Discrete Laplacian Operator and Its Applications in Signal Processing

WASEEM WAHEED[ID][1]**, GUANG DENG**[ID][1]**, AND BO LIU**[ID][2]
[1]Department of Engineering, La Trobe University, Melbourne, VIC 3086, Australia
[2]School of Computer Science, University of Technology Sydney, Ultimo, NSW 2007, Australia

Corresponding author: Waseem Waheed (w.waheed@latrobe.edu.au)

ABSTRACT Fractional calculus has increased in popularity in recent years, as the number of its applications in different fields has increased. Compared to the traditional operations in calculus (integration and differentiation) which are uniquely defined, the fractional-order operators have numerous definitions. Furthermore, a consensus on the most suitable definition for a given task is yet to be reached. Fractional operators are defined as continuous operators and their implementation requires a discretization step. In this article, we propose a discrete fractional Laplacian as a matrix operator. The proposed operator is real (non-complex) which makes it computationally efficient. The construction of the proposed fractional Laplacian utilizes the DCT transform avoiding the complexity associated with the discretization step which is typical in the constructions based on signal processing. We demonstrate the utility of the proposed operator on a number of data modeling and image processing tasks.

INDEX TERMS Fractional-Laplacian, discrete operator, image-processing, trend-filtering, fractional calculus.

I. INTRODUCTION

Two main operations and building blocks in many engineering disciplines in general [1], [2], image processing and computer vision tasks in particular [3], are first and second order derivatives, otherwise known as the *Gradient* and *Laplacian*. The ubiquity of these operators come from the way we model our systems and problems. Mathematical models relying on the language of calculus are at the center. Some of these modeling paradigms include variational methods, partial differential equations (PDE), statistical and linear/non-linear optimization models.

Over the years, many attempts have been dedicated towards the generalization of those operators to different settings. Among the generalizations is the graph Laplacian [4] which found numerous applications in signal and image processing [5].

Of relevance to this work, is the generalization of integer order differential and integral operators to fractional orders. *Fractional Calculus* (FC) is a 300 years old concept dating back to the days of l'Hôpital and Leibniz [6]. FC has received increased interests over the last 30 years mainly due to their long memory property [2]. For a more recent historical survey, the reader is referred to [7].

Fractional-order derivatives have found numerous applications in electronic circuits and control systems [1], [8], signal processing [9], image processing, computer vision and pattern recognition [10], [11], biological systems and economics [12].

A. FRACTIONAL DERIVATIVES

The integer differential and integral operators are defined uniquely, and they are local. In other words, they consider the values of very close neighboring points to the point of interest. On the other hand, fractional-order differential operators are non-local; larger neighborhoods are considered in the computation resulting in long-term memory effect. This is one of the main reasons behind their appeal. There is a multitude of definitions of the fractional order derivatives [13], [14]. We are going to adopt the notation D^v to denote the derivative of a fractional order v. The following is a list of the most common definitions:

- Forward Grunwald-Letnikov (GL)

$$D^v f(z) = e^{-j\theta v} \lim_{|h| \to 0} \frac{\sum_{k=0}^{\infty}(-1)^k \binom{v}{k} f(z-kh)}{|h|^v} \quad (1)$$

The associate editor coordinating the review of this manuscript and approving it for publication was Peng Wang.

where $\binom{\nu}{k}$ is the binomial coefficient and $h = |h|e^{j\theta}$ is a complex number with $\theta \in (-\pi, \pi]$.
- Riemann-Liouville (RL) fractional derivatives

$$D^{\nu}f(t) = \frac{1}{\Gamma(n-\nu)} \frac{d^n}{dt^n} \int_a^t \frac{f(\tau)}{(t-\tau)^{\nu-n+1}} d\tau \quad (2)$$

where $n - 1 < \nu < n$ and $\Gamma(.)$ is Euler's Gamma function.
- Fourier domain fractional derivatives

$$D^{\nu}f(t) = \mathcal{F}^{-1}\left[(j\omega)^{\nu}\mathcal{F}\{f\}(\omega)\right], \quad \text{Re } \nu > 0 \quad (3)$$

where ω is the Fourier variable, $\mathcal{F}$ and $\mathcal{F}^{-1}$ are the Fourier and the inverse Fourier transform respectively.

