Circuits, Systems, and Signal Processing (2020) 39:363–390
https://doi.org/10.1007/s00034-019-01186-y

Generalized Fractional Filter-Based Algorithm for Image Denoising

Anil K. Shukla[1] · Rajesh K. Pandey[1] · Swati Yadav[1] · Ram Bilas Pachori[2]

Received: 28 July 2018 / Revised: 25 June 2019 / Accepted: 26 June 2019 / Published online: 13 July 2019

Abstract

This paper presents a new algorithm for image denoising using a fractional integral mask of the K-operator. K-operator is the generalized fractional operator, and it reduces to Riemann–Liouville and Caputo fractional derivatives in a special case. The proposed algorithm is applied to digital images of different nature to demonstrate the performance of image denoising. Experimental results are compared with other existing filters together with block matching and 3-D filtering, and weighted nuclear norm minimization-based approaches. The obtained experimental results show that the proposed algorithm is computationally efficient and its average performance is comparatively better than other discussed methods.

Keywords Fractional calculus · Difference equations · Interpolation · Texture · Image denoising

1 Introduction

The unwanted signals come in the image during the process of acquisition and transmission. These signals appear due to causes like inference in the electronic devices, light intensity, temperature, and vibration in the sensors. Noise deforms the image

✉ Rajesh K. Pandey
rkpandey.mat@iitbhu.ac.in

Anil K. Shukla
anil1shukla2@gmail.com

Swati Yadav
swatiy.rs.mat16@itbhu.ac.in

Ram Bilas Pachori
pachori@iiti.ac.in

[1] Department of Mathematical Sciences, Indian Institute of Technology (BHU), Varanasi 221005, India

[2] Discipline of Electrical Engineering, Indian Institute of Technology Indore, Indore, India

and creates a problem in further processing such as segmentation, object detection, and classification of the image. These issues made the problem of image denoising an essential task for researchers. In literature, several methods of image denoising based on the models [44], patch [32], convolution neural network [27], Retinex [1,30], spatial filtering [1,30,57], and sparse filtering [10] have been developed. In 2016, Ghimpeteanu et al. [15] used a decomposition model and processed the components in a moving frame to denoise the image. This method was useful to preserve local geometry. Some nonlocal mean-based image denoising methods are discussed in [13,53,55]. A new image denoising scheme by exploring the internal and external correlation together with two-stage filtering was proposed in [58]. Panetta et al. [40] introduced the concept of sequence to sequence similarity (SSS) and applied it for image denoising. An algorithm based on hyperbolic wavelet transformed is discussed in [12]. This method recovers ultrasound images and preserves tissue details. A weighted nuclear norm minimization (WNNM)-based denoising method was proposed in [17]. Further, it was extended in [34] by applying the stopping criteria based on Pearson's correlation coefficients during iteration. Some variational approach-based image denoising algorithms are discussed in [6,26,47,59]. Some other methods of image denoising are discussed in [5,51]. In 2016, Yang et al. [56] presented a review paper on the applications of fractional calculus in various fields of image processing. Fractional calculus plays an important role in various fields like viscoelasticity [3], continuum mechanics [7], fluid mechanics [11], thermodynamics [41], bioengineering [35], differential and integral calculus [33,36,45], boundary value problems [50], control system [14], signal processing [37,48,52], image enhancement [9,49], signal classification [29], and image denoising [8,43].

Use of fractional calculus in image denoising problems are now recognized by many researchers in recent years [4,30]. In 2010, Pu et al. [42] designed six fractional differential masks for enhancement of texture-rich images and concluded that the second differential mask is best among them. In 2011, a novel image denoising algorithm based on Riemann–Liouville definition of the fractional operator was proposed in [25]. Experimental results show that the algorithm is better than the Gaussian smoothing filter for a small noise variance. Guo et al. [18] proposed a fractional mask for image denoising with the help of Grunwald Letnikov definition of a fractional operator. It is further upgraded in [24] by modifying its coefficients. Simulation results show that the denoising performance of mask results better for texture-rich images. Jalab et al. [28] proposed two new filters named Alexander fractional differential (AFD) and Alexander fractional integral (AFI) with the help of fractional Alexander polynomials. Experimental results show that these filters remove noise effectively in the image. Recently, Li et al. proposed an adaptive image denoising algorithm based on small probability strategy and fractional calculus.

Motivated by all these works, we derived a new image denoising algorithm based on the newly introduced fractional operator named as K-operator [2] using Riesz kernel. Riemann–Liouville, Caputo, Riesz Riemann Liouville, and Riesz Caputo operators are the particular case of K-operator. The advantage of the K-operator over other known operator is the parameters p and q with the help of which we can adjust the weight of fractional mask according to the necessity to improve the performance of image denoising. For the recent applications of the K-operator, we refer the readers to

[38,39,46]. We compared the experimental results of the proposed method with other existing methods [10,18,25,28] on database [19] and some standard images from the literature. Based on experimental results such as peak signal to noise ratio (PSNR), mean square error (MSE), structure similarity index measure (SSIM) and visual perception, we demonstrate that the proposed filter is better than the Gaussian smoothing filter (GF), average filter (AF), median filter (MF), Grunwald Letnikov filter (GLF), Riemann Liouville filter (RLF), improved Grunwald Letnikov filter (IGLF), Alexander fractional differential (AFD), Alexander fractional integral (AFI) filter, block matching and 3-D (BM3D) filtering [10], and weighted nuclear norm minimization (WNNM)-based image denoising method. In our experiments, fractional order for the proposed integral mask is chosen manually and varies with the noise variance.

The remainder of this paper is organized as follows. In Sect. 2, we have discussed the theory related to K-operator. The denoising algorithm based on K-operator is derived in Sect. 3. Experimental results and computational cost of the proposed method are presented in Sect. 4 and conclusions are drawn in Sect. 5.

2 Mathematical Analysis

The K-operator [2] of a regular function f for order α is defined as,

$$(K_P^{\alpha} f)(x) = p \int_a^x k_{\alpha}(x,\xi) f(\xi) \mathrm{d}\xi + q \int_x^b k_{\alpha}(\xi,x) f(\xi) \mathrm{d}\xi \tag{1}$$

where $a < x < b$, α is a positive real number and $P = \{a, x, b, p, q\}$ is a parameter set. We extend the K-operator in n-dimension for a regular function f on Euclidean space $\mathbb{R}^n$ with compact support $[a,b]^n$ as,

$$\begin{aligned}(K_{P_i}^{\alpha} f)(x) = p \int_a^b \ldots \int_a^{x_i} \ldots \int_a^b k_{\alpha}(x,\xi) f(\xi) \mathrm{d}\xi \\ + q \int_a^b \ldots \int_{x_i}^b \ldots \int_a^b k_{\alpha}(\xi,x) f(\xi) \mathrm{d}\xi, \qquad x,\xi \in \mathbb{R}^n,\end{aligned} \tag{2}$$

The kernel $k_{\alpha}(x,\xi)$ may be dependent on parameter α. Parameters p and q are any two real numbers. Here limit of integral can be extended to $\mathbb{R}^n$. Let us define the kernel as follows:

$$k_{\alpha}(x,\xi) = \frac{1}{\gamma_n(\alpha)} \begin{cases} |x-\xi|^{(\alpha-n)} & \text{for } \alpha - n \neq 0, 2, 4, \ldots \\ |x-\xi|^{(\alpha-n)} \log\left(\frac{1}{|x-\xi|}\right) & \text{for } \alpha - n = 0, 2, 4, \ldots \end{cases}. \tag{3}$$

Here, $|x-\xi|$ denotes the Euclidean distance between x and ξ, parameter $\gamma_n(\alpha)$ is defined as,

$$\gamma_n(\alpha) = \begin{cases} 2^{\alpha} \pi^{\frac{n}{2}} \left(\Gamma \frac{(n-\alpha)}{2}\right)^{-1} & \text{for } \alpha - n \neq 0, 2, 4, \ldots \\ (-1)^{\frac{(n-\alpha)}{2}} 2^{(\alpha-1)} \pi^{\frac{n}{2}} \Gamma\left(1 + \frac{\alpha-n}{2}\right) \Gamma \frac{\alpha}{2} & \text{for } \alpha - n = 0, 2, 4, \ldots \end{cases}, \tag{4}$$

the kernel defined in Eq. (3) is called Riesz kernel [33]. On taking $p = 1, q = 0$ in Eq. (1) with kernel defined in Eq. (3), operator$(K^{\alpha}_{P_i} f)(x)$ reduces to Riesz-integral operator [33]. For $\alpha \in (0, 2)$, it follows

$$\begin{aligned}(K^{\alpha}_{P_i} f)(x) = {}& \frac{p}{\gamma_n(\alpha)} \int_a^b \ldots \int_a^{x_i} \ldots \int_a^b |x-\xi|^{(\alpha-n)} f(\xi)\mathrm{d}\xi \\ &+ \frac{q}{\gamma_n(\alpha)} \int_a^b \ldots \int_{x_i}^b \ldots \int_a^b |x-\xi|^{(\alpha-n)} f(\xi)\mathrm{d}\xi, \qquad (5)\end{aligned}$$

without loss of generality, on taking $n = 1$, $a = 0$, using the symmetric property of convolution in the first part and replacing ξ by $\xi + x$ in the second part of integral, Eq. (5) reduces to the following:

$$(K^{\alpha}_{P} f)(x) = \frac{p}{\gamma_1(\alpha)} \int_0^x |\xi|^{(\alpha-1)} f(x-\xi)\mathrm{d}\xi + \frac{q}{\gamma_1(\alpha)} \int_0^{b-x} |\xi|^{(\alpha-1)} f(x+\xi)\mathrm{d}\xi. \qquad (6)$$

This is one dimensional K-operator with Riesz kernel for $\alpha \in (0, 2)$ with parameter set $P = \{0, x, b, p, q\}$.

3 Design of Fractional Integral Mask

To design the fractional integral mask applicable for digital signals, we replace the integration of Eq. (6) by the summation of products. For sufficiently large N and M, divide the interval $[0, x]$ and $[0, b-x]$ into N and M equal parts, respectively, as follows:

$$\begin{aligned}(K^{\alpha}_{P} f)(x) = {}& \frac{p}{\gamma_1(\alpha)} \sum_{k=0}^{N-1} \int_{\frac{kx}{N}}^{\frac{(k+1)x}{N}} |\xi|^{(\alpha-1)} f(x-\xi)\mathrm{d}\xi \\ &+ \frac{q}{\gamma_1(\alpha)} \sum_{L=0}^{M-1} \int_{\frac{Lx}{M}}^{\frac{(L+1)(b-x)}{M}} |\xi|^{(\alpha-1)} f(x+\xi)\mathrm{d}\xi. \qquad (7)\end{aligned}$$

Now, approximating f by average value on each sub-interval and replacing $|\xi|$ by ξ in the above-mentioned integral as ξ is nonnegative on the given interval, we have following:

$$\begin{aligned}(K^{\alpha}_{P} f)(x) \cong {}& \frac{p}{\gamma_1(\alpha)} \sum_{k=0}^{N-1} \frac{1}{2}\left[f\left(x-\frac{kx}{N}\right) + f\left(x-\frac{(k+1)x}{N}\right)\right] \int_{\frac{kx}{N}}^{\frac{(k+1)x}{N}} \xi^{(\alpha-1)}\mathrm{d}\xi \\ &+ \frac{q}{\gamma_1(\alpha)} \sum_{L=0}^{M-1} \frac{1}{2}\left[f\left(x+\frac{L(b-x)}{M}\right)\right. \\ &\left. + f\left(x+\frac{(L+1)(b-x)}{M}\right)\right] \int_{\frac{Lx}{M}}^{\frac{(L+1)(b-x)}{M}} \xi^{(\alpha-1)}\mathrm{d}\xi\end{aligned}$$

$$\begin{aligned}
&\cong \frac{p}{2\alpha\gamma_1(\alpha)}\left(\frac{x}{N}\right)^{\alpha}\sum_{k=0}^{N-1}\left[f\left(x-\frac{kx}{N}\right)\right.\\
&\left.+f\left(x-\frac{(k+1)x}{N}\right)\right]\left[(k+1)^{\alpha}-k^{\alpha}\right]\\
&+\frac{q}{2\alpha\gamma_1(\alpha)}\left(\frac{b-x}{M}\right)^{\alpha}\sum_{L=0}^{M-1}\left[f\left(x+\frac{L(b-x)}{M}\right)\right.\\
&\left.+f\left(x+\frac{(L+1)(b-x)}{M}\right)\right]\left[(L+1)^{\alpha}-L^{\alpha}\right].
\end{aligned} \tag{8}$$

