen as  $g(\nabla_k I)$  so that (11) can be written as

$$\begin{aligned}
 I(u\Delta x, v\Delta y, (n + 1)\Delta t) &= I(u\Delta x, v\Delta y, n\Delta t) \\
 &+ \lambda \left[ \frac{1}{4} \sum_{k=1}^4 g(\nabla_k I) \nabla_k I \right],
 \end{aligned}
 \tag{12}$$

where control parameter  $\lambda = 4\Delta t/(\Delta x)^2$ . If  $g(\nabla_k I)$  is semigroup, (12) corresponds to

$$\begin{aligned}
 I(u\Delta x, v\Delta y, (n + 1)\Delta t) &= I(u\Delta x, v\Delta y, n\Delta t) \\
 &* G_\sigma(u\Delta x, v\Delta y, n\Delta t),
 \end{aligned}
 \tag{13}$$

where (13) defines scale-space filtering which was introduced by Witkin [1]. According to this definition, the scale space of an image can be derived by convolving the image by a Gaussian kernel  $G_\sigma(u\Delta x, v\Delta y, n\Delta t)$  where  $\sigma$  is the

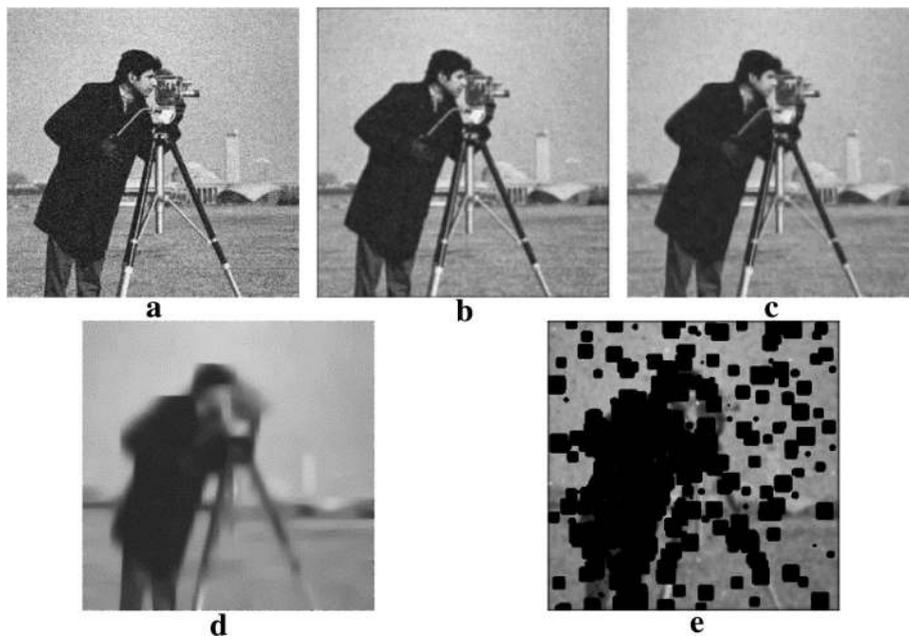
standard deviation, and the asterisk represents convolution [1]. If impulse noise is present, weighted averaging in (12) spreads the impulse noise corrupted image gradients along with noise free image gradients, as is clear from (5–7) and Fig. 1e. For Laplacian or long-tailed distribution, median is the posterior estimate and, hence, the second term on the RHS of (12) is an expectation such that it is an ML estimate in the  $L_1$  sense [17]. Then, it is possible to write (12) as

$$\begin{aligned}
 I(u\Delta x, v\Delta y, (n + 1)\Delta t) &= I(u\Delta x, v\Delta y, n\Delta t) \\
 &+ \lambda g(\text{Med}[\nabla_k I]) \text{Med}[\nabla_k I].
 \end{aligned}
 \tag{14}$$

Equation (14) describes a smooth first-order diffusion process that is robust to impulse noise or outliers with long-tailed distributions. It is first order on account of the fact that the median estimate is simply the selection of one of the first order spatial derivatives  $\nabla_N I, \nabla_S I, \nabla_E I,$  and  $\nabla_W I$  in (9) after sorting. Median sorting renders (14) optimal in the sense of the length functional [18]. A first-order space-time differential equation has an exponential impulse response, and, therefore, the weighting function or kernel function in (14) can be chosen as an exponential function

$$g(\text{Med}[\nabla_k I]) = \exp.(-(\text{Med}[\nabla_k I]/K)),
 \tag{15}$$

where  $K$  is a constant; (15) is consistent with first-order smoothing process. Equation (14) is free



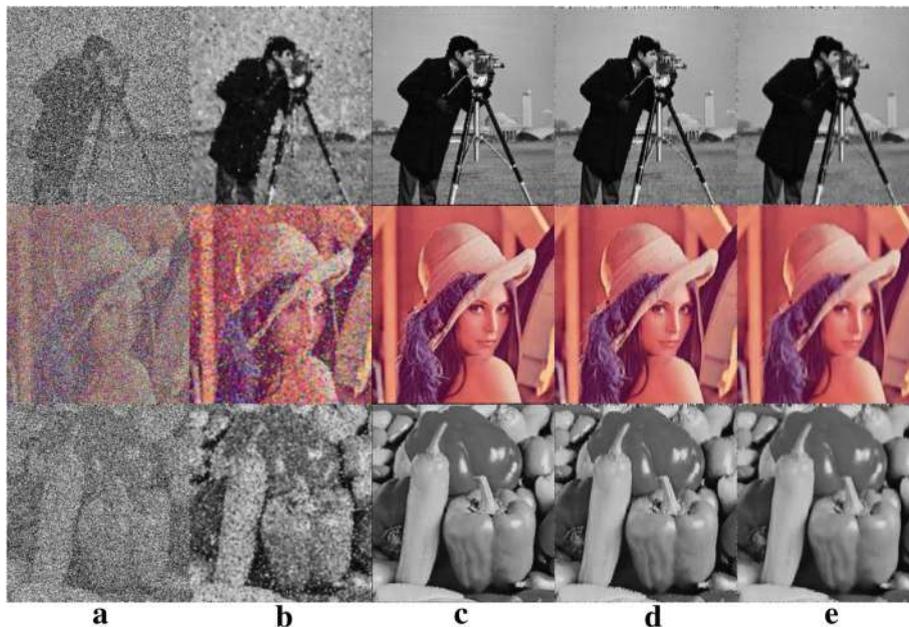
**Fig. 1** Comparison of **a** noisy image, **b** PM as a smoother at  $n = 5$ , **c** FORADF as a smoother at  $n = 5$ , **d** FORADF at  $n = 25$ , and **e** the spreading effect in PM. Noisy image **a** is corrupted with Gaussian noise of variance 0.1.  $n$  represents the number of iterations. The dark patches indicate the spreading effect in PM algorithm