Most of the fractional derivative formulations start from a continuous formula which requires discretization to facilitate implementation.

B. FRACTIONAL LAPLACIAN (FL)

The idea of extending the standard Laplacian operator to fractional order is an old idea (c.f [15] and references therein). The standard Laplacian is a local operator. Local operators utilize the immediate neighborhood only in the calculation where outliers can have big impact on the result. Many attempts have been made to create non-local operators that can alleviate the shortcomings of local operators in image processing applications [16]. Utilizing fractional-order Laplacian is such an attempt.

In a similar vein to the fractional derivatives, a multitude of fractional Laplacian (FL) definitions have been proposed over the past few decades but, a consensus on the most appropriate definition for an application is yet to be reached [17].

Of special interest to our work are two variants: the spectral fractional Laplacian [17] which is defined as follows:

$$-(-\Delta)^{\alpha/2}f(x) := \sum_{i \in \mathbb{N}} f_i \lambda_i^{\alpha/2} \phi_i(x) \quad (4)$$

where Δ is the continuous Laplacian operator applied on the function $f(x)$. ϕ_i and λ_i are the eigenfunctions and eigenvalues of the continuous Laplacian Δ respectively.

Secondly, the Fourier-transform based definition (pseudo-differential) [17]

$$\mathcal{F}\left\{(-\Delta)^{\alpha/2}f\right\}(\omega) = |\omega|^{\alpha}\mathcal{F}\{f\}(\omega) \quad (5)$$

It is important to note that both definitions are continuous, and a discretization step is required to make them applicable to digital data. For a comprehensive list of the different formulations, the reader is referred to [12], [15], [17] and references therein.

C. RELATED WORKS

1) FRACTIONAL DERIVATIVE OPERATORS IN IMAGE PROCESSING

Fractional-order derivatives have found numerous applications in image processing. These applications can be categorized into three main categories based on the framework under which the derivative is used. The first category is *linear filtering*. In this category, a fractional-order derivative of an image is calculated through a linear filtering process. In some applications, the gradient image is the goal such as in edge detection [18]–[31] which is the earliest image processing application of fractional calculus. Another application is to use the gradient image to enhance the input image through, for example an un-sharp masking scheme [26], [32]–[37]. The third application in this category is contrast enhancement [29], [38]–[40].

The second category is referred to as *PDE-based* models. Modeling of image restoration problems using integer derivatives dates back to the late 80s and early 90s but utilizing fractional-order derivatives was first presented in 2007 [41] for the image denoising application and was later refined, improved and adopted for a number of applications. Among these applications; image denoising [42]–[49], contrast enhancement [50], [51], image deblurring [52] and image super-resolution [53], [54].

The third category is related to the second and we will refer to it as *variational* models for image restoration problems. Similar to PDE(s), variational models with integer derivatives in the context of image processing date back to the 80s, but the introduction of fractional-order variational models is recent. Applications in this category include image denoising [55]–[66], in-painting [58], [67], fusion [56], [64], non-rigid registration [68], super-resolution [56] and optical flow estimation [69].

For the sake of completeness, we cover a fourth category that is based on a global optimization technique namely: fractional-order Darwinian particle swarm optimization (FODPSO). The reason for distinguishing this category is that the fractional-order derivative is not applied to the image but to a parameter of the model at hand such as the optimal threshold. The main application in this category is image segmentation [70]–[72].

2) FRACTIONAL LAPLACIAN OPERATORS IN IMAGE PROCESSING

The fractional Laplacian has not seen the same amount of adoption as is the case with the fractional derivative in signal and image processing. In [73], the authors introduced a scale-space model. In [74], the authors proposed a quadratic optimization model for blind image deconvolution involving the fractional Laplacian. In [75], the authors proposed a PDE model for vector field estimation flow-sensitive MRI imaging. In [76], the authors have demonstrated image denoising by solving a fractional diffusion equation (PDE). In [77], the authors proposed a variational model for image denoising based on the fractional Laplacian and later was extended to tomographic reconstruction involving the fractional Laplacian as a regularizer [78].