Equation (8) is the derivation of a numerical algorithm of fractional integral for one dimensional signal. In digital images, the number of pixels are finite and minimum length of step-size can be assumed as one pixel that is N and M can be large enough such that $\frac{x}{N}=1$, and $\frac{b-x}{M}=1$. Thus the numerical algorithm of fractional integral for two-dimensional image signal in X-direction is given by,

$$\begin{aligned}
K_P^{\alpha}f(x,y)&=\frac{p}{\gamma_1(\alpha)}\int_0^x|\xi|^{(\alpha-1)}f(x-\xi,y)\mathrm{d}\xi\\
&\quad+\frac{q}{\gamma_1(\alpha)}\int_0^{b-x}|\xi|^{(\alpha-1)}f(x+\xi,y)\mathrm{d}\xi\\
&\cong\frac{p}{2\alpha\gamma_1(\alpha)}\left[\left(N^{\alpha}-(N-1)^{\alpha}\right)f\left(x-\frac{Nx}{N},y\right)\right.\\
&\quad\left.+\sum_{N-1}^{1}\left((K+1)^{\alpha}-(k-1)^{\alpha}\right)f\left(x-\frac{kx}{N},y\right)\right]+\frac{(p+q)}{2\alpha\gamma_1(\alpha)}f(x,y)\\
&\quad+\frac{q}{2\alpha\gamma_1(\alpha)}\left[\sum_{L=1}^{M-1}\left((L+1)^{\alpha}+(L-1)^{\alpha}\right)f\left(x+\frac{L(b-x)}{M},y\right)\right.\\
&\quad\left.+\frac{q}{2\alpha\gamma_1(\alpha)}\left(M^{\alpha}-(M-1)^{\alpha}\right)f\left(x+\frac{M(b-x)}{M},y\right)\right].
\end{aligned} \tag{9}$$

When $k=n\le(N-1)$ and $L=m\le(M-1)$, the interior $(m+n+1)$ difference approximation of fractional order integral is given by following:

$$\begin{aligned}
K_P^{\alpha}f(x,y)&=\frac{p}{\gamma_1(\alpha)}\int_0^x|\xi|^{(\alpha-1)}f(x-\xi,y)\mathrm{d}\xi\\
&\quad+\frac{q}{\gamma_1(\alpha)}\int_0^{b-x}|\xi|^{(\alpha-1)}f(x-\xi,y)\mathrm{d}\xi,\\
&\cong\frac{p}{2\alpha\gamma_1(\alpha)}\left[\left(n^{\alpha}-(n-1)^{\alpha}\right)f\left(x-\frac{nx}{n},y\right)\right.\\
&\quad\left.+\sum_{k=n-1}^{1}\left((k+1)^{\alpha}-(k-1)^{\alpha}\right)f\left(x-\frac{kx}{n},y\right)\right]
\end{aligned}$$

$$
\begin{aligned}
&+\frac{p+q}{2\alpha\gamma_1(\alpha)} f(x,y)\\
&+\frac{q}{2\alpha\gamma_1(\alpha)}\left[\sum_{L=1}^{m-1}\left((L+1)^{\alpha}-(L-1)^{\alpha}\right) f\left(x+\frac{L(b-x)}{m}, y\right)\right.\\
&\left.+\frac{q}{2\alpha\gamma_1(\alpha)}\left(m^{\alpha}-(m-1)^{\alpha}\right) f\left(x+\frac{m(b-x)}{m}, y\right)\right]. \qquad (10)
\end{aligned}
$$

Similarly, the numerical algorithms for a two-dimensional image signal in the Y-direction is given by the following expression:

$$
\begin{aligned}
K_P^{\alpha} f(x,y) &= \frac{p}{\gamma_1(\alpha)}\int_0^{y} |\xi|^{(\alpha-1)} f(x, y-\xi)\mathrm{d}\xi\\
&\quad+\frac{q}{\gamma_1(\alpha)}\int_0^{b-y} |\xi|^{(\alpha-1)} f(x, y-\xi)\mathrm{d}\xi\\
&\cong \frac{p}{2\alpha\gamma_1(\alpha)}\left[\left(n^{\alpha}-(n-1)^{\alpha}\right) f\left(x, y-\frac{ny}{n}\right)\right.\\
&\quad\left.+\sum_{k=n-1}^{1}\left((k+1)^{\alpha}-(k-1)^{\alpha}\right) f\left(x, y-\frac{ky}{n}\right)\right]\\
&\quad+\frac{p+q}{2\alpha\gamma_1(\alpha)} f(x,y)\\
&\quad+\frac{q}{2\alpha\gamma_1(\alpha)}\left[\sum_{L=1}^{m-1}\left((L+1)^{\alpha}-(L-1)^{\alpha}\right) f\left(x, y+\frac{L(b-y)}{m}\right)\right]\\
&\quad+\frac{q}{2\alpha\gamma_1(\alpha)}\left[\left(m^{\alpha}-(m-1)^{\alpha}\right) f\left(x, y+\frac{m(b-y)}{m}\right)\right]. \qquad (11)
\end{aligned}
$$

Thus, the coefficients of fractional integral masks are given as follows:

$$
\begin{cases}
c_{f_{-m}} = \frac{q}{2\alpha\gamma_1(\alpha)}\left(m^{\alpha}-(m-1)^{\alpha}\right)\\
c_{f_{-(m-1)}} = \frac{q}{2\alpha\gamma_1(\alpha)}\left(m^{\alpha}-(m-2)^{\alpha}\right)\\
c_{f_{-L}} = \frac{q}{2\alpha\gamma_1(\alpha)}\left((L+1)^{\alpha}-(L-1)^{\alpha}\right), \qquad 2 \le L \le (m-2)\\
c_{f_{-1}} = \frac{2^{\alpha} q}{2\alpha\gamma_1(\alpha)}\\
c_{f_0} = \frac{p+q}{2\alpha\gamma_1(\alpha)}\\
c_{f_1} = \frac{2^{\alpha} p}{2\alpha\gamma_1(\alpha)}\\
c_{f_k} = \frac{p}{2\alpha\gamma_1(\alpha)}\left((k+1)^{\alpha}-(k-1)^{\alpha}\right), \qquad 2 \le k \le (n-2)\\
c_{f_{(n-1)}} = \frac{p}{2\alpha\gamma_1(\alpha)}\left(n^{\alpha}-(n-2)^{\alpha}\right)\\
c_{f_n} = \frac{p}{2\alpha\gamma_1(\alpha)}\left(n^{\alpha}-(n-1)^{\alpha}\right).
\end{cases} \qquad (12)
$$

Table 1 The superposition of integral masks in all possible eight directions

$C_{f_n}+C_{f_{-n}}$	0	$\ldots$	0	$C_{f_n}+C_{f_{-n}}$	0	$\ldots$	0	$C_{f_n}+C_{f_{-n}}$
0	$\ddots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$	$\iddots$	0
$\vdots$	$\ddots$	$C_{f_2}+C_{f_{-2}}$	0	$C_{f_2}+C_{f_{-2}}$	0	$C_{f_2}+C_{f_{-2}}$	$\iddots$	$\vdots$
0	$\ldots$	0	$C_{f_1}+C_{f_{-1}}$	$C_{f_1}+C_{f_{-1}}$	$C_{f_1}+C_{f_{-1}}$	0	$\ldots$	0
$C_{f_n}+C_{f_{-n}}$	$\ldots$	$C_{f_2}+C_{f_{-2}}$	$C_{f_1}+C_{f_{-1}}$	$8C_{f_0}$	$C_{f_1}+C_{f_{-1}}$	$C_{f_2}+C_{f_{-2}}$	$\ldots$	$C_{f_n}+C_{f_{-n}}$
0	$\ldots$	0	$C_{f_1}+C_{f_{-1}}$	$C_{f_1}+C_{f_{-1}}$	$C_{f_1}+C_{f_{-1}}$	0	$\ldots$	0
$\vdots$	$\iddots$	$C_{f_2}+C_{f_{-2}}$	0	$C_{f_2}+C_{f_{-2}}$	0	$C_{f_2}+C_{f_{-2}}$	$\ddots$	$\vdots$
0	$\iddots$	$\iddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	0
$C_{f_n}+C_{f_{-n}}$	0	$\ldots$	0	$C_{f_n}+C_{f_{-n}}$	0	$\ldots$	0	$C_{f_n}+C_{f_{-n}}$

Table 2 The superposition of integral masks in all possible eight directions for $q = 0$

C_{f_n}	0	$\ldots$	0	C_{f_n}	0	$\ldots$	0	C_{f_n}
0	$\ddots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\iddots$	$\iddots$	0
$\vdots$	$\ddots$	C_{f_2}	0	C_{f_2}	0	C_{f_2}	$\iddots$	$\vdots$
0	$\ldots$	0	C_{f_1}	C_{f_1}	C_{f_1}	0	$\ldots$	0
C_{f_n}	$\ldots$	C_{f_2}	C_{f_1}	$8C_{f_0}$	C_{f_1}	C_{f_2}	$\ldots$	C_{f_n}
0	$\ldots$	0	C_{f_1}	C_{f_1}	C_{f_1}	0	$\ldots$	0
$\vdots$	$\iddots$	C_{f_2}	0	C_{f_2}	0	C_{f_2}	$\ddots$	$\vdots$
0	$\iddots$	$\iddots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\ddots$	0
C_{f_n}	0	$\ldots$	0	C_{f_n}	0	$\ldots$	0	C_{f_n}

Here coefficients $c_{f_{-m}}$, $c_{f_{-L}}(1 \leq L \leq (m-1))$ are the mask coefficients on noncausal pixels. Coefficient c_{f_0} is mask coefficient of the interested pixel and coefficients $c_{f_k}(1 \leq k \leq n)$ are the mask coefficients of the causal pixels. For $p = q$, the mask coefficients of noncausal pixels are same as that of the causal pixels. To make the mask symmetric about interest pixel, we take $m = n$ in Eq. (12). Thus, for given value of n, size of the mask will be $(2n + 1) \times (2n + 1)$. To make it isotropic, we apply it in all possible eight directions. We can increase or decrease the weight of the mask coefficients on causal as well as on noncausal pixel according to the necessity of the problem by varying the values of p and q. By taking $q = 0$, we make all the mask coefficients on a noncausal pixel to zero to work only with mask coefficients on causal pixels. All the eight integral masks corresponding to the above derivation can be superimposed to a single mask, that is given in Table 1.

Here, we intend to work only on the causal pixel and the corresponding integral mask can be obtained by taking $q = 0$. For the sake of simplicity and to reduce the number of iterations, we superimpose all eight fractional integral masks. The resultant mask obtained by superimposing all eight fractional masks is shown in Table 2. Flowchart of the proposed algorithm is shown in Fig. 1. The experimental results of the presented fractional mask will be discussed in the upcoming section.

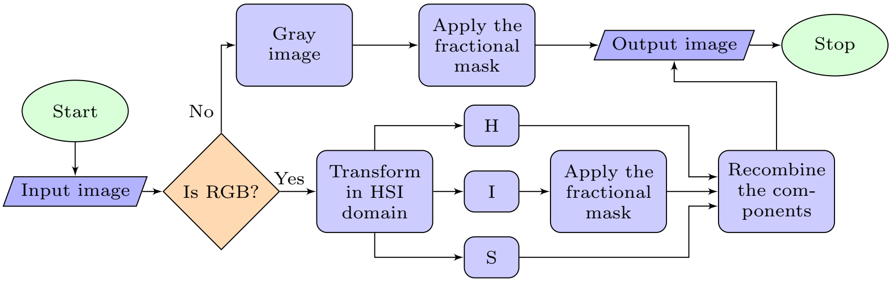

Fig. 1 Flowchart of the proposed algorithm

4 Experimental Results and Analysis

In this section, the designed fractional integral mask is applied on digital images to demonstrate the performance of the proposed method. Further, the section is divided into three subsections in which the first subsection presents the comparative results of the proposed method with other existing spatial domain methods, and the second subsection gives the same with transformed-based methods. Computation time is discussed in the last subsection. To implement the proposed algorithm, we normalize the filter mask by dividing each coefficient of mask given in Table 2 by the sum of all the coefficients of the mask. Convolution operation has been taken to denoise the gray-scale image. Corner pixels of images are filtered by zero padding. The RGB color image is denoised by transforming it in hue saturation intensity (HSI) color model and then filtering the I-component of the image. Experimental results are compared by visual perception and evaluation metrics such as MSE, PSNR, and SSIM.