**Fig. 2** Performance of **a** RF ( $\lambda = 0.25$ ), **b** AMD ( $\lambda = 0.25$ ), **c** AADF ( $\lambda = 0.25$ ), **d** FORADF ( $\lambda = 0.25$ ), **e** FORADF ( $\lambda = 1$ ) for salt and pepper noise at 20% density for  $n = 5$ .  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations

of the spreading effect with respect to impulse noise samples on account of the elimination of impulse noise-corrupted gradients prior to smoothing. This is in contrast to the well-established mean filter [19], PDE-based diffusion smoothing filters [20], bilateral filters [21], and guided filters [22, 23] which

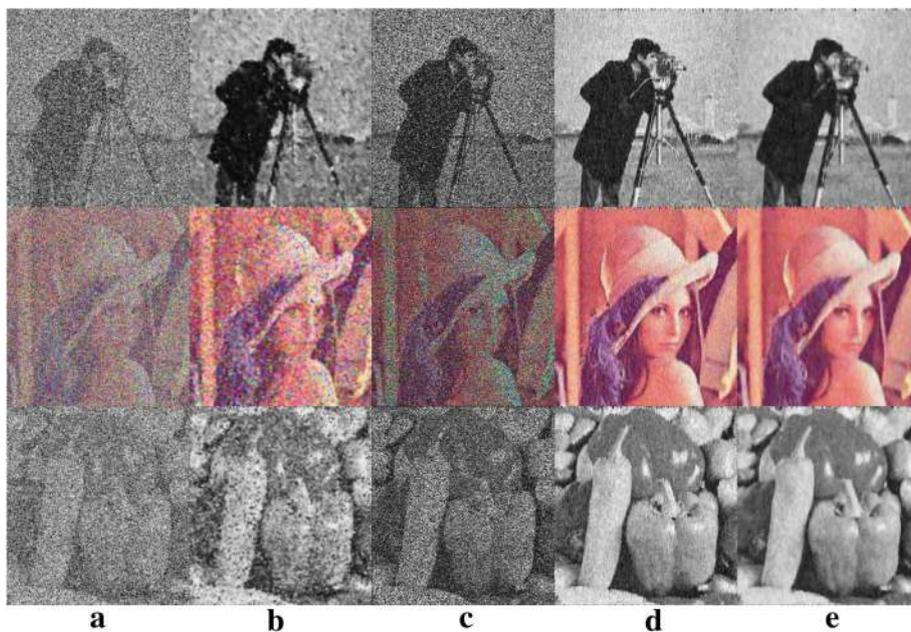
are not intrinsically robust. They require pre-diffusion processing and elaborate post-diffusion processing for good performance in the presence of outliers [4–10, 19–23]. It is shown in Section 3 that (14) provides smoothing, Gaussian and impulse noise elimination, and edge preservation and is



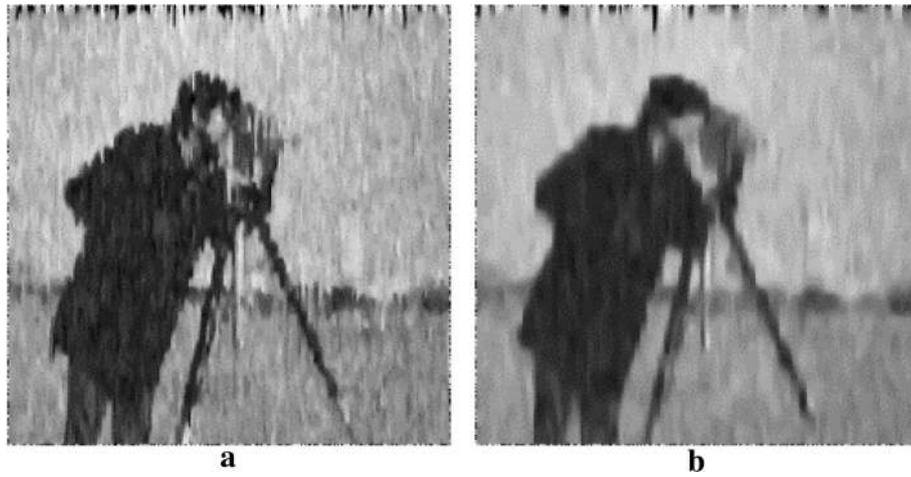
**Fig. 3** Performance of **a** RF ( $\lambda = 0.25$ ), **b** AMD ( $\lambda = 0.25$ ), **c** AADF ( $\lambda = 0.25$ ), **d** FORADF ( $\lambda = 0.25$ ), and **e** FORADF ( $\lambda = 1$ ) for salt and pepper noise at 70% density.  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations



**Fig. 4** Performance of **a** RF ( $\lambda = 0.25$ ), **b** AMD ( $\lambda = 0.25$ ), **c** AADF ( $\lambda = 0.25$ ), **d** FORADF ( $\lambda = 0.25$ ), and **e** FORADF ( $\lambda = 1$ ) for low density mixed noise.  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations. The results are given for low density mixed noise, i.e., Gaussian noise of variance 0.1 plus salt and pepper noise of density 20%



**Fig. 5** Performance of **a** RF ( $\lambda = 0.25$ ), **b** AMD ( $\lambda = 0.25$ ), **c** AADF ( $\lambda = 0.25$ ), **d** FORADF ( $\lambda = 0.25$ ), and **e** FORADF ( $\lambda = 1$ ) for high-density mixed noise.  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations. The results are given for high-density mixed noise, i.e., Gaussian noise of variance 0.1 plus salt and pepper noise of density 70%