D. CONTRIBUTIONS

The contributions in this work can be summarized in the following:

I) Motivated by the construction of the spectral Laplacian (4), we propose a discrete fractional Laplacian in the form of a matrix operator using the DCT transform. The discrete construction avoids the need for discretization which is typically required by the other fractional Laplacian construction techniques. The discretization step in the case of the pseudo-differential formulation in (5) usually involves solving a filter design problem [79]–[81].

II) Motivated by the reliance of the trend filter on the Laplacian operator, we develop a computationally efficient implementation of traditional trend filtering in the DCT transform domain.

III) Utilizing the proposed discrete fractional Laplacian, we extend both the traditional and the ℓ_1 trend filters to fractional order and demonstrate their effectiveness at the image denoising task for higher levels of noise.

IV) We finally demonstrate the applications of the proposed fractional Laplacian on a number of image processing tasks. However, it is important to stress that we are not claiming state-of-the-art performance in any of these applications. Our aim is to demonstrate the potential of the proposed operator.

The remainder of this article is organized as follows. Section II introduces the fractional Laplacian in the 1D and 2D cases. In section III, we provide two groups of applications for the proposed operator. In the first group namely: fast trend filters, we first introduce the trend filtering followed by a DCT-based implementation of the ℓ_2 trend filter. We then generalize the trend filtering by proposing a fractional ℓ_1 and a fractional ℓ_2 trend filters. In the second group, we present a number of image processing applications demonstrating the effectiveness of the proposed fractional Laplacian. Lastly, a discussion and conclusions are presented in section IV.

II. A DISCRETE FRACTIONAL LAPLACIAN OPERATOR

A. DEFINITION FOR 1D CASE

The Laplacian operator of single variable continuous function $f(x)$ is defined as follows:

$$\Delta f(x) = \frac{\partial^2 f(x)}{\partial x^2} \tag{6}$$

To implement the Laplacian operator, an approximate discretization is used as follows:

$$L\{f(x)\} = \frac{f(x-h) - 2f(x) + f(x+h)}{h^2} \tag{7}$$

where L is the discrete Laplacian operator and h is a small constant. This discretization amounts to uniformly sampling the function $f(x)$ with distance h. This operation can be implemented as a convolution as follows:

$$L\{f(x)\} = l * f \tag{8}$$

where $l = [1, -2, 1]$ is an FIR filter. The Laplacian in (8) can be formulated as a matrix operator in few different ways based on the boundary condition assumed. In this work we

choose a specific formulation with desirable properties as we are going to see later. Specifically, we formulate the Laplacian operator $L \in \mathbb{R}^{N \times N}$ as follows:

$$L = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & . & . & . & \\ & & & -1 & 2 & -1 \\ & & & & 1 & -1 \end{bmatrix} \tag{9}$$

where, the rows between $[1, N-2]$ [1] are shifts of the FIR filter discussed earlier. The signals we are dealing with are finite, and as we have seen in (7), the Laplacian filter is centered around x. In other words, to perform the convolution in (8), $f(x)$ needs to exist before and after x which, is not true at the boundaries. As a solution, we extend the signal on both sides [82].

Thus, the first and last rows in (9) are different from the middle rows. This signal extension near the boundary can be performed in several ways [83]. In (9), we assume a symmetric signal extension as we are going to see next.

The operator in (9) has the same first and last rows, this means equal boundary condition on both sides. For a signal $u \in \mathbb{R}^{N \times 1}$, the boundary condition in (9) assumes the signal u has zero slope at the boundary ($\dot{u}(0) = 0$, $\dot{u}(N-1) = 0$) and is symmetrically extended around the midpoint ($u(-1) = u(0)$, $u(N) = u(N-1)$). For more details about the boundary conditions, the reader is referred to [83].

It is important to note that this particular operator in (9) is diagonalizable by the DCT type-II matrix [83] denoted by M

$$L = M^T E M \tag{10}$$

where E is a diagonal matrix of which the diagonal elements denoted $\{e(k)\}$ are the eigenvalues of L. This spectral decomposition is real which results in efficient computation.

Motivated by the spectral Laplacian in (4) and the diagonalization property of the discrete Laplacian in (10), we define the fractional Laplacian L_α [2] as follows

$$L_\alpha = M^T E^\alpha M \tag{11}$$

where E^α is diagonal matrix with the diagonal elements raised to the power α.