The MSE [16] is defined as follows:

$$\text{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}[X'(x, y) - X(x, y)]^2,$$

where $X'(x, y)$ is the sampled image without noise and $X(x, y)$ is the denoised image. The 8-bit images are used in the experiment and PSNR [16] in terms of MSE is calculated using the following formula,

$$\text{PSNR} = 20\text{log}_{10}\left(\frac{2^8 - 1}{\sqrt{\text{MSE}}}\right).$$

The SSIM measure [54] can be expressed in terms of luminance, contrast, and structure by the following formula

$$\text{SSIM}(X, X') = [l(X, X')]^{\alpha}[c(X, X')]^{\beta}[s(X, X')]^{\gamma},$$

where,

$$l(X, X') = \frac{2\mu_x\mu_{x'} + c_1}{\mu_x^2 + \mu_{x'}^2 + c_1},$$

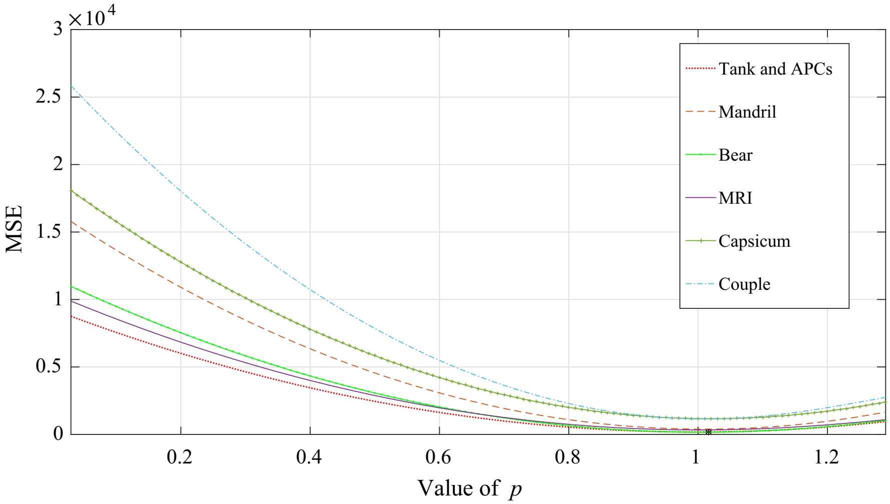

Fig. 2 Denoising performance of the proposed filter at different values of parameter p for images Truck and APCs, Mandrill, Bear, MRI, Capsicum, and Couple corrupted by the Gaussian noise of mean 0 and variance 0.02

$$c(X, X') = \frac{2\sigma_x\sigma_{x'} + c_2}{\sigma_x^2 + \sigma_{x'}^2 + c_2},$$
$$s(X, X') = \frac{\sigma_{xx'} + c_3}{\sigma_x\sigma_{x'} + c_3},$$

Here $\mu_x, \mu_{x'}, \sigma_x, \sigma_{x'}$ and $\sigma_{xx'}$ are the local means, standard deviations, and cross-covariance for image $X(x, y)$ and $X'(x, y)$. Constants α, β and γ are used to adjust the relative weights of luminance, contrast, and structure. For the simplicity, values of the constants α, β, γ are taken as $(\alpha = 1, \beta = 1, \gamma = 1)$.

From the graph plotted in Fig. 2, we demonstrate that the performance of the proposed algorithm is best at the parameter value ($p \approx$1.02).

4.1 Comparison with Traditional and Fractional Filters

This section presents the comparative study of the proposed method with other spatial domain methods. Results are evaluated based on metrics MSE, PSNR, SSIM, and the visual perception. The order of fractional derivative used in the fractional mask varies according to the noise variance and image features. The order of fractional mark is optimized manually. The fractional orders used to perform the experiment for the image in Fig. 3, corresponding to variances 0.001, 0.01, 0.02, 0.03, 0.04, 0.05 are 0.021, 0.0741, 1.061, 1.221, 1.301, 1.361, respectively. The PSNR, MSE and SSIM are optimum corresponding to these fractional orders in K-operator. Similarly for the image in Fig. 7, the corresponding fractional orders are 0.3611, 0.5810, 0.9410, 1.1010, 1.2010, 1.2210. Further, for the image in Fig. 11, corresponding fractional orders are 0.0210, 0.6010, 0.9010, 1.0210, 1.0410, 1.0610 and fractional orders for

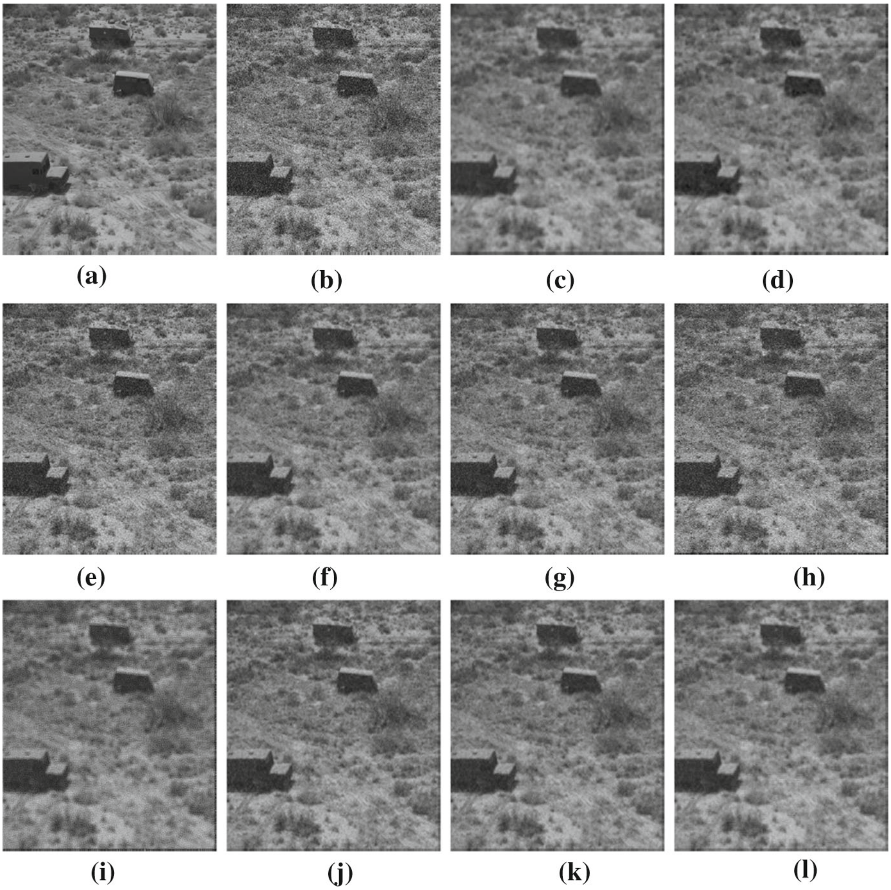

Fig. 3 Experimental results using proposed and existing filters with Gaussian noise, mean 0 and variance 0.02. Gray-scale image of size (512 × 512) pixels "Truck and APCs", where **a** sampled image. **b** Image with the Gaussian noise of mean 0 and variance 0.02. **c** Average filter. **d** Median filter. **e** Gaussian smoothing filter. **f** Riemann–Liouville filter. **g** Grunwald–Letnikov filter. **h** Improved Grunwald–Letnikov filter. **i** Alexander fractional differential filter. **j** Alexander fractional integral filter. **k** Proposed filter with $p = 1.0$. **l** Proposed filter with $p = 1.05$

image in Fig. 15 are 0.0210, 0.3410, 0.7210, 0.8210, 0.9010, 0.9610, respectively, for the proposed filter mask.

MSE versus variance graphs of Fig. 3 are plotted in Fig. 4 and its values are shown in Table 3. PSNR versus variance, and SSIM versus variance graphs of Fig. 3 are plotted in Figs. 5 and 7 and their values are shown in Tables 4 and 5, respectively. From all these plots and tabulated values, we observe that MSE of the image denoised by proposed method at $p = 1.05$ is less, PSNR is high, and SSIM is comparable to all other filters for small noise variance and higher for noise variance greater than 0.02. PSNR, MSE and SSIM (as shown in Fig. 6) of the image denoised by the proposed method at $p = 1.0$ is comparable to other discussed methods.

MSE versus variance graphs of Fig. 7 are plotted in Fig. 8, corresponding values are shown in Table 6. PSNR versus variance graphs are plotted in Fig. 9, corresponding

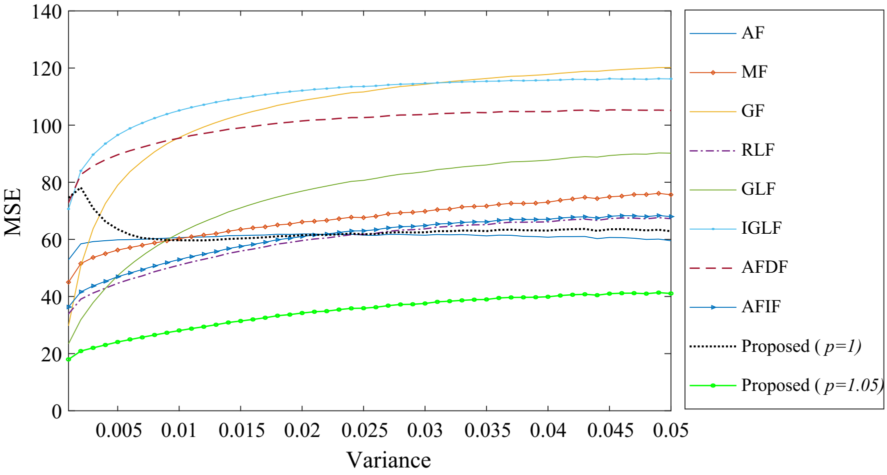

Fig. 4 MSE versus Variance comparison of different filters with the proposed filter of image in Fig. 3

Table 3 The values of MSE of image in Fig. 3 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	52.8657	45.0047	29.7611	33.9011	23.4295	70.6248	72.8547	36.3204	74.5322	17.9802
0.01	60.5865	60.2529	95.6668	50.9863	62.0974	105.1230	95.4234	52.9176	59.6748	28.1069
0.02	61.8956	66.0972	108.6169	59.5864	76.9620	112.0852	101.4778	61.0088	61.4724	34.2138
0.03	61.4700	69.8239	114.2958	63.6920	83.7128	114.6433	103.7226	64.8284	62.4779	37.5554
0.04	60.7683	73.0326	117.7588	66.1310	87.7387	115.7261	104.7131	67.0312	63.0872	39.9039
0.05	59.5246	75.6420	120.1731	67.2520	90.1967	116.1885	105.1440	67.9882	62.8511	41.0566

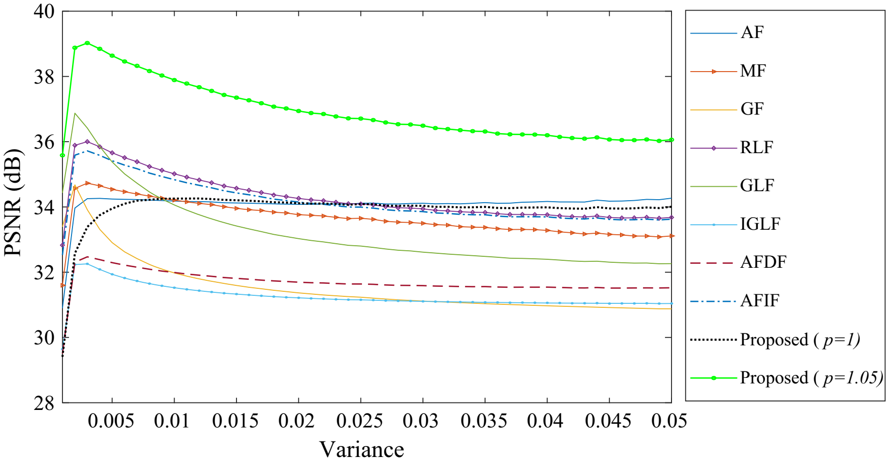

Fig. 5 PSNR versus variance comparison of different filters with proposed filter of image in Fig. 3