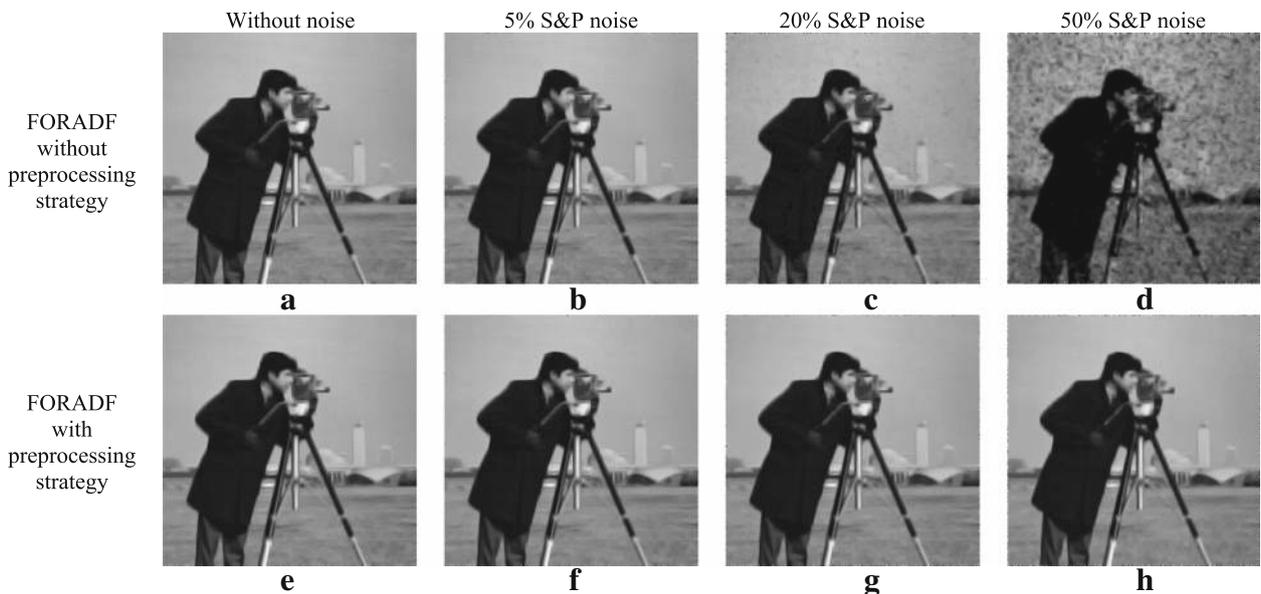
**Fig. 6** Performance of FORADF ( $\lambda = 1$ ) for very high-density mixed noise **a** at  $n = 5$  and **b**  $n = 25$ .  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations. The results are given for very high-density mixed noise, i.e., Gaussian noise of variance 0.1 plus salt and pepper noise of density 90%

intrinsically robust. We also demonstrate that (14) is the simplest in terms of arithmetic complexity. In the context of digital image processing, (14) can be algorithmically expressed as

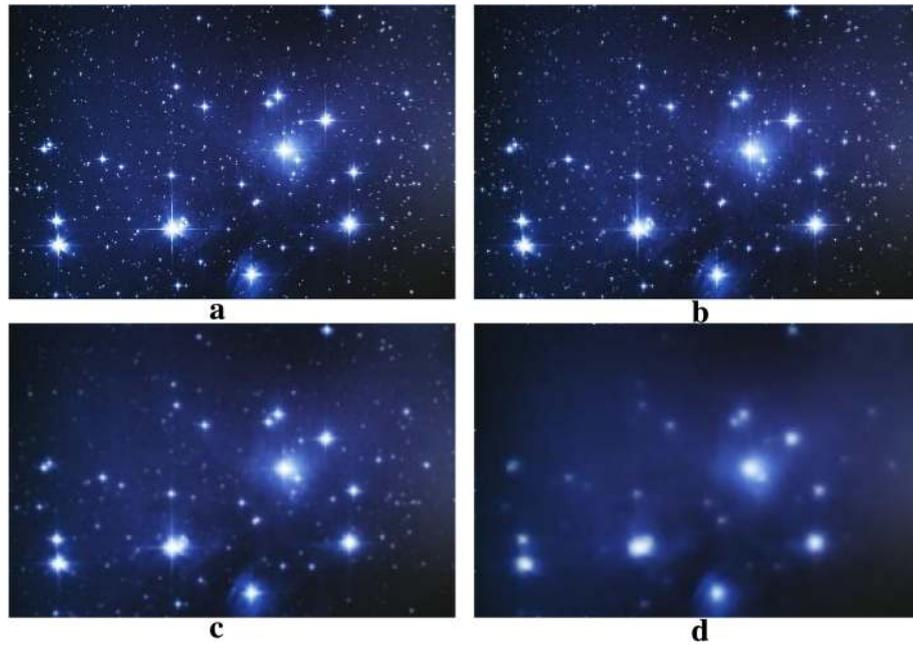
$$I_{i,j}^{n+1} = I_{i,j}^n + \lambda g(\text{Med}[\nabla_k I]) \cdot \left\{ \text{Med} \left[ \begin{matrix} (I_{i-1,j}^n - I_{i,j}^n), (I_{i+1,j}^n - I_{i,j}^n), \\ (I_{i,j+1}^n - I_{i,j}^n), (I_{i,j-1}^n - I_{i,j}^n) \end{matrix} \right] \right\}, \quad (16)$$

$$I_{i,j}^{n+1} = I_{i,j}^n + \lambda g \left\{ \text{Med} \left[ \begin{matrix} (I_{i-1,j}^n - I_{i,j}^n), (I_{i+1,j}^n - I_{i,j}^n), \\ (I_{i,j+1}^n - I_{i,j}^n), (I_{i,j-1}^n - I_{i,j}^n) \end{matrix} \right] \right\} \cdot \left\{ \text{Med} \left[ \begin{matrix} (I_{i-1,j}^n - I_{i,j}^n), (I_{i+1,j}^n - I_{i,j}^n), \\ (I_{i,j+1}^n - I_{i,j}^n), (I_{i,j-1}^n - I_{i,j}^n) \end{matrix} \right] \right\}, \quad (17)$$

where  $i, j$  are spatial indices;  $n$  is the iteration number;  $\lambda$  is the control parameter;  $I_{i,j}^n$  represents the pixel



**Fig. 7** Performance of FORADF ( $\lambda = 1, n = 10$ ) with and without preprocessing strategy. Without preprocessing strategy: **a** without noise, **b** - **d** 5%, 20%, and 50% salt and pepper noise respectively. With preprocessing strategy: **e** without noise, **f** - **h** 5%, 20%, and 50% salt and pepper noise respectively.  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations



**Fig. 8** Robust smoothing of color Pleiades image for  $n = 5$ . **a** Original image, **b** RF ( $\lambda = 0.25$ ), **c** AMD ( $\lambda = 0.25$ ), and **d** FORADF ( $\lambda = 1$ ).  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations. The results are given to demonstrate the superiority in robustness of the proposed FORADF in comparison with other robust algorithms