To consolidate, we started from the discrete Laplacian filter $l = [1, -2, 1]$ which we used to formulate the matrix L in (9). Then we performed the eigen-decomposition in (10) and we raised the diagonal eigenvalues matrix E to the power α to finally get the fractional Laplacian matrix L_α in (11).

Recall that when $\alpha = 1$ in (11) we recover the standard Laplacian in (9) but, for α in the open set $\alpha \in (0, 1)$ we get different fractional Laplacian filters. The fractional Laplacians we get are still diagonalizable by the DCT matrix as in (11) and share the same structure as the standard Laplacian

---

[1] Numbering starts from 0

[2] We are using the subscript notation for α to avoid the confusion with the exponent as L is a matrix operator

in (9) with the difference that smaller values of α result in less sparse operator L_α. In other words, smaller α results in Laplacians with more nonzero coefficients hence the memory effect.

As a result of the increase in the number of coefficients, two things become clear: firstly, the dimensions of the fractional Laplacian need to be larger for smaller α to reduce the numerical inaccuracy due to truncation. Secondly, the number of rows in L_α that are involved in the boundary condition increases. One way to recover a good filter l_α from the matrix L_α is to choose the middle row.

It is important to note that the eigenvalues of $L_\alpha \in \mathbb{R}^{N \times N}$ can be obtained by the relation $e(k) = \left[2 - 2\cos\left(k\frac{\pi}{N}\right)\right]$ [84]. Thus, the eigenvalues of the fractional-order Laplacian L_α are defined as follows:

$$e(k; \alpha) = \left[2 - 2\cos\left(k\frac{\pi}{N}\right)\right]^\alpha \quad k = 0, 1, \ldots, N-1. \quad (12)$$

and the coefficients of the eigenvectors matrix are the same as DCT-II matrix which are as follows:

$$M(i, k) = \sqrt{\frac{2}{N}} \cos\left[\left(k + \frac{1}{2}\right)\frac{i\pi}{N}\right], \quad k, i = 0, 1, \ldots, N-1 \quad (13)$$

To get the FIR filter l_α from the operator L_α, we choose the $\frac{N-1}{2}$th row of L_α based on our earlier discussion about avoiding the boundary condition. It can be readily demonstrated that the coefficients of the l_α have the following form:

$$l_\alpha(k) = \sum_{i=0}^{N-1} e(i; \alpha) M\left((N-1)/2, i\right) M(k, i) \quad (14)$$

B. EXTENSION TO THE 2D CASE
Generalization to the 2D case is straight-forward as the 2D-DCT is usually performed as 1D-DCT calculated once on the rows and once on the columns [83]. The benefits of this formulation of the fractional Laplacian in the DCT domain over the traditional Fourier transform formulation are twofold. First, we can construct the fractional Laplacian filter l_α easily allowing for flexibility in applications. Second, it is computationally more efficient as filtering in the DCT domain avoids the need for the image extension as required by the DFT [85].

C. COMPUTATIONAL COMPLEXITY ANALYSIS
From the earlier development, we notice that the proposed fractional Laplacian has two equivalent forms. The first being a matrix operator $L_\alpha \in R^{N \times N}$ which is constructed using (11). To calculate the fractional Laplacian of a signal using L_α, a matrix-vector multiplication is performed, which has $O(N^2)$ computational complexity.

The second form is a linear FIR filter $l_\alpha \in R^N$ which can be constructed directly using (14). Calculating the fractional Laplacian of a signal of length M using l_α is a convolution operation with a computational complexity of $O(NM)$.

D. DISCUSSION
We have delayed the discussion about the other formulations of the matrix operator L for a reason that will become clear soon. The alternative to the proposed Laplacian is to formulate it as a circulant matrix, assuming that the signal has a periodic extension as follows [83]:

$$L = \begin{bmatrix} 2 & -1 & & & & -1 \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & . & . & . & \\ & & & -1 & 2 & -1 \\ -1 & & & & -1 & 2 \end{bmatrix} \quad (15)$$

The L matrix in this formulation is diagonalizable by the discrete Fourier matrix (DFT) [83] which could be generalized the same way we did in (11). However, the proposed Laplacian is computationally more efficient. This computational efficiency comes from two aspects. Firstly, the DFT matrix is complex while, the DCT is real. Secondly, the boundary condition in (15) assumes that the signal is periodic, in other words, the signal needs to be padded with a copy of itself on both sides creating discontinuities along the boundaries. On the other hand, the Laplacian in (9) assumes that the signal is padded with a mirror-image of itself along the boundary. This is a natural boundary condition and it is what MATLAB uses in the `imfilter` command with the `symmetric` option.