Table 4 The values of PSNR in dB of image in Fig. 3 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	30.8991	31.5982	33.3943	32.8287	34.4332	29.6412	29.5062	32.5293	29.4074	35.5831
0.01	34.1827	34.2092	31.9788	35.0157	34.0645	31.5238	31.9910	34.8362	34.2562	37.8896
0.02	34.0799	33.7630	31.3660	34.2634	33.0286	31.2143	31.6941	34.1496	34.1131	36.9408
0.03	34.1131	33.4983	31.1201	33.9417	32.6228	31.1054	31.5885	33.8564	34.0346	36.4908
0.04	34.1686	33.2815	30.9761	33.7605	32.3961	31.0601	31.5426	33.6952	33.9879	36.1983
0.05	34.2684	33.1121	30.8781	33.6793	32.2628	31.0408	31.5228	33.6268	34.0059	36.0608

Table 5 The values of SSIM of image in Fig. 3 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	0.5268	0.5529	0.8316	0.7629	0.8554	0.2676	0.1111	0.7297	0.1637	0.7231
0.01	0.5693	0.5667	0.4257	0.7395	0.6747	0.1320	0.0675	0.7078	0.5805	0.7190
0.02	0.5513	0.5342	0.2771	0.6535	0.5243	0.0879	0.0495	0.6262	0.5726	0.6574
0.03	0.5356	0.5092	0.2093	0.5907	0.4359	0.0676	0.0399	0.5667	0.5441	0.6086
0.04	0.5214	0.4883	0.1696	0.5408	0.3757	0.0564	0.0345	0.5194	0.5151	0.5690
0.05	0.5083	0.4699	0.1434	0.5009	0.3321	0.0488	0.0308	0.4818	0.4892	0.5363

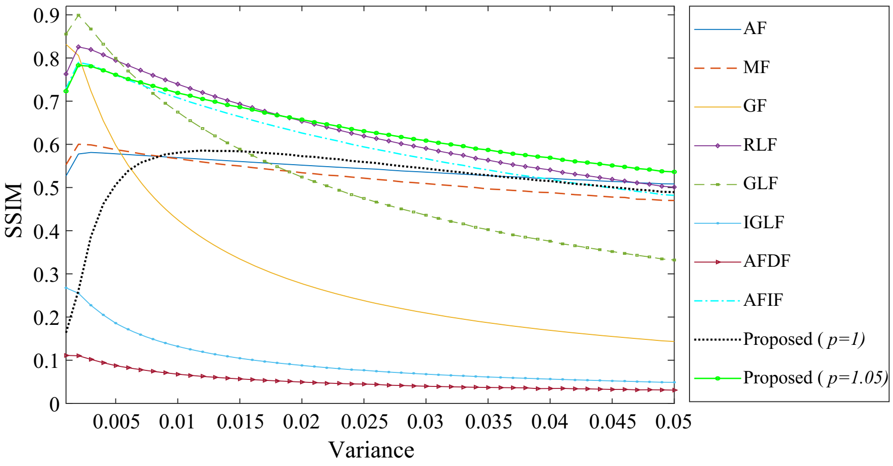

Fig. 6 SSIM versus Variance comparison of different filters with proposed filter of image in Fig. 3

values of PSNR are presented in Table 7 and Variance-SSIM graphs of Fig. 7 are plotted in Fig. 10, and corresponding values are given in Table 8. Visual perception of Fig. 7 shows that the proposed filter has better denoising performance and preserves texture and edge details correctly. MSE versus variance comparison graph of Fig. 7 and its tabulated values show that MSE of image denoised by the proposed filter at $p = 1.05$ is minimum among them. PSNR versus variance graphs of Fig. 7 and its

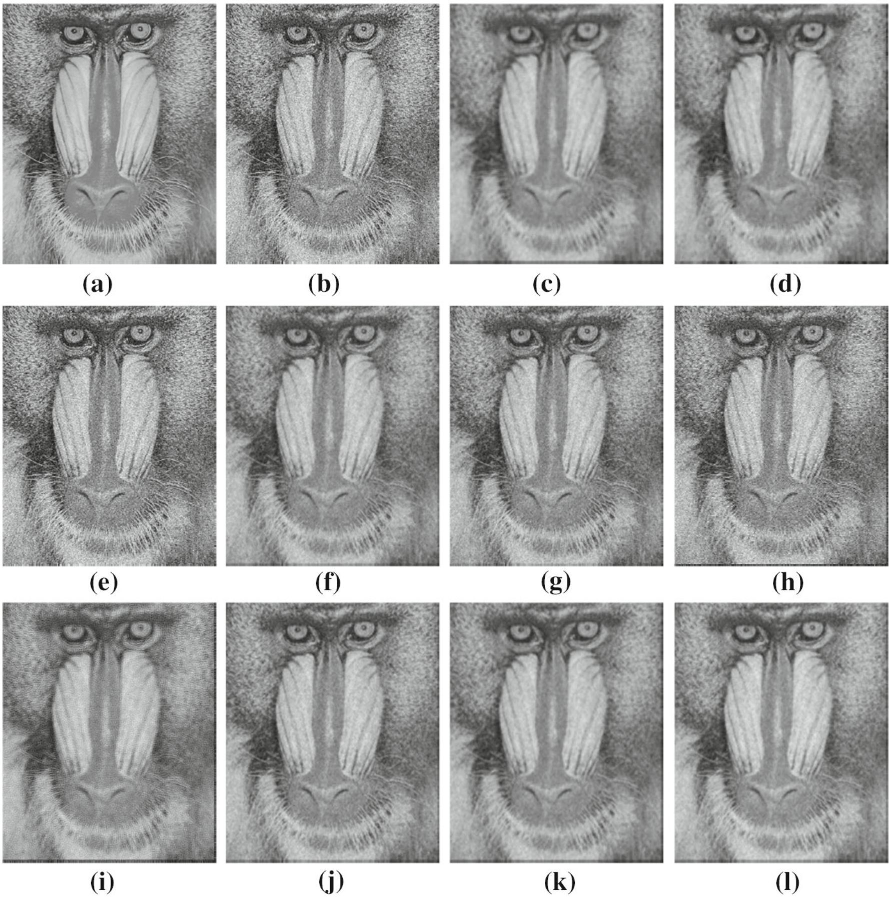

Fig. 7 Experimental results using proposed and existing filters with Gaussian noise, mean 0 and variance 0.02. Gray-scale image of size (512 × 512) pixels "Mandrill", where **a** sampled image. **b** Image with the Gaussian noise of mean 0 and variance 0.02. **c** Average filter. **d** Median filter. **e** Gaussian smoothing filter. **f** Riemann–Liouville filter. **g** Grunwald–Letnikov filter. **h** Improved Grunwald–Letnikov filter. **i** Alexander fractional differential filter. **j** Alexander fractional integral filter. **k** Proposed filter with $p = 1.0$. **l** Proposed filter with $p = 1.05$

corresponding tabulated values show that PSNR of image denoised by the proposed filter is higher than the other filters at $p = 1.05$, and comparable at $p = 1.0$ whereas SSIM versus variance graphs and its tabulated values show that SSIM of the image denoised by the proposed filter is higher than all other filters discussed in this paper. Based on simulation results, we demonstrate that the proposed filter has effective denoising performance.

As the components of a color image are highly correlated so direct implementation of a filter can distort the image. Thus, RGB color images are denoised by transforming it in HSI and then filtering I component only. The image denoised by different filters are shown in Fig. 10, and corresponding graphs of MSE, PSNR, and SSIM are shown in Figs. 12, 13, and 14, respectively. Further values of PSNR, MSE, and SSIM are

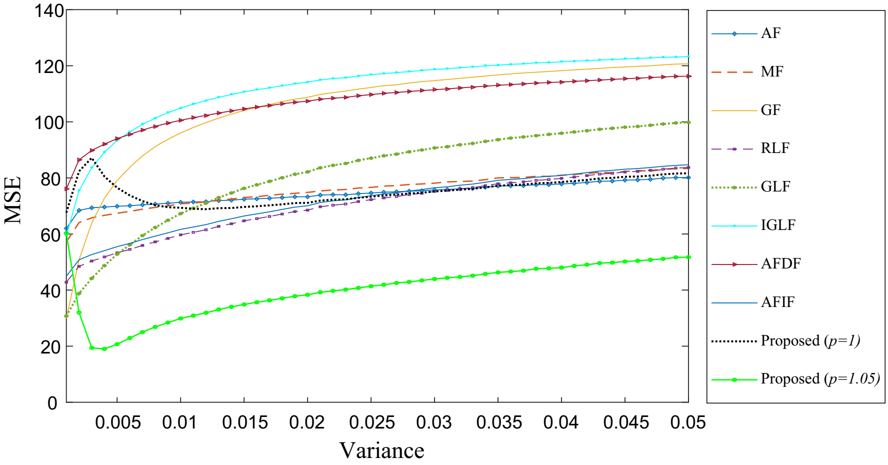

Fig. 8 MSE versus variance comparison of different filters with proposed filter of the image in Fig. 7

Table 6 The values of MSE of image in Fig. 7 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	61.9976	57.1629	29.8533	42.7905	30.7132	59.3785	76.0699	44.9575	67.5992	60.2675
0.01	71.2974	70.7865	96.1016	59.7005	67.2753	104.8547	100.5386	61.6448	69.3623	29.8822
0.02	73.2623	74.8240	108.7321	68.5139	82.1341	114.1143	107.3646	70.0642	71.1080	38.3160
0.03	75.6972	78.1368	114.7544	75.2007	90.7055	118.6877	111.5180	76.4932	75.1746	43.9693
0.04	77.8527	80.9611	118.2525	79.8281	95.9191	121.4453	114.1403	80.8931	78.5527	48.0177
0.05	80.1267	83.3639	120.7048	83.7214	99.7959	123.2631	116.2441	84.7087	81.6520	51.7473

tabulated in Tables 9, 10, and 11, respectively. Based on all these experimental results and visual perception, we demonstrate that the average performance of the proposed method is effective and applicable for image denoising.

From the visual perception of images in Fig. 15, it is clear that the proposed filter preserves edges and texture detail well and removes the additive noise effectively. MSE versus variance graphs of the image in Fig. 15 are plotted in Fig. 16. PSNR versus variance graphs are plotted in Fig. 17, and SSIM versus variance graphs are shown in Fig. 18. Corresponding values are tabulated in Tables 12, 13, and 14, respectively. By all these results, we demonstrate that the average performance of proposed filter is better than other discussed filters.

4.2 Comparison with Some Other Methods

The purpose of this section is to present the advantages of the proposed method over BM3D, a transformed domain method, and a weighted nuclear norm minimization (WNNM)-based image denoising method. The tests are performed on two datasets, named Set12 and BSD68 [19] in which first is the set of 12 images and second contains

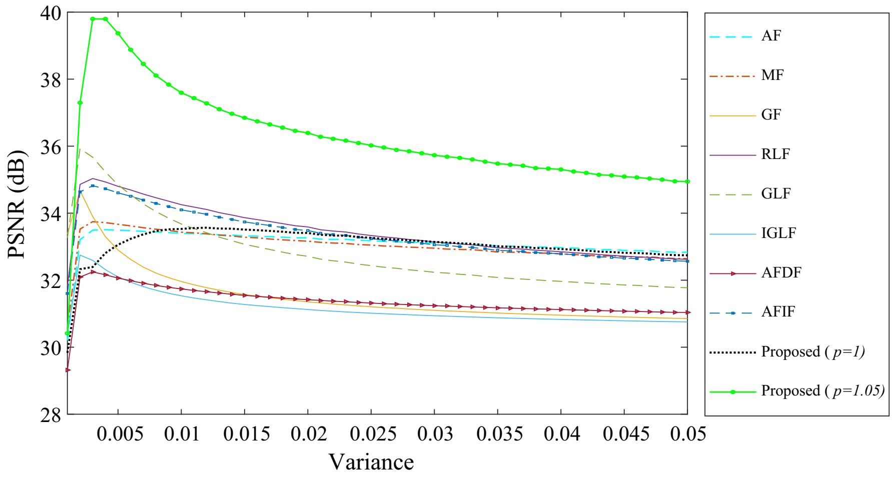

Fig. 9 PSNR versus variance comparison of different filters with proposed filter of image in Fig. 7

Table 7 The values of PSNR in dB of images in Fig. 7 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	30.2071	30.5597	33.3809	31.8173	33.2576	30.3945	29.3187	31.6028	29.8314	30.4144
0.01	33.3974	33.4322	31.9570	34.2542	33.6781	31.5362	31.7390	34.0996	33.5303	37.5946
0.02	33.2663	33.1645	31.3609	33.5897	32.7147	31.1278	31.4220	33.4817	33.4103	36.3942
0.03	33.1086	32.9555	31.1008	33.1403	32.2357	30.9381	31.2388	33.0581	33.1420	35.7301
0.04	32.9731	32.7843	30.9559	32.8522	31.9660	30.8273	31.1267	32.7882	32.9299	35.3051
0.05	32.8342	32.6430	30.8568	32.6223	31.7748	30.7556	31.0385	32.5657	32.7431	34.9440

a set of 68 texture-rich images. Some images of these datasets are shown in Fig. 19. Denoising performance of the proposed method with BM3D and WNNM method on two randomly selected images from datasets [19] are shown in Figs. 20 and 21. Variation of PSNR with noise variance of these two images is plotted in Figs. 22 and 23, respectively. Comparative PSNR of all the images of these two datasets at two different noise variance are shown in Figs. 24, 25, and 26.