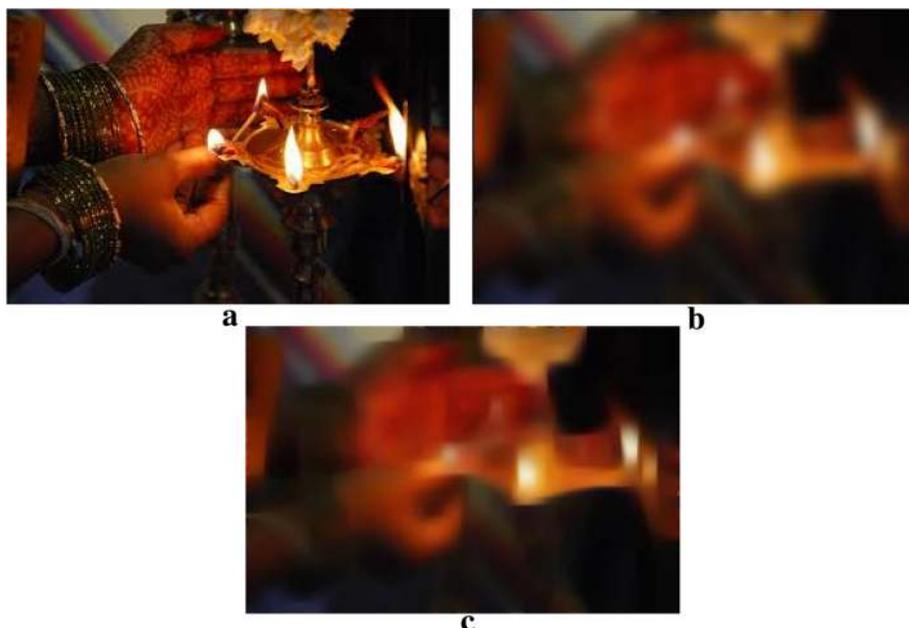
intensity of an image pixel after  $n$  iterations at the spatial location  $(i, j)$ ;  $(I_{i-1,j}^n - I_{i,j}^n)$ ,  $(I_{i+1,j}^n - I_{i,j}^n)$ ,  $(I_{i,j+1}^n - I_{i,j}^n)$ , and  $(I_{i,j-1}^n - I_{i,j}^n)$  represent first-order spatial derivatives along north, south, east, and west directions, respectively; and  $g(\cdot)$  is an exponential function given by (15). Perona-Malik [2] and Canny [24] have suggested the procedure for choosing the value(s) for  $K$ . This procedure is followed in fixing  $K = 2$ . Algorithm (16) works well for any value of  $\lambda$  in the interval  $0 < \lambda \leq 1$  in accordance with Courant-Friedrichs-Lewy (CFL) condition [25]. With  $0 < \lambda \leq 1$ , the algorithm is an excellent smoother as  $n \rightarrow \infty$ , and, therefore, the number of time iterations can be selected on the basis of objective performance metrics and subjective visual quality. Algorithm (17) involves the following step by step process: with respect to an image pixel  $I_{i,j}^n$ , i.e., the center pixel, along with its four neighboring pixels along north ( $I_{i-1,j}^n$ ), south ( $I_{i+1,j}^n$ ), east ( $I_{i,j+1}^n$ ), and west ( $I_{i,j-1}^n$ ) directions,

1. Get the four directional derivatives of the center pixel along north, south, east, and west directions by calculating the difference in pixel intensities of each neighboring pixel along that direction to that of the center pixel. Pixel padding can be used to process the first row, last row, first column, and last column image pixels.

2. Select the median value of the four directional derivatives of step 1.
3. Using the median value the directional derivatives of step 2, find the value of the weighting function using the analytical formula given in (15).
4. Select an appropriate control parameter ( $\lambda$ ) value in between 0 and 1, which gives an optimal performance in  $L_1$  sense for the given input image.
5. Find a scalar value which is the product of the values of step 2, step 3, and step 4.
6. To get an updated pixel intensity value  $I_{i,j}^{n+1}$ , add the value of step 5 to the pixel intensity of the center pixel  $I_{i,j}^n$ .
7. In case of color images, the same procedure has to be extended to three image planes.

### 3 Results and discussion

Three foundational diffusion smoothing methods [2, 4, 5] and one extended model are considered for comparison. The benchmark Perona-Malik algorithm [2] provides edge-preserving Gaussian smoothing. The goal of the other two foundational diffusion smoothing methods [4, 5] is to provide image smoothing while suppressing impulse noise without affecting edge information. The state-of-the-art extended model [12] is included to highlight the increased computational complexity by adding additional stages in attempting to improve the performance. The proposed



**Fig. 9** Anisotropic smoothing of lamp image for  $n = 25$ . **a** Original image, **b** PM ( $\lambda = 0.25$ ), and **c** FORADF ( $\lambda = 1$ ).  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations. The results are given to demonstrate the superiority in anisotropic behavior of the proposed FORADF in comparison with the benchmark Perona-Malik anisotropic diffusion filter

algorithm (16) is primarily a low arithmetic complexity, edge-preserving smoothing algorithm, and impulse noise removal is an additional strength of the algorithm without a requirement for additional stages. Other methods [6–12] are the extensions or modifications of the three foundational methods by adding additional stages for the removal of impulse noise which results in high complexity and poor smoothing. However, one method [12] from [6–12] is considered for comparison to highlight the trade-off between performance and complexity. Salt and pepper noise is included as a convenient model of strong outliers; similar results can be generated for random-valued impulse noise. The proposed algorithm is mainly meant for edge-preserving image smoothing at low complexity in the presence of salt and pepper noise. The results for mixed noise, i.e., mild Gaussian noise of variance 0.1 plus salt and pepper noise are also included to show that no image will ever be free of a mild Gaussian noise. This result for mixed noise is only an addition. Cameraman image, Lena image, and Pepper image of size  $512 \times 512$  are selected as test images. An additional non-iterative

simplifying strategy employed is that if the center pixel inside the mask is an impulse corrupted pixel, it can be replaced by an immediate neighbor pixel [26], assuming that impulse noise pixels do not occur consecutively.

Figure 1a shows an image corrupted by Gaussian noise of variance 0.1. Figure 1b and c show the performance of PM algorithm and the proposed first-order robust anisotropic diffusion filter (FORADF), respectively. The number of iterations is kept low at 5. It is confirmed that FORADF works well for Gaussian noise. Figure 1d demonstrates that FORADF is a good smoother ( $n = 25$ ). Figure 1e shows the spreading effect in PM algorithm for the image corrupted by 20% salt and pepper noise for iteration number  $n = 20$ .

Figure 2a–e show the results of the robust filter (RF;  $n = 5$ ) [5], anisotropic median diffusion (AMD;  $n = 5$ ) [4], adaptive anisotropic diffusion filter (AADF;  $n = 1$ ) [12], FORADF( $\lambda = 0.25, n = 5$ ), and FORADF( $\lambda = 1, n = 5$ ), respectively, with salt and pepper noise at 20%, for the three test images. It can be seen that AMD ( $K_1 = 0.5, \lambda = 0.25$ ) gives better performance than the robust filter ( $\lambda = 0.25$ ). The proposed FORADF ( $\lambda = 0.25$  and  $\lambda = 1$ ) outperforms the RF and AMD. It is clear from the results that FORADF is robust, preserves edge information, and is a good smoother. The performance of AADF is very good in terms of salt and pepper removal, but it is a poor smoother in comparison with FORADF. The major advantage of FORADF in comparison with AADF is very low complexity.