From Fig. 20, we observe that the image denoised by the WNNM method is over-smoothed due to which washed-out effect can be seen in Fig 20b as the stick is absent in that image. Image denoised by BM3D method is noise-free up to a certain extent, but regions of very small intensity variation in the backgrounds are not visible due to over-smoothing. Image denoised by the proposed method is better than WNNM method while comparable with BM3D method. Some artifacts are visible in the image denoised by the proposed method due to use of nonoptimal order of derivative for which future studies are needed to extend the proposed work for the development of an adaptive method.

From Fig. 22, it is clear that PSNR of the image denoised by the proposed method at $p = 1.05$ is greater than WNNM method at every level of noise variance while PSNR

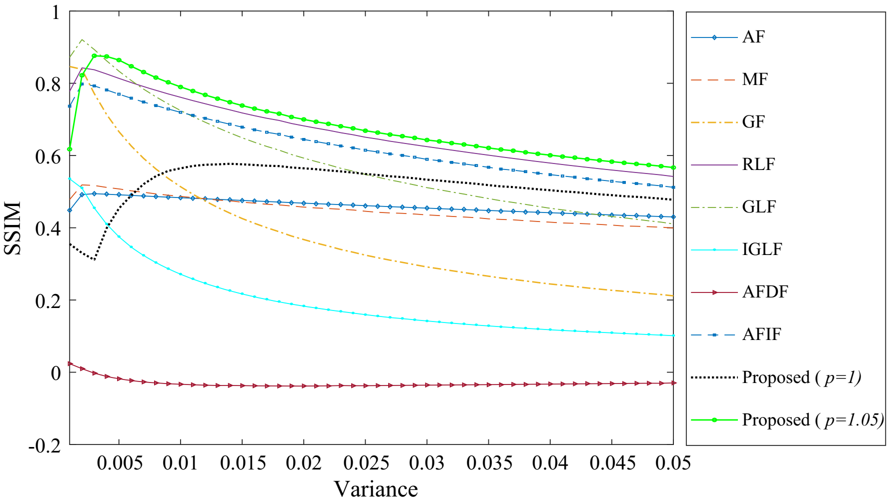

Fig. 10 SSIM versus variance comparison of different filters with proposed filter of image in Fig. 7

Table 8 The values of SSIM of images in Fig. 7 denoised by different filters

Variance\ SSIM	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	0.4484	0.4783	0.8464	0.7778	0.8719	0.5354	0.0238	0.7366	0.3552	0.6175
0.01	0.4832	0.4869	0.5137	0.7609	0.7247	0.2712	−0.0333	0.7196	0.5646	0.7898
0.02	0.4679	0.4573	0.3673	0.6821	0.5929	0.1830	−0.0380	0.6445	0.5646	0.7000
0.03	0.4545	0.4355	0.2915	0.6239	0.5111	0.1414	−0.0354	0.5895	0.5331	0.6429
0.04	0.4417	0.4155	0.2444	0.5789	0.4540	0.1176	−0.0324	0.5471	0.5040	0.6007
0.05	0.4301	0.3992	0.2115	0.5419	0.4106	0.1012	−0.0295	0.5120	0.4775	0.5667

of the image denoised by BM3D method is greater than the proposed method for small variance of the noise. However, the graph of PSNR denoised by BM3D method decays sharply with the increment of noise variance and becomes smaller than the proposed method for greater noise variance. This shows the inconsistency of the BM3D method with noise variance, whereas the PSNR of image denoised by the proposed method is consistent with noise variance.

The image presented in Fig. 21 is a texture-rich image and proposed method has a nice property to preserve the texture during the denoising process that is visible in Fig. 23. Due to this, the PSNR of the image denoised by the proposed method is higher than other methods except for small noise variance. From graph plotted in Fig. 23 and the images in Fig. 21, we demonstrate that the proposed method is comparable to the BM3D method for small noise variance and better for large values of noise variance. The performance of the proposed method is continuously better than WNNM method for all values of noise variance. The advantage of the proposed method is that it is consistent with noise variance.

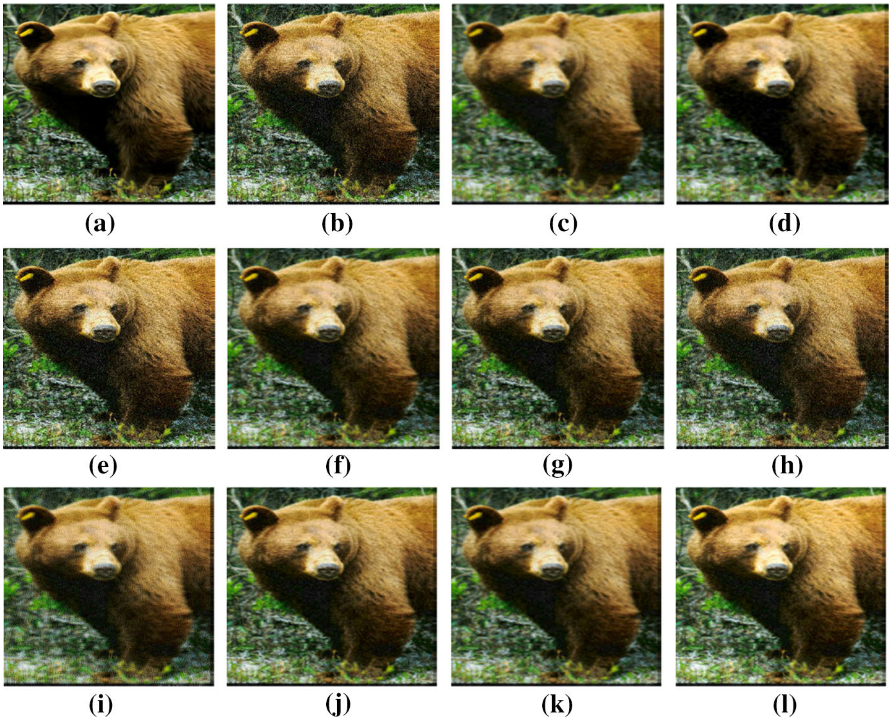

Fig. 11 Experimental results using proposed and existing filters with Gaussian noise, mean 0 and variance 0.03. Color image of size (600 × 394 × 3) pixels "Bear", where **a** sampled image. **b** Image with the Gaussian noise of mean 0 and variance 0.03. **c** Average filter. **d** Median filter. **e** Gaussian smoothing filter. **f** Riemann–Liouville filter. **g** Grunwald–Letnikov filter. **h** Improved Grunwald–Letnikov filter. **i** Alexander fractional differential filter. **j** Alexander fractional integral filter. **k** Proposed filter with $p = 1.0$. **l** Proposed filter with $p = 1.05$

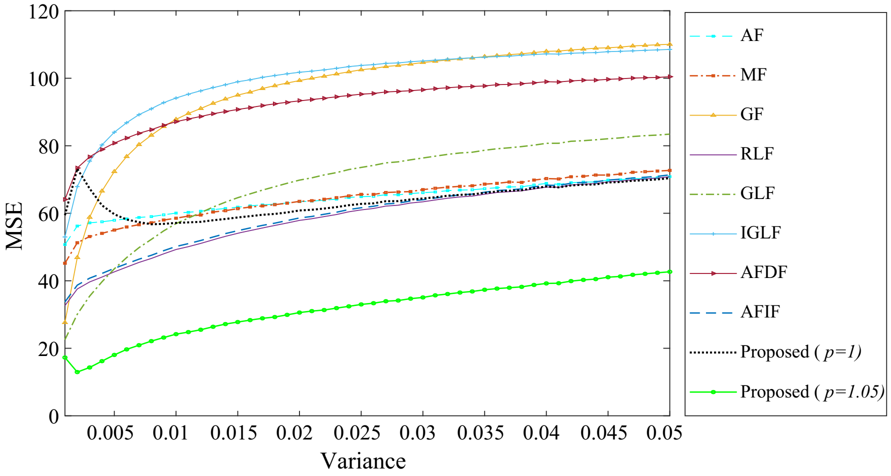

Fig. 12 MSE versus Variance comparison of different filters with proposed filter of image in Fig. 11

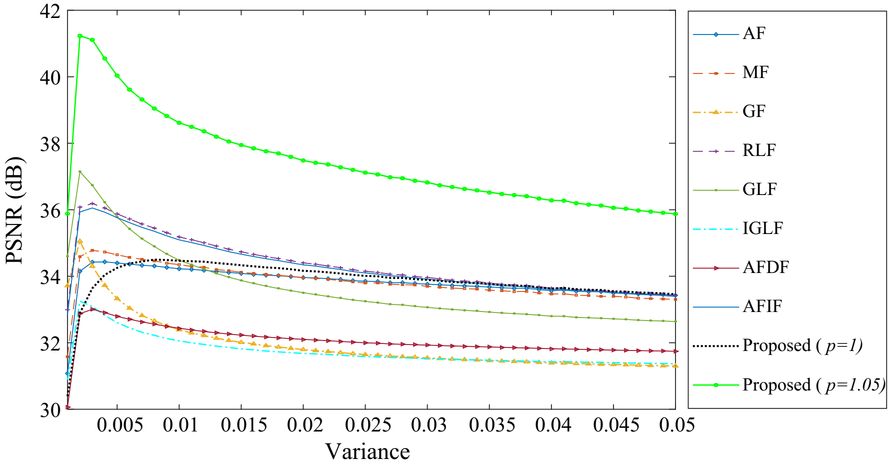

Fig. 13 PSNR versus variance comparison of different filters with proposed filter of image in Fig. 11

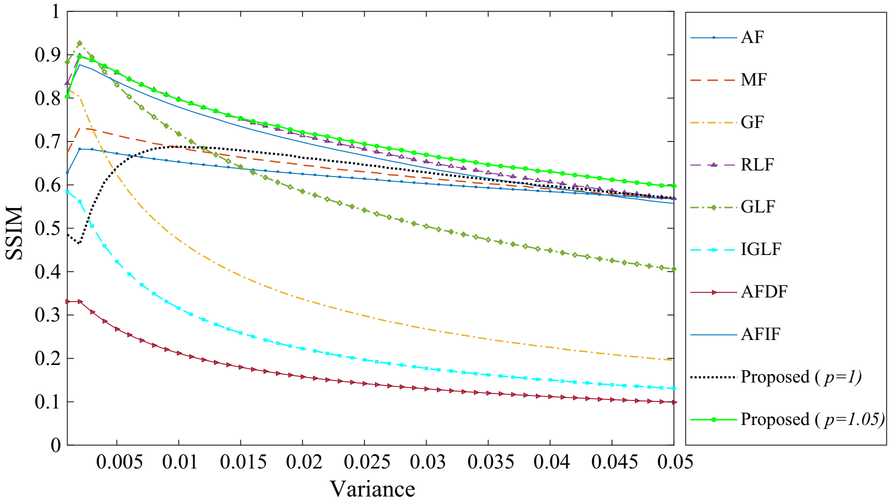

Fig. 14 SSIM versus variance comparison of different filters with proposed filter of image in Fig. 11

From Fig. 24, it is clear that the PSNR of the images denoised by the proposed method is higher than BM3D and WNNM methods for all the images. The average PSNR of the proposed method at $p = 1.05$ is 34.2767 while the PSNR of the BM3D and WNNM methods are 32.6249, and 31.0780, respectively. The average PSNR of the proposed method at $p = 1.0$ is 29.9949 which is slightly less than WNNM method.

The images used in Figs. 25 and 26 are texture-rich images. From these figures, it is observed that PSNR of the image denoised by the proposed method at $p = 1.05$ is higher than BM3D and WNNM methods for almost all the images. The average PSNR of the proposed method at $p = 1.05$ is 34.4693 which is higher than BM3D and WNNM methods as its average PSNR are 31.9877 and 30.3242, respectively. The

Table 9 The values of MSE of image in Fig. 11 denoised by different filters

Variance	AF	MF	GF	RLF	GLF	IGLF	AFDF	AFIF	Proposed	Proposed
	[16]	[31]	[16]	[25]	[18]	[24]	[28]	[28]	$p = 1$	$p = 1.05$
0.001	50.7237	45.1931	27.6271	32.6433	22.5485	52.9880	64.0974	33.7284	59.4233	17.2690
0.01	60.0567	58.5334	87.8222	49.2644	56.9400	94.1284	87.1359	50.1612	57.1435	24.1855
0.02	63.4425	63.5254	99.3385	57.8898	69.8226	101.7793	93.3969	58.6119	60.8293	30.5816
0.03	66.1016	67.0076	104.6998	63.3998	76.3786	105.1061	96.5771	64.0033	64.3742	35.0655
0.04	68.8563	70.2837	107.9738	67.7698	80.7252	107.2296	98.9814	68.3187	67.9106	39.2158
0.05	70.8583	72.7005	110.0636	70.8323	83.4472	108.5433	100.4634	71.3733	70.4010	42.6656

Table 10 The values of PSNR in dB of image in Fig. 11 denoised by different filters

Variance	AF	MF	GF	RLF	GLF	IGLF	AFDF	AFIF	Proposed	Proposed
	[16]	[31]	[16]	[25]	[18]	[24]	[28]	[28]	$p = 1$	$p = 1.05$
0.001	31.0787	31.5801	33.7175	32.9929	34.5996	30.8890	30.0624	32.8509	30.3912	35.8846
0.01	34.2254	34.3494	32.3917	35.1814	34.4829	32.0569	32.4294	35.0944	34.4653	38.6151
0.02	33.9607	33.9545	31.7969	34.4028	33.4984	31.6798	32.0945	34.3430	34.1637	37.4824
0.03	33.7626	33.6969	31.5433	33.9640	33.0652	31.5246	31.9329	33.9182	33.8904	36.8222
0.04	33.5656	33.4668	31.3947	33.6424	32.7982	31.4280	31.8143	33.6034	33.6323	36.2825
0.05	33.4273	33.3035	31.3022	33.4290	32.6381	31.3693	31.7426	33.3923	33.4585	35.8752

Table 11 The values of SSIM of image in Fig. 11 denoised by different filters

Variance	AF	MF	GF	RLF	GLF	IGLF	AFDF	AFIF	Proposed	Proposed
	[16]	[31]	[16]	[25]	[18]	[24]	[28]	[28]	$p = 1$	$p = 1.05$
0.001	0.6277	0.6739	0.8214	0.8341	0.8829	0.5846	0.3310	0.8146	0.4855	0.8036
0.01	0.6528	0.6864	0.4734	0.7980	0.7175	0.3158	0.2121	0.7796	0.6882	0.7963
0.02	0.6247	0.6453	0.3367	0.7137	0.5850	0.2223	0.1576	0.6979	0.6625	0.7211
0.03	0.6029	0.6161	0.2680	0.6535	0.5045	0.1772	0.1296	0.6391	0.6289	0.6693
0.04	0.5847	0.5909	0.2257	0.6072	0.4488	0.1505	0.1120	0.5946	0.5977	0.6308
0.05	0.5680	0.5703	0.1958	0.5690	0.4061	0.1308	0.0989	0.5574	0.5702	0.5975

average PSNR of the proposed method at $p = 1.0$ is 30.2638 which is comparable to the WNNM method. Based on all these results, we demonstrate that the proposed method is considerable for image denoising.

The datasets used in Sect. 4.1 is a part of publicly available standard datasets which are extensively used in the research due to the availability of most of the features in the image. We have randomly selected some of them in our experiments. The images used in Figs. 3, 7, 11, and 15 were downloaded from [20–23].

4.3 Computation Time

This section provides the average computation time of the proposed method and other methods discussed in Sects. 4.1 and 4.2 . The computation time of any algorithm

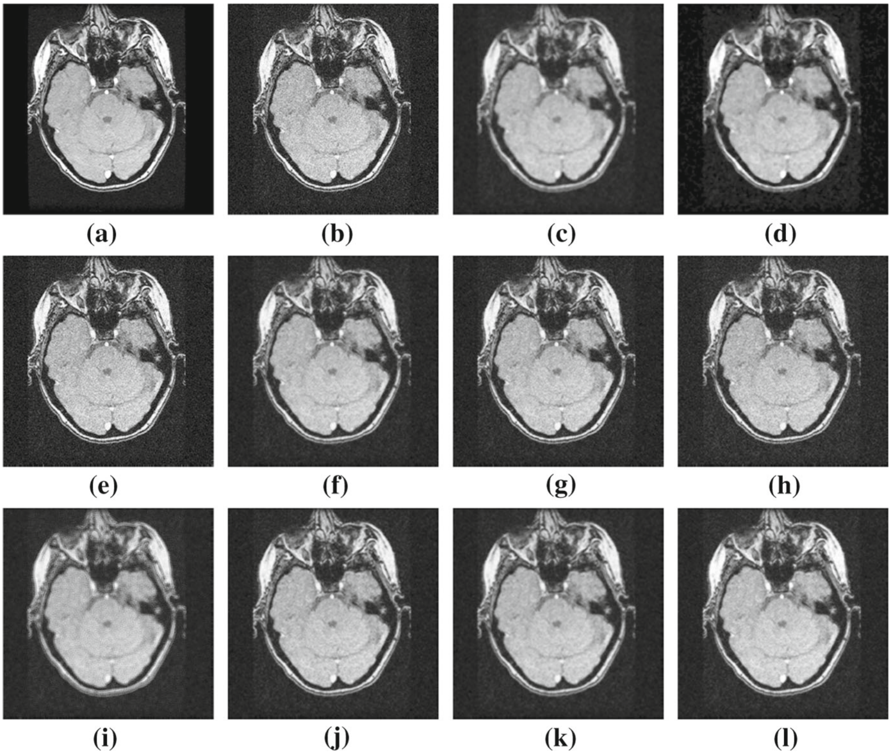

Fig. 15 Experimental results using proposed and existing filters with Gaussian noise, mean 0 and variance 0.02. Gray-scale image of size (512 × 512) pixels "MRI", where **a** sampled image. **b** Image with the Gaussian noise of mean 0 and variance 0.02. **c** Average filter. **d** Median filter. **e** Gaussian smoothing filter. **f** Riemann–Liouville filter. **g** Grunwald–Letnikov filter. **h** Improved Grunwald–Letnikov filter. **i** Alexander fractional differential filter. **j** Alexander fractional integral filter. **k** Proposed filter with $p = 1.0$. **l** Proposed filter with $p = 1.05$

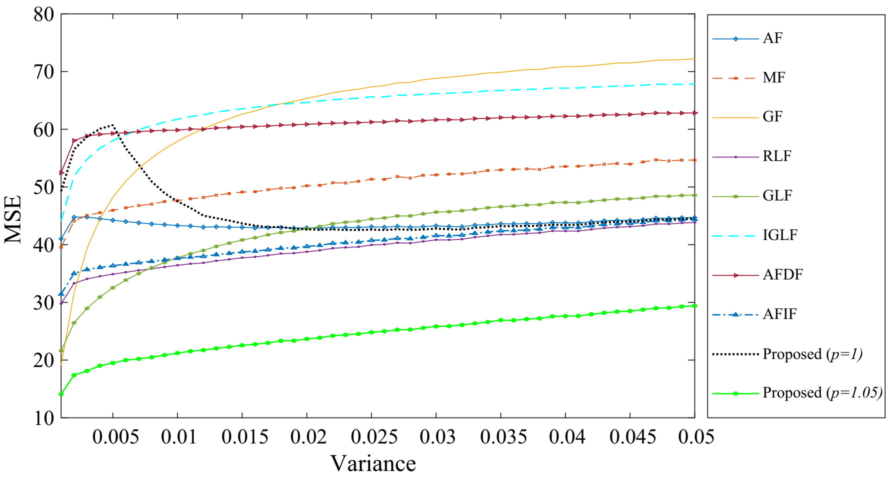

Fig. 16 MSE versus variance comparison of different filters with proposed filter of image in Fig. 15

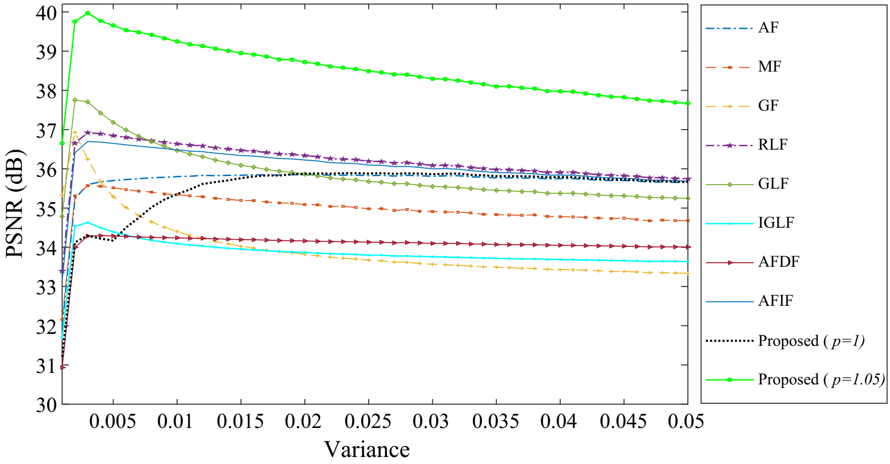

Fig. 17 PSNR versus variance comparison of different filters with proposed filter of image in Fig. 15

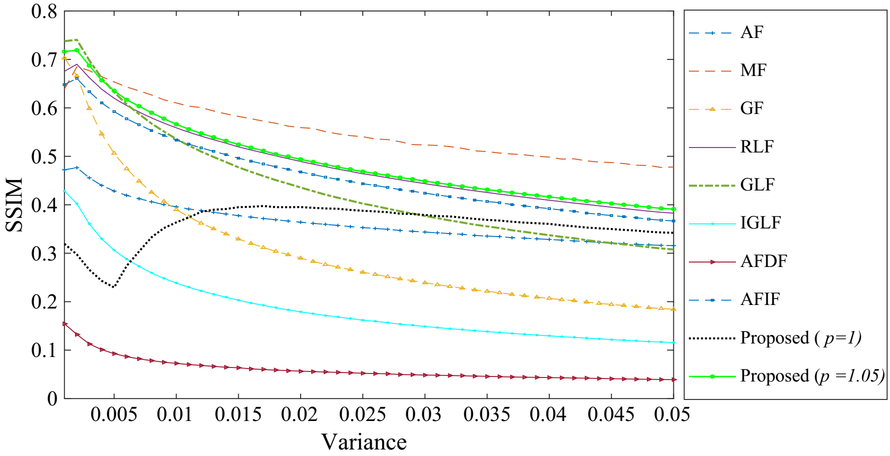

Fig. 18 SSIM versus variance comparison of different filters with proposed filter of image in Fig. 15

plays an important role to decide the applicability in real time. All the algorithms are implemented in MATLAB R2015a using 64-bit operating system, Intel Core i7-3770 CPU with 3.40GHz clock speed, and 4.00GB (3.88GB usable) RAM. The results of computational time of different methods are shown in Table 15. The average computation time of WNNM method is very high in comparison with other discussed methods which is presented separately.