**Table 1** Performance for Gaussian noise

Methods	Noise type	Variance	$\lambda$	NI	PSNR	SSIM
PM	GN	0.1	0.25	5	19.267	0.8287
FORADF	GN	0.1	1	5	20.233	0.8384
FORADF	GN	0.1	1	100	Good smoothing	

GN Gaussian noise, NI number of iteration, S&P salt and pepper noise

**Table 2** Performance for low-density salt and pepper noise

Methods	Test image	NI	Noise type – S&P Noise % – 20		Noise type – S&P Noise % – 70		Noise type-GN + S&P Noise % – 0.1 + 20		Noise type-GN + S&P Noise % – 0.1 + 70	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
RF $\lambda = 0.25$	Cameraman	5	19.413	0.8673	12.5457	0.7796	16.906	0.8159	12.4372	0.7782
	Lena	5	20.4622	0.9693	12.6513	0.8751	12.9513	0.8622	13.0337	0.8639
	Pepper	5	19.7905	0.9622	13.2004	0.8719	17.2988	0.9096	12.9513	0.8622
AMD $\lambda = 0.25$	Cameraman	5	26.079	0.9958	20.5091	0.8927	19.178	0.8270	19.7392	0.8033
	Lena	5	28.7007	0.9964	18.0798	0.9520	16.4546	0.8962	18.2120	0.9057
	Pepper	5	28.6385	0.9923	18.6443	0.9462	19.5751	0.9199	16.4546	0.8962
AADF $\lambda = 0.25$	Cameraman	1	40.1719	0.9997	28.7566	0.9967	14.4454	0.8380	10.5868	0.7873
	Lena	1	41.9936	0.9999	26.0650	0.9975	15.9324	0.9441	9.3239	0.8923
	Pepper	1	42.2684	0.9999	31.4408	0.9999	14.4793	0.9380	10.8085	0.9021
FORADF $\lambda = 0.25$	Cameraman	5	34.8502	0.9997	24.4484	0.9947	19.2064	0.8296	17.6850	0.8265
	Lena	5	37.7526	0.9998	27.0534	0.9962	18.1284	0.9212	19.9176	0.9315
	Pepper	5	37.5130	0.9998	25.5219	0.9942	19.5065	0.9260	18.1284	0.9212
FORADF $\lambda = 1$	Cameraman	5	28.1563	0.9925	23.916	0.9771	20.080	0.8377	18.822	0.8328
	Lena	5	31.2451	0.9990	26.2455	0.9951	19.3510	0.9271	20.5072	0.9381
	Pepper	5	31.7647	0.9986	25.1724	0.9883	20.7433	0.9342	19.3510	0.9271

Figure 3a–e show the results of the RF ( $\lambda = 0.25$ ,  $n = 5$ ), AMD ( $\lambda = 0.25$ ,  $n = 5$ ), AADF ( $\lambda = 0.25$ ,  $n = 1$ ), FORADF ( $\lambda = 0.25$ ,  $n = 5$ ), and FORADF ( $\lambda = 1$ ,  $n = 5$ ), respectively, with salt and pepper noise at 70% for the three test images. The proposed filter works very well at higher noise densities and for very low number of iterations, whereas RF and AMD fail to provide acceptable performance at high-density salt and pepper noise with low number of iterations. The performance of RF [5] and AMD [4] can be improved by increasing the number of iterations. For acceptable performance at higher noise densities, both RF [5] and AMD [4] require the number of iterations be high (above 100), whereas FORADF performs better at much lower number of iterations. Since the focus of AADF is on salt and pepper noise removal, it works well at higher noise densities also. It is clear from the result that FORADF outperforms AADF in terms of image smoothing and computational complexity.

Figure 4a–e show the results of RF ( $\lambda = 0.25$ ,  $n = 5$ ), AMD ( $\lambda = 0.25$ ,  $n = 5$ ), AADF ( $\lambda = 0.25$ ,  $n = 1$ ), FORADF ( $\lambda = 0.25$ ,  $n = 5$ ), and FORADF ( $\lambda = 1$ ,  $n = 5$ ) for  $n = 5$ , respectively, for the three images corrupted by mixed noise consisting of Gaussian noise of variance 0.1 plus salt and pepper noise of density 20%. The proposed FORADF works better in comparison with the other three algorithms. The non-iterative AADF is very good

**Table 3** Performance of FORADF for very high-density mixed noise

Methods	Noise type	Noise %	$\lambda$	NI	PSNR	SSIM
FORADF	GN + S&P	0.1 + 90	1	5	16.338	0.8190
FORADF	GN + S&P	0.1 + 90	1	25	16.825	0.8190

in the removal of salt and pepper noise, but it fails to work even in the presence of low-density Gaussian noise; in addition, it is not a good smoother in comparison with FORADF and is computationally complex.

Figure 5a–e show the performance of the RF ( $\lambda = 0.25$ ,  $n = 5$ ), AMD ( $\lambda = 0.25$ ,  $n = 5$ ), AADF ( $\lambda = 0.25$ ,  $n = 1$ ), FORADF ( $\lambda = 0.25$ ,  $n = 5$ ), and FORADF ( $\lambda = 1$ ,  $n = 5$ ), respectively, for the three images corrupted by mixed noise consisting of Gaussian noise of variance 0.1 plus salt and pepper noise of noise density 70%. The results show that FORADF performs well at high noise density with low number of iterations. The performance of RF and AMD are very poor at high noise density with low number of iterations. Even with the increased number of iterations, the performance of RF [5] and AMD [4] at high noise density are not appreciable. AADF is also not suitable in the presence of even mild Gaussian noise.

Figure 6a and b show the results of FORADF for the image corrupted by mixed noise consisting of Gaussian noise of variance 0.1 and salt and pepper noise density 90% for  $n = 5$  and  $n = 25$ , respectively. These are the worst case results which demonstrate the intrinsic robustness of FORADF.