Note that the average CPU time of WNNM [17] method on a single gray image of size (512 × 512) is **3.8585 s**. The computation time of WNNM method is calculated on the same configuration as that of remaining methods. Table 15 and CPU time of WNNM method show that the computational cost of the proposed method is lower than the average filter, median filter, BM3D method, and WNNM method while it is

Table 12 The values of MSE of image in Fig. 15 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	41.0123	39.5616	19.0624	29.7629	21.6041	44.1050	52.4470	31.4410	49.2442	14.0755
0.01	43.3156	47.6380	57.8881	36.4289	37.7292	61.7321	59.8535	37.6247	47.5016	21.2119
0.02	42.9228	50.2162	65.3531	38.7642	42.7528	64.6332	60.8517	39.6843	42.6625	23.6612
0.03	43.2635	52.0984	68.8282	40.8339	45.6424	66.1759	61.6501	41.5846	42.7930	25.8387
0.04	43.7771	53.5594	70.8415	42.3453	47.3218	67.1279	62.2636	42.9621	43.4100	27.6161
0.05	44.6997	54.6522	72.2110	43.8493	48.5804	67.8464	62.8321	44.3840	44.5439	29.4049

Table 13 The values of PSNR in dB of image in Fig. 15 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	32.0017	32.1581	35.3291	33.3941	34.7855	31.6859	30.9336	33.1559	31.2073	36.6513
0.01	35.8022	35.3432	34.4029	36.6377	36.4686	34.0926	34.2417	36.4819	35.3574	39.2476
0.02	35.8462	35.0891	33.8175	36.3380	35.8654	33.8709	34.1619	36.2248	35.8756	38.7203
0.03	35.8081	34.9116	33.5675	36.0870	35.5498	33.7571	34.0990	35.9991	35.8609	38.2954
0.04	35.7511	34.7779	33.4283	35.9116	35.3753	33.6882	34.0512	35.8418	35.7918	37.9743
0.05	35.6504	34.6804	33.3359	35.7431	35.2487	33.6368	34.0073	35.6846	35.6673	37.6714

Table 14 The values of SSIM of image in Fig. 15 denoised by different filters

Variance	AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	Proposed $p = 1$	Proposed $p = 1.05$
0.001	0.4722	0.6387	0.7030	0.6755	0.7377	0.4286	0.1544	0.6480	0.3195	0.7162
0.01	0.3958	0.6102	0.3901	0.5586	0.5358	0.2384	0.0727	0.5335	0.3637	0.5662
0.02	0.3640	0.5593	0.2899	0.4896	0.4357	0.1791	0.0562	0.4677	0.3953	0.4938
0.03	0.3437	0.5234	0.2391	0.4435	0.3769	0.1487	0.0484	0.4239	0.3789	0.4487
0.04	0.3284	0.4985	0.2069	0.4097	0.3373	0.1296	0.0435	0.3920	0.3606	0.4167
0.05	0.3157	0.4780	0.1842	0.3827	0.3078	0.1155	0.0391	0.3666	0.3421	0.3910

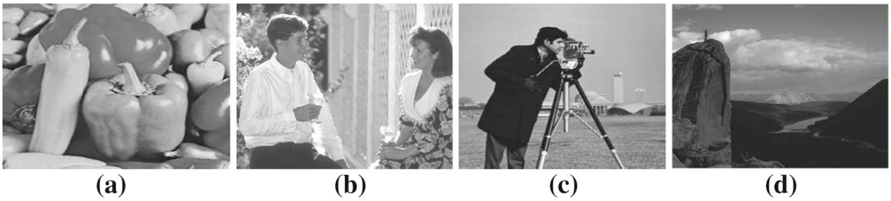

Fig. 19 Sample images used in the experiments

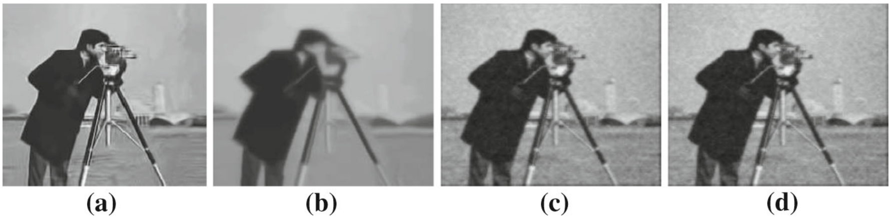

Fig. 20 Experimental results using proposed and other methods with Gaussian noise, mean 0 and variance 0.03. Gray-scale image of size (256 × 256) pixels, where **a** BM3D method [10]. **b** WNNM method [17]. **c** Proposed method with $p = 1.0$. **d** Proposed method with $p = 1.05$

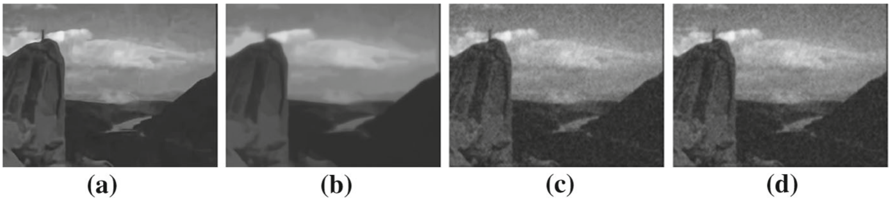

Fig. 21 Experimental results using proposed and other methods with Gaussian noise, mean 0 and variance 0.02. Gray-scale image of size (321 × 481) pixels, where **a** BM3D method [10]. **b** WNNM method [17]. **c** Proposed method with $p = 1.0$. **d** Proposed method with $p = 1.05$

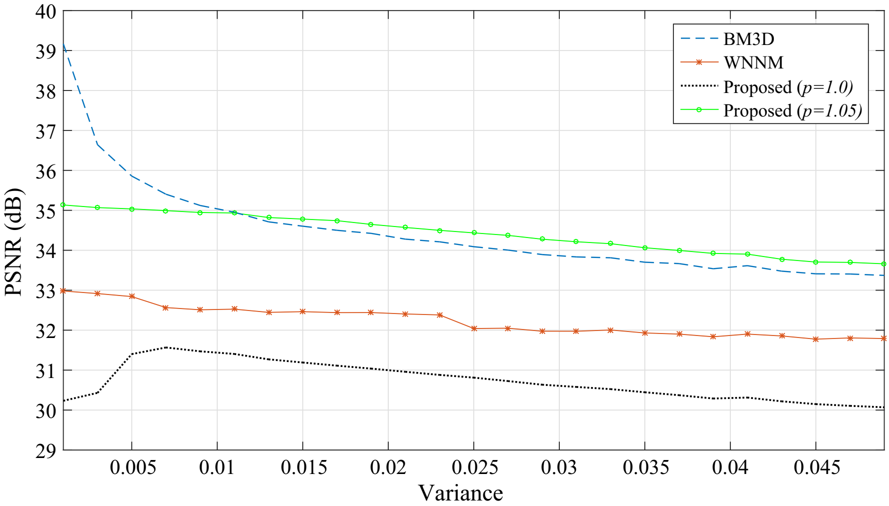

Fig. 22 PSNR versus variance comparison of BM3D and WNNM methods with proposed method of image in Fig. 20

comparable to other discussed methods. Based on computational efficiency, we can say that the proposed method is applicable in real time. The computational cost of the WNNM method is high in comparison with all other discussed methods.

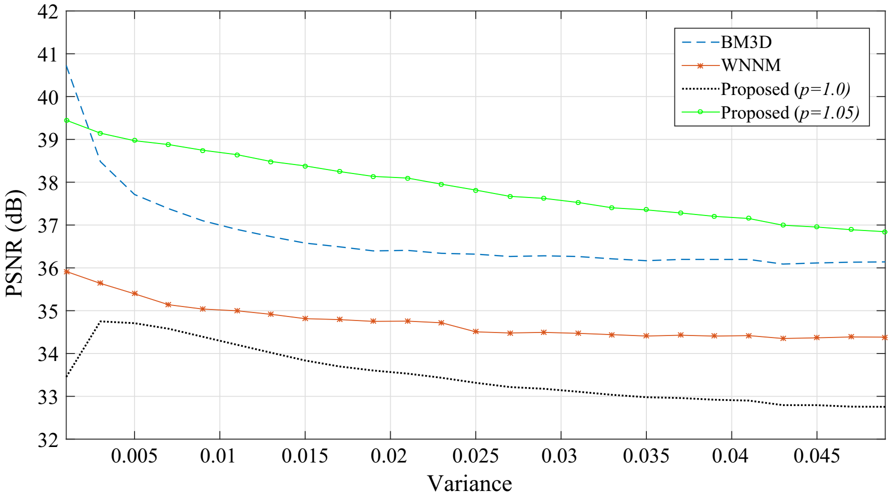

Fig. 23 PSNR versus variance comparison of BM3D and WNNM methods with proposed method of image in Fig. 21

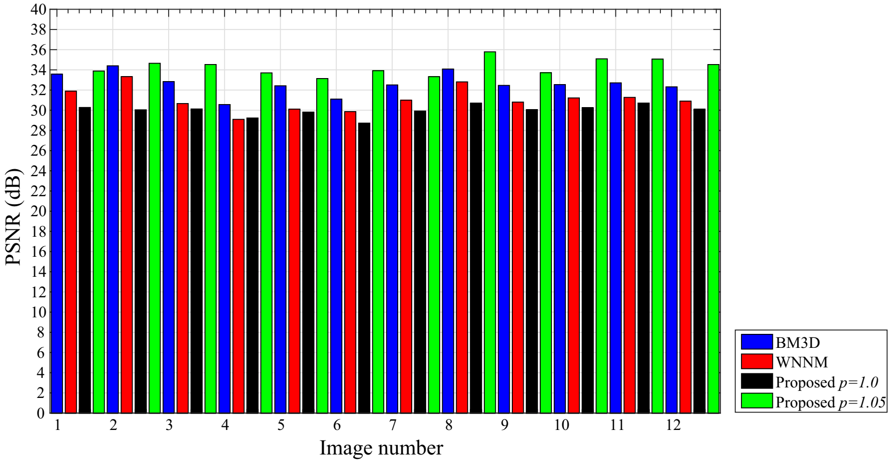

Fig. 24 PSNR comparison on dataset Set12 [19] with Gaussian noise, mean 0 and variance 0.04

5 Conclusions

A novel image denoising algorithm based on K-operator using the Riesz kernel is introduced in this paper. A fractional mask is designed and used in the proposed algorithm. Denoising performance of the algorithm is demonstrated through various parameters such as PSNR, MSE, SSIM, and visual perception. Performance of the proposed method is compared with other existing traditional filters, fractional filters, BM3D, and WNNM methods. Based on numerical results and visual perception, we conclude that the average performance of the proposed algorithm is better than other

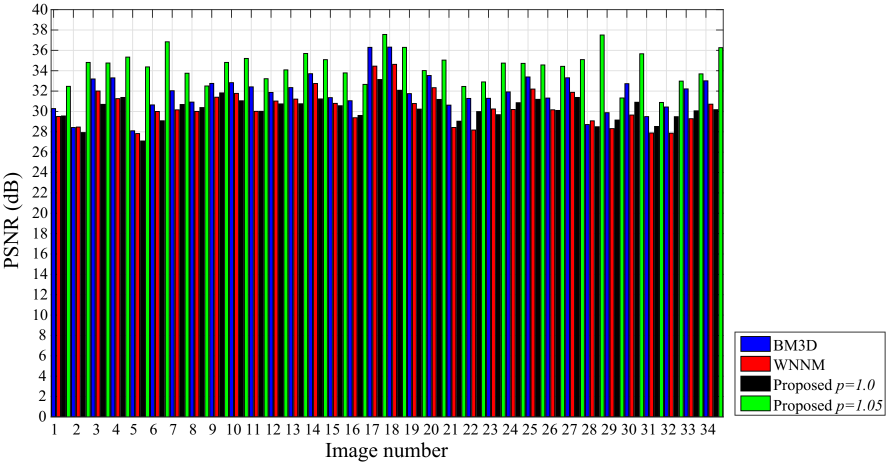

Fig. 25 PSNR comparison of first 34 images taken from dataset BSD68 [19] with Gaussian noise, mean 0 and variance 0.03

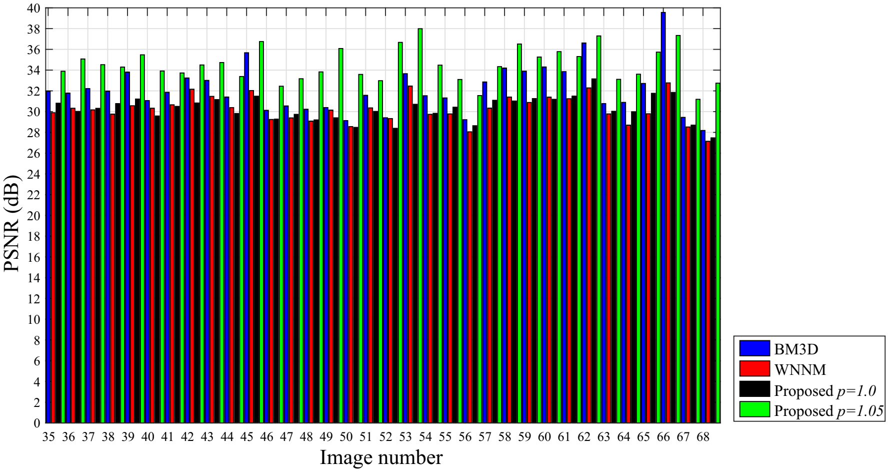

Fig. 26 PSNR comparison of last 34 images taken from dataset BSD68 [19] with Gaussian noise, mean 0 and variance 0.03

Table 15 The average computation times of different methods on 68 image dataset [19] named BSD68

AF [16]	MF [31]	GF [16]	RLF [25]	GLF [18]	IGLF [24]	AFDF [28]	AFIF [28]	BM3D [10]	Proposed
1.898 s	6.227 s	0.952 s	0.761 s	0.761 s	0.713 s	0.7405 s	0.767 s	138.399 s	0.823 s

methods discussed in this paper. The proposed method is computationally efficient and can be implemented in real time. The advantage of the proposed filter is that it removes the noise and preserves texture and edges simultaneously. There are two parameters p and q by which the weights of coefficients of the filter mask are adjusted. The experiments are performed with $(p = 1, q = 0)$ and $(p = 1.05, q = 0)$.

It is noteworthy that the order of integral operator chosen is dependent on the noise and the image feature in this work so we have experimentally obtained the best order for a given noise variance. Future studies are needed to find an adaptive model for the optimum order of integration or derivative to improve the performance.

Acknowledgements The authors sincerely thank the Editor and reviewers for their constructive comments to improve the quality of the manuscript.

References

1. A. Aboshosha, M. Hassan, M. Ashour, M. El Mashade, Image denoising based on spatial filters, an analytical study, in *International Conference on Computer Engineering & Systems, 2009. ICCES 2009* (IEEE, 2009), pp. 245–250
2. O.P. Agrawal, Generalized variational problems and Euler–Lagrange equations. Comput. Math. Appl. **59**(5), 1852–1864 (2010)
3. R.L. Bagley, P.J. Torvik, Fractional calculus in the transient analysis of viscoelastically damped structures. AIAA J. **23**(6), 918–925 (1985)
4. J. Bai, X.C. Feng, Fractional-order anisotropic diffusion for image denoising. IEEE Trans. Image Process. **16**(10), 2492–2502 (2007)
5. G. Baloch, H. Ozkaramanli, R. Yu, Residual correlation regularization based image denoising. IEEE Signal Process. Lett. **25**(2), 298–302 (2018)
6. R. Campagna, S. Crisci, S. Cuomo, L. Marcellino, G. Toraldo, Modification of TV-ROF denoising model based on split Bregman iterations. Appl. Math. Comput. **315**, 453–467 (2017)
7. A. Carpinteri, F. Mainardi, *Fractals and Fractional Calculus in Continuum Mechanics*, vol. 378 (Springer, Berlin, 2014)
8. R.H. Chan, K. Chen, A multilevel algorithm for simultaneously denoising and deblurring images. SIAM J. Sci. Comput. **32**(2), 1043–1063 (2010)
9. Y. Chen, B.M. Vinagre, A new IIR-type digital fractional order differentiator. Signal Process. **83**(11), 2359–2365 (2003)
10. K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
11. L. Debnath, Fractional integral and fractional differential equations in fluid mechanics. Fract. Calc. Appl. Anal. **6**, 119–155 (2003)
12. Y. Farouj, J.M. Freyermuth, L. Navarro, M. Clausel, P. Delachartre, Hyperbolic wavelet-fisz denoising for a model arising in ultrasound imaging. IEEE Trans. Comput. Imaging **3**(1), 1–10 (2017)
13. V. Fedorov, C. Ballester, Affine non-local means image denoising. IEEE Trans. Image Process. **26**(5), 2137–2148 (2017)
14. S. Ghasemi, A. Tabesh, J. Askari-Marnani, Application of fractional calculus theory to robust controller design for wind turbine generators. IEEE Trans. Energy Convers. **29**(3), 780–787 (2014)
15. G. Ghimpeţeanu, T. Batard, M. Bertalmío, S. Levine, A decomposition framework for image denoising algorithms. IEEE Trans. Image Process. **25**(1), 388–399 (2016)
16. R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, vol. 455, 2nd edn. (Publishing House of Electronics Industry, Beijing, 2002)
17. S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 2862–2869

18. H. Guo, X. Li, C. Qing-li, W. Ming-rong, Image denoising using fractional integral, in *IEEE International Conference on Computer Science and Automation Engineering (CSAE), 2012*, vol. 2 (IEEE, 2012), pp. 107–112
19. https://github.com/cszn/DnCNN/tree/master/testsets
20. https://www.nlm.nih.gov/research/visible/image/mri.html
21. https://www.petitcolas.net/watermarking/image_database./index.html
22. https://sipi.usc.edu/database/database.php?volume=misc
23. https://www.imageprocessingplace.com/root_filesv3/image_database.html
24. N. He, J.B. Wang, L.L. Zhang, K. Lu, An improved fractional-order differentiation model for image denoising. Signal Process. **112**, 180–188 (2015)
25. J. Hu, Y.F. Pu, J. Zhou, A novel image denoising algorithm based on Riemann–Liouville definition. JCP **6**(7), 1332–1338 (2011)
26. M.R. Islam, C. Xu, R.A. Raza, Y. Han, An effective weighted hybrid regularizing approach for image noise reduction. Circuits Syst. Signal Process. **38**(1), 1–24 (2018)
27. V. Jain, H.S. Seung, Natural image denoising with convolutional networks. in *Proceedings of the 21st International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2008), pp. 769–776
28. H.A. Jalab, R.W. Ibrahim, Fractional Alexander polynomials for image denoising. Signal Process. **107**, 340–354 (2015)
29. V. Joshi, R.B. Pachori, A. Vijesh, Classification of ictal and seizure-free EEG signals using fractional linear prediction. Biomed. Signal Process. Control **9**, 1–5 (2014)
30. Z. Jun, W. Zhihui, A class of fractional-order multi-scale variational models and alternating projection algorithm for image denoising. Appl. Math. Model. **35**(5), 2516–2528 (2011)
31. B. Justusson, Median filtering: Statistical properties, in *Two-Dimensional Digital Signal Processing II*, ed. by T.S. Huang (Springer, Berlin, 1981), pp. 161–196
32. C. Kervrann, J. Boulanger, Optimal spatial adaptation for patch-based image denoising. IEEE Trans. Image Process. **15**(10), 2866–2878 (2006)
33. A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, *Theory and Applications of Fractional Differential Equations* (Elsevier, Amsterdam, 2006)
34. X. Liu, X.Y. Jing, G. Tang, F. Wu, Q. Ge, Image denoising using weighted nuclear norm minimization with multiple strategies. Signal Process. **135**, 239–252 (2017)
35. R.L. Magin, Fractional calculus in bioengineering, part 1. Crit. Rev. Biomed. Eng. **32**(1), 1–104 (2004)
36. K. Nishimoto, *An Essence of Nishimoto's Fractional Calculus (Calculus in the 21st Century): Integrations and Differentiations of Arbitrary Order* (Descartes Press Company, Waterloo, 1991)
37. M.D. Ortigueira, J.T. Machado, Fractional calculus applications in signals and systems. Signal Process. **10**(86), 2503–2504 (2006)
38. R.K. Pandey, O.P. Agrawal, Comparison of four numerical schemes for isoperimetric constraint fractional variational problems with A-operator. In: ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. V009T07A025–V009T07A025. American Society of Mechanical Engineers (2015)
39. R.K. Pandey, O.P. Agrawal, Numerical scheme for a quadratic type generalized isoperimetric constraint variational problems with A-operator. J. Comput. Nonlinear Dyn. **10**(2), 021,003 (2015)
40. K. Panetta, L. Bao, S. Agaian, Sequence-to-sequence similarity-based filter for image denoising. IEEE Sens. J. **16**(11), 4380–4388 (2016)
41. J. Polack, Time domain solution of Kirchhoff's equation for sound propagation in viscothermal gases: a diffusion process. J. Acoust. **4**, 47–67 (1991)
42. Y.F. Pu, J.L. Zhou, X. Yuan, Fractional differential mask: a fractional differential-based approach for multiscale texture enhancement. IEEE Trans. Image Process. **19**(2), 491–511 (2010)
43. H.K. Rafsanjani, M.H. Sedaaghi, S. Saryazdi, An adaptive diffusion coefficient selection for image denoising. Digit. Signal Process. **64**, 71–82 (2017)
44. L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms. Phys. D Nonlinear Phenom. **60**(1–4), 259–268 (1992)
45. S.G. Samko, A.A. Kilbas, O.I. Marichev, Fractional integrals and derivatives (Gordon and Breach Science Publishers, Yverdon-les-Bains, Switzerland, 1993)
46. S. Sharma, R.K. Pandey, K. Kumar, Collocation method with convergence for generalized fractional integro-differential equations. J. Comput. Appl. Math. **342**, 419–430 (2018)

47. Y. Shen, Q. Liu, S. Lou, Y.L. Hou, Wavelet-based total variation and nonlocal similarity model for image denoising. IEEE Signal Process. Lett. **24**(6), 877–881 (2017)
48. H. Sheng, Y. Chen, T. Qiu, *Fractional Processes and Fractional-Order Signal Processing: Techniques and Applications* (Springer, Berlin, 2011)
49. K.K. Singh, M.K. Bajpai, R.K. Pandey, A novel approach for enhancement of geometric and contrast resolution properties of low contrast images. IEEE/CAA J. Autom. Sin. **5**(2), 628–638 (2018)
50. S. Somali, Implicit midpoint rule to the nonlinear degenerate boundary value problems. Int. J. Comput. Math. **79**(3), 327–332 (2002)
51. S. Suresh, S. Lal, Two-dimensional cs adaptive fir Wiener filtering algorithm for the denoising of satellite images. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. **10**(12), 5245–5257 (2017)
52. C.C. Tseng, Design of fractional order digital FIR differentiators. IEEE Signal Process. Lett. **8**(3), 77–79 (2001)
53. X. Wang, H. Wang, J. Yang, Y. Zhang, A new method for nonlocal means image denoising using multiple images. PLoS ONE **11**(7), e0158,664 (2016)
54. Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity. IEEE Trans. Image Process. **13**(4), 600–612 (2004)
55. S. Xu, Y. Zhou, H. Xiang, S. Li, Remote sensing image denoising using patch grouping-based nonlocal means algorithm. IEEE Geosci. Remote Sens. Lett. **14**(12), 2275–2279 (2017)
56. Q. Yang, D. Chen, T. Zhao, Y. Chen, Fractional calculus in image processing: a review. Fract. Calc. Appl. Anal. **19**(5), 1222–1249 (2016)
57. J. Yu, L. Tan, S. Zhou, L. Wang, M.A. Siddique, Image denoising algorithm based on entropy and adaptive fractional order calculus operator. IEEE Access **5**, 12275–12285 (2017)
58. H. Yue, X. Sun, J. Yang, F. Wu, Image denoising by exploring external and internal correlations. IEEE Trans. Image Process. **24**(6), 1967–1982 (2015)
59. W. Zhao, H. Lu, Medical image fusion and denoising with alternating sequential filter and adaptive fractional order total variation. IEEE Trans. Instrum. Meas. **66**(9), 2283–2294 (2017)