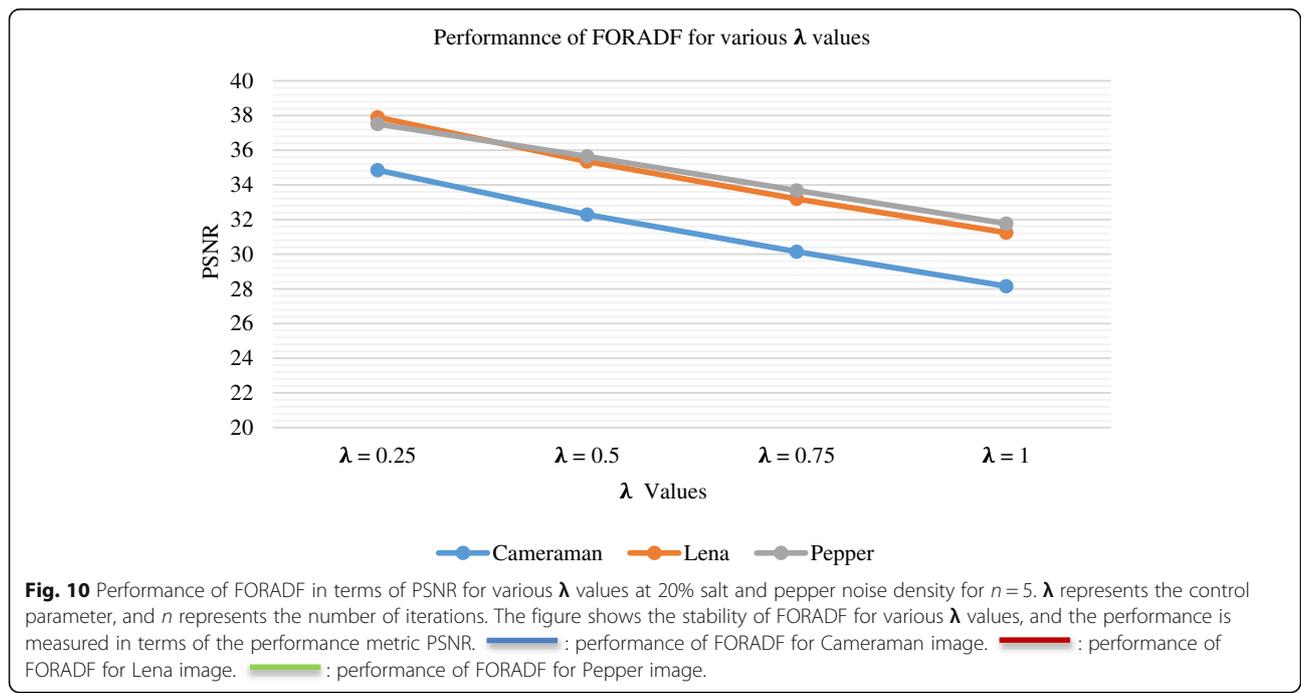
Figure 7a–d show the performance of FORADF without the preprocessing strategy and Fig. 7e–h show the performance of FORADF with the preprocessing strategy. It is clear from Fig. 7e and a that FORADF is a good image smoother with and without the preprocessing strategy. Figure 7b and c show that FORADF is also a good edge-preserving impulse noise suppresser at low and medium impulse noise densities even without the preprocessing strategy. The preprocessing strategy is an

**Table 4** Edge Preservation Index (EPI)

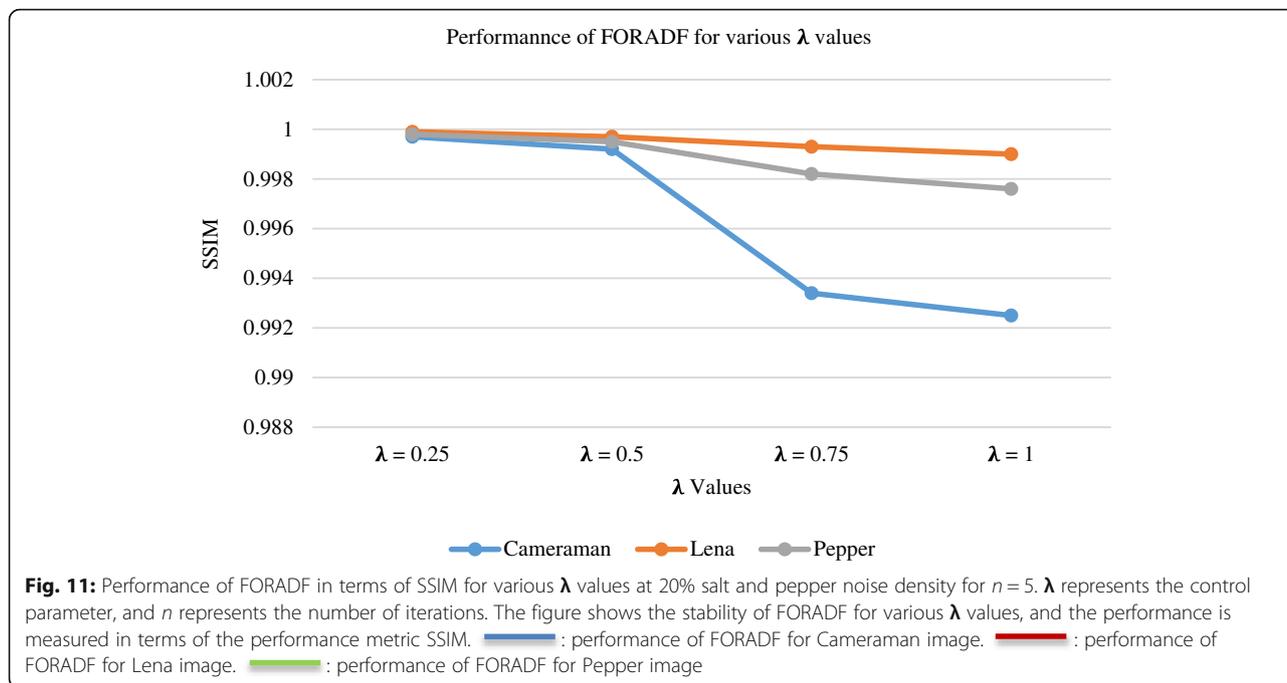
Methods	Test image	NI	Noise type – S&P	Noise type – S&P	Noise type-GN + S&P <sup>3</sup>	Noise type-GN + S&P <sup>3</sup>
			Noise % – 20	Noise % – 70	Noise % – 0.1 + 20	Noise % – 0.1 + 70
			EPI	EPI	EPI	EPI
RF $\lambda = 0.25$	Cameraman	5	0.4003	0.3288	0.3696	0.3316
	Lena	5	0.3560	0.3565	0.3468	0.3457
	Pepper	5	0.3551	0.3850	0.3520	0.3861
AMD $\lambda = 0.25$	Cameraman	5	0.4672	0.3485	0.4040	0.3363
	Lena	5	0.4038	0.2490	0.3793	0.2487
	Pepper	5	0.4482	0.2705	0.4173	0.2424
AADF $\lambda = 0.25$	Cameraman	1	0.8831	0.5740	0.3672	0.3356
	Lena	1	0.9081	0.5811	0.4220	0.4143
	Pepper	1	0.8679	0.5313	0.4134	0.4209
FORADF $\lambda = 0.25$	Cameraman	5	0.8495	0.5391	0.4064	0.3297
	Lena	5	0.8071	0.4870	0.4340	0.3243
	Pepper	5	0.7626	0.4592	0.3639	0.3179

addition for the purpose of increasing FORADF efficiency at higher impulse noise densities. The added preprocessing strategy does not significantly increase the computational complexity and, hence, is retained at all noise densities. Figure 7h and d demonstrate a comparison of FORADF performance with and without the preprocessing strategy at higher impulse noise density. In a practical situation, robustness implies up to 20% impulse noise, and, hence, the proposed algorithm is an excellent robust smoother even without the preprocessing strategy and is attractive for computational photography [27, 28].

Figure 8 demonstrates the intrinsic robustness of FORADF in comparison with the foundational robust diffusion algorithms RF and AMD. Figure 8d shows the excellent intrinsic robustness characteristic in addition to the anisotropic smoothing of FORADF at a very low number of iterations. Figure 9 demonstrates the anisotropic smoothing property of FORADF in comparison with the foundational PM algorithm. It is clearly seen from Fig. 9b and c that anisotropic smoothing behavior of FORADF is superior in terms of smoothing and edge preservation in comparison with PM algorithm.



**Fig. 10** Performance of FORADF in terms of PSNR for various  $\lambda$  values at 20% salt and pepper noise density for  $n = 5$ .  $\lambda$  represents the control parameter, and  $n$  represents the number of iterations. The figure shows the stability of FORADF for various  $\lambda$  values, and the performance is measured in terms of the performance metric PSNR. — : performance of FORADF for Cameraman image. — : performance of FORADF for Lena image. — : performance of FORADF for Pepper image.



The peak signal to noise ratio (PSNR) and the structural similarity index metric (SSIM), listed in Tables 1, 2, and 3, clearly demonstrate the superiority of FORADF in terms of noise removal and edge-preserving image smoothing. Even though AADF shows slightly better performance than FORADF in terms of PSNR and SSIM for 20% and 70% salt and pepper noise densities (Table 2), AADF fails to perform even in the presence of low-density Gaussian noise and AADF is not a smoother; in addition, AADF, which is non-iterative requiring one-shot solution, is computationally complex. Table 4 shows the performance of various filters in terms of edge preservation index (EPI). Figures 10 and 11 show the performance of FORADF for λ values 0.25, 0.5, 0.75, and 1 in terms of the performance metrics PSNR and SSIM, respectively. The results are obtained for the input images corrupted by 20% salt and pepper noise density, and the number of iterations is chosen as

n = 5. It is clear from Figs. 10 and 11 that FORADF provides very good performance in terms of intrinsic robustness, smoothing, and edge preservation over all λ values in the interval (0, 1]. In real-time DSP, which does not always mean high speed [29], multipliers and adders determine the computational efficiency [30] and hardware complexity [31] which is an important consideration in meeting power budgets [29]. A typical comparison from this perspective is presented in Table 5. Computational efficiency of FORADF in comparison with RF and AMD is expressed in terms of percentage excess/savings on arithmetic complexity (C) where C<sub>ref</sub>, C<sub>AMD</sub>, and C<sub>RF</sub> represent the complexity of the reference FORADF, AMD, and RF, respectively, as the number of arithmetic operations per pixel per iteration. In RF, 41.37% excess arithmetic computations are required in comparison with FORADF to perform one pixel operation. Similarly, 75.36% excess computations are

**Table 5** Hardware complexity in DSP context

Methods	Noise type	Noise %	λ	NIAP	No. of A/M/S per iteration (C)	Running time per iteration in seconds	Running time for NIAP in seconds*	Computational efficiency of FORADF (C <sub>ref</sub> ) in comparison with RF and AMD in terms of excess/savings on C (%) ((C - C <sub>ref</sub> )/C)
RF	GN + S&P	0.1 + 70	0.25	100	11/18/0	3.5244	352.44	((C <sub>RF</sub> - C <sub>ref</sub> )/C <sub>RF</sub> )= 41.37% excess
AMD	GN + S&P	0.1 + 70	0.25	100	12/33/24	5.5495	554.95	((C <sub>AMD</sub> - C <sub>ref</sub> )/C <sub>AMD</sub> )= 75.36% excess
FORADF	GN + S&P	0.1 + 70	1	5	5/4/8	4.9407	24.7038	((C <sub>ref</sub> - C <sub>ref</sub> )/C <sub>ref</sub> )= 0% excess

A/M/S additions/multiplications/sorting, NIAP number of iterations required for acceptable performance, C arithmetic complexity  
 \*MATLAB R2014a, 567u7 Intel (R) core (TM) i5-5200 U CPU @ 2.20 GHz, 64-bit Operating System, RAM-4.00 GB

required in AMD in comparison with FORADF to perform one pixel operation. The total number of multipliers and adders is the lowest for FORADF; in the context of DSP, this spirals down to power efficiency, since multipliers are power guzzlers [31]. The lowest number of iterations of FORADF compensates for the increased running time due to sorting. Therefore, the total energy consumed by FORADF is lower than that of AMD and is comparable with that of RF. The results clearly demonstrate that FORADF is an efficient edge-preserving robust smoothing in comparison with the four filters for Gaussian, salt and pepper, and mixed noise situations across all noise densities, and, hence, FORADF is significant in the context of power-efficient implementations.

#### 4 Conclusion

Diffusion smoothing of images with edge preservation in robust environment is a growing area of research. Existing robust diffusion smoothing filters employ two or more stages and are generally complex in terms of arithmetic operations. A low arithmetic complexity model and an algorithm for robust diffusion smoothing of images are derived. The algorithm exhibits superior performance in terms of standard performance criteria and visual results in Gaussian, impulse, and mixed noise situations. The intrinsic computational simplicity of the proposed model and algorithm are significant in the context of power efficient implementations.

#### Abbreviations

AADF: Adaptive anisotropic diffusion filter; AMD: Anisotropic median diffusion; CFL: Courant-Friedrichs-Lewy; DSP: Digital signal processing; ENI: Edge pixels, noisy pixels, and image pixels; EPI: Edge preservation index; FORADF: First-order robust anisotropic diffusion filter; GN: Gaussian noise; LHS: Left hand side; ML: Maximum likelihood; NI: Number of iteration; NIAP: Number of Iterations required for Acceptable Performance; PDE: Partial differential equation; PM: Perona-Maillik algorithm; PSNR: Peak signal to noise ratio; RF: Robust filter; RHS: Right hand side; S&P: Salt and pepper noise; SSIM: Structural similarity index metric

#### Acknowledgements

Not Applicable.

#### Funding

This work was not supported by any funding.

#### Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

#### Authors' contributions

DE contributed to the conceptual and theoretical formulations. NRR contributed to the algorithm development, programming, and validation. RS contributed to the computing platforms and methodology. All authors read and approved the final manuscript.

#### Authors' information

Resmi R. Nair received AMIE degree in Electronics and Communication Engineering from The Institution of Engineers (India) in 2007 and she received M.E degree in VLSI Design from Anna University in 2009. She is a research scholar of Anna University, Chennai, India. Her research interests

include Digital Signal Processing, Digital Image Processing, and VLSI. She is a member of IEEE, IETE, and ISTE (India).

Ebenezer David obtained his Ph. D degree in communication engineering from Anna University, Chennai, India in 1994. He is a Professor Emeritus in the Department of Electronics and Communication Engineering, College of Engineering, Anna University, Chennai. His research publications are in the area of nonlinear digital filtering with well over 45 journal and conference publications and 1200 citations. He has also delivered keynote speeches in international conferences. He served as a referee for journal of Medical Engineering and physics (U.K), IET, IEEE Transactions on Image Processing, and JEI (U.S.A). He is an elected AMIE (India), a member of ISTE (India), and a senior member IEEE.

Sivakumar Rajagopal is a Professor and Head of Department of Electronics and Communication Engineering at RMK Engineering College, Tamil Nadu, India. He has been teaching in the Electronics and Communication field since 1997. He obtained his Master's degree and Ph. D from College of Engineering Guindy, Anna University, Chennai. His research interests include Bio Signal Processing, Medical Image Processing, wireless body sensor networks and VLSI. He has published over 34 journal and 42 conference papers. He has chaired a number of international conferences and has delivered keynote speeches Dr. Siva is a life member of the Indian Society of Technical Education, a senior member of IEEE.

#### Competing interests

The authors declare that they have no competing interests.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

#### Author details

<sup>1</sup>Department of Electronics and Communication Engineering, RMK Engineering College, Kavaripeetai, Tamil Nadu 601206, India. <sup>2</sup>Department of Electronics and Communication Engineering, Anna University, Chennai, Tamil Nadu 600025, India.

Received: 1 July 2017 Accepted: 15 February 2019

Published online: 27 February 2019

#### References

1. J. Babaud, A. Witkin, M. Baudin, R. Duda, Uniqueness of the Gaussian kernel for scale space filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 26–33 (1986)
2. P. Perona, J. Malik, Scale space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1990)
3. M.J. Black, D.H. Marimont, Robust anisotropic diffusion. *IEEE Trans. Image Process.* **7**(3), 421–432 (1998)
4. J. Ling, A.C. Bovik, Smoothing low-SNR molecular images via anisotropic median diffusion. *IEEE Trans. Med. Imag.* **12**(4), 377–383 (2002)
5. B. Ham, D. Min, K. Sohn, Robust scale space filter using second order partial differential equations. *IEEE Trans. Image Process.* **21**(9), 3937–3951 (2012)
6. J. Wu, C. Tang, PDE-based random-valued impulse noise removal based on new class of controlling functions. *IEEE Trans. Image Process.* **20**(9), 2428–2438 (2011)
7. W. Wang, P. Lu, *IEEE Proceedings of the 10th world congress on intelligent control and automation. Adaptive Switching Anisotropic Diffusion Model for Universal Noise Removal* (Beijing, 2012), pp. 4803–4808
8. X. Chen, C. Tang, X. Yan, Switching degenerate diffusion PDE filter based on impulse like probability for universal noise removal. *Elsevier Int. J. Electron. Commun.* **68**(9), 851–857 (2014)
9. N.U. Khan, K.V. Arya, M. Pattanaik, Edge preservation of impulse noise filtered images by improved anisotropic diffusion, *Springer Sci. + Bus. Media New York. Multimed. Tools Appl.* **73**(1), 573–597 (2014)
10. H. Tian, H. Cai, J. Lai, A novel diffusion system for impulse noise removal based on a robust diffusion tensor. *Elsevier Neurocomputing* **133**, 222–230 (2014)
11. S.I. Cho, S.J. Kang, H.S. Kim, Y.H. Kim, Dictionary-based anisotropic diffusion for noise reduction. *Elsevier Pattern Recogn. Lett.* **46**, 36–45 (2014)
12. T. Veerakumar, S. Esakkirajan, I. Vennila, Edge preservation adaptive anisotropic diffusion filter approach for the suppression of impulse noise in images. *Elsevier Int. J. Electron. Commun.* **68**(5), 442–452 (2014)

13. B. Marami, B. Scherrer, O. Afacan, B. Erem, S.K. Warfield, A. Gholipour, Motion-robust diffusion-weighted brain MRI reconstruction through slice-level registration-based motion tracking. *IEEE Trans. Med. Imag.* **35**(10), 2258–2269 (2016)
14. A. Einstein, *Investigation on the Theory of the Brownian Movement*, ed. By R. Furth and Transl. by AD Cowper (Dover Publications, New York, 1956), pp. 9–15
15. M.H. Hayes, *Statistical Digital Signal Processing and Modelling* (Wiley, Wiley-India, New Delhi, 2011), pp. 99–102
16. K. Kim, G. Shevlyakov, Why Gaussianity [an attempt to explain this phenomenon]. *IEEE Signal Process. Mag.* **25**(2), 112 (2008)
17. J. Astola, P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering* (CRC Press, New York, 1997), p. 40
18. P.J. Huber, *Robust Statistics*, 2nd edn. (Wiley, New Jersey, 2009), p. 46
19. S. Rakshit, A. Ghosh, B.U. Shankar, Fast mean filtering technique (FMFT). *Elsevier Pattern Recogn.* **40**(3), 890–897 (2007)
20. J. Weickert, *Anisotropic Diffusion in Image Processing* (BG Teubner, Stuttgart, 2008), pp. 1–26
21. S. Paris, P. Kornprobst, J. Tumblin, F. Durand, *Bilateral Filtering* (Theory and Applications, (now Publishers, USA, 2009), pp. 5–9
22. K. He, J. Sun, X. Tang, Guided image filtering. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(6), 1397–1409 (2013)
23. L. Caraffa, J.P. Tarel, P. Charbonnier, The guided bilateral filter: when the joint/cross bilateral filter becomes robust. *IEEE Trans. Image Process.* **24**(4), 1199–1208 (2015)
24. J. Canny, A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **8**, 679–698 (1986)
25. R. Courant, K. Friedrichs, H. Lewy, Über die partiellen Differenzgleichungen der mathematischen Physik. *Math. Ann.* **100**, 32–74 (1928) (Trans.: in L. English by PD Lax, *Hyperbolic difference equations: a review of the courant-friedrichs-lewy paper in the light of recent developments*, *IBM J. of Res. and Develop.*, **11**(2), 235–238(1967)
26. K.S. Srinivasan, D. Ebenezer, A new fast and efficient decision-based algorithm for removal of high-density impulse noises. *IEEE Signal Process. Lett.* **14**(3), 189–192 (2007)
27. R. Ramanath, W.E. Snyder, Y. Yoo, M.S. Drew, Color image processing pipeline. *IEEE Signal Process. Mag.* **22**(1), 34–43 (2005)
28. W. van Houten, Z. Geradts, Using anisotropic diffusion for efficient extraction of sensor noise in camera identification. *J. Forensic Sci.* **57**(2), 521–527 (2012)
29. P. Deodar, *Dissertation, University of Cincinnati* (2014)
30. H.K. Garg, *Digital Signal Processing Algorithms: Number Theory, Convolution, Fast Fourier Transforms, and Application* (CRC Press, Florida, 1998), p. 1
31. K.K. Parhi, *VLSI Digital Signal Processing Systems: Design and Implementation* (Wiley, New York, 1997), p. 268 p. 645, p. xvi

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)

---