In [86], the authors have proposed to compute the fractional derivatives of images implicitly by utilizing a discretization of the following result [87]:

$$D^\nu \cos(\omega t) = \omega^\nu \cos\left(\omega t + \frac{\nu\pi}{2}\right) \quad (16)$$

where ν is the fractional exponent and ω is the continuous frequency variable. In other words, in contrast to what we do in this work, the authors do not provide a direct method for constructing the fractional differential operator (D^ν) in [86]; rather a method to compute its effect on a function. Being able to construct the operators affords more flexibility and applicability in linear algebraic settings.

In [88], the authors propose to approximate the function (16) in the DCT domain by treating it as a FIR filter design problem. This technique produces coefficients of a FIR filter which approximates the ideal fractional derivative operator. The main difference between this technique and ours is that our technique does not use an approximation and it generalizes the Laplacian operator rather than the first order derivative.

III. APPLICATIONS
The Laplacian operator is very ubiquitous in applications and generalizations of which can be examined on such applications. We have split the discussion about applications into two groups. The first group deals with a specific trend estimation technique which relies on the standard Laplacian. We introduce it briefly, propose a new implementation, and

generalize it. In the second group, we consider five image processing applications.

A. FAST TREND FILTERS IN THE DCT DOMAIN

1) DEFINITION

Estimating the trend of a signal or time-series is a common problem in many disciplines [89]. More specifically, given a signal $b \in \mathbb{R}^{N \times 1}$, it is assumed that the signal is composed of two components: a slowly varying component u known as the trend and a rapidly varying white Gaussian noise component denoted ϵ as follows:

$$b = u + \epsilon \tag{17}$$

Trend estimation is the process of producing an estimate $\hat{u}$ for the underlying trend u. The literature on trend estimation is very rich with various parametric and non-parametric techniques [90]. In this article however, we are interested in a specific trend estimation technique which we will refer to as *trend filter*.

Concretely, we use capital letters to represent matrices and small letters to represent column vectors. For example, the nth column vector of the matrix M is denoted M_n, while the nth element of a vector u is denoted u_n. The ℓ_p−norm of u is $||u||_p$. The trend filter [89], [91] is formally defined as follows:

$$\min_u ||u - b||_2^2 + \lambda ||Lu||_p^p \tag{18}$$

A special case of the model (18) is the generalized Wiener filter [92], [93] when $p = 2$, which is otherwise known as the Hodrick-Prescott trend filter in the statistics community [91]. By utilizing the fact that the Laplacian matrix L is diagonalized by the DCT matrix (10), the cost function can equivalently be defined in the DCT domain:

$$\min_u ||u - b||_2^2 + \lambda ||Lu||_2^2 = ||\bar{u} - \bar{b}||_2^2 + \lambda ||M^T E\bar{u}||_2^2 \tag{19}$$

where $\bar{u} = Mu$ and $\bar{b} = Mb$. By minimizing the cost function, we have

$$\bar{u}(k) = \frac{\bar{b}(k)}{1 + \lambda e(k)^2} \tag{20}$$

This result shows that the trend filter can be very efficiently implemented in the DCT domain as an element-wise division since the eigenvalues $e(k)$ can be pre-calculated.

This result also allows us to interpret the trend filter as a low-pass filter in the DCT domain. To this end, let $G = M^T$, the trend filter can thus be represented as follows:

$$u = G\bar{u} \tag{21}$$

$$= \sum_{k=1}^{N} \bar{u}(k)G(k) \tag{22}$$

$$= \sum_{k=1}^{N} \frac{\bar{b}(k)}{1 + \lambda e(k)^2} G(k) \tag{23}$$

On the other hand, the observed vector b can be represented as

$$b = G\bar{b} \tag{24}$$

$$= \sum_{k=1}^{N} \bar{b}(k)G(k) \tag{25}$$

Comparing (23) with (25), we can clearly see an interpretation of the trend filter as a shrinkage operation in the DCT domain. The level of shrinkage is controlled by the regularization parameter λ. Since each column vector $G(k)$ can be regarded as a frequency component with $k = 0$ corresponding to the DC and $k = N - 1$ corresponding to the highest frequency, the shrinkage is also frequency dependent. It can be shown that the eigenvalues have the property $e(0) = 0$ and $e(i) < e(k)$ for $i < k$. As a result, a higher frequency component will be shrunk more. Similar to a linear low-pass filter which attenuates the high frequency components in the Fourier transform domain, the trend filter is a low-pass filter which attenuates the high frequency components in the DCT domain.

2) FRACTIONAL ℓ_2 TREND FILTER

As an application of the proposed operator in (11), we can generalize the trend filter (19) by replacing the L operator with L_α operator leading to a new model. We call it the *fractional trend filter* which is the minimization of the following cost function:

$$||u - b||_2^2 + \lambda ||L_\alpha u||_2^2 \tag{26}$$

The solution to this cost function, can be performed in DCT domain as follows:

$$\bar{u}(k) = \frac{1}{1 + \lambda e(k)^{2\alpha}} \cdot \bar{b}(k) \tag{27}$$

Compared with (20), we have more control over the magnitude response by changing α. To demonstrate the impact of α on the shape of the filter, we plot the filter function (27) at various values of α in Fig. 1. To facilitate the comparison, we normalize the x-axis because, at each α we generate a new Laplacian L_α that has its diagonal elements $e(k) \in [0, 4^\alpha]$. It is important to note that a similar model was proposed in [74] with the difference that the solution was done in the Fourier domain.

From Fig. 1 we notice that lower values of α allow more high frequencies to pass than the higher values of α.

3) FRACTIONAL ℓ_1 TREND FILTERING

We extend the ℓ_1 trend filtering [89] to fractional ℓ_1 trend filtering of the form:

$$\min_u ||u - b||_2^2 + \lambda |L_\alpha u|_1 \tag{28}$$

To solve this problem, we adopt the ADMM algorithm [94] as follows:

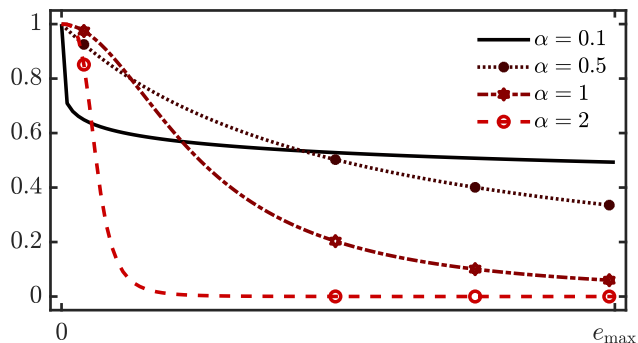$$\min_{u,z} ||u - b||_2^2 + \lambda |z|_1 + \frac{\rho}{2} ||z - L_\alpha u + v||_2^2 \tag{29}$$

FIGURE 1. The shrinkage effect of the fractional ℓ_2 trend filter (first term in (27)) using various values of α.

Algorithm 1 ADMM for Fractional ℓ_1 Trend Filter Using Matrix Form

Result: u^{k+1}
1 $u^0 = b, z^0, v^0 = 0$;
2 **while** $||u^{k+1} - u^k||_2^2/u^{k+1} > \epsilon$ **do**
3    $\quad u^{k+1} = \left(I + \rho L_\alpha^T L_\alpha \right)^{-1} \left(b + \rho L_\alpha^T \left(z^k - u^k \right) \right)$
4    $\quad z^{k+1} = S_{\lambda/\rho} \left(L_\alpha u^{k+1} + v^k \right)$ where
       $\quad S_\kappa(a) = (a - \kappa)_+ - (-a - \kappa)_+$
5    $\quad v^{k+1} = v^k + L_\alpha u^{k+1} - z^{k+1}$
6 **end**

Algorithm 2 ADMM for Fractional ℓ_1 Trend Filter Using Convolutions

Result: u^{k+1}
1 $u^0 = b, z^0, v^0 = 0$;
2 **while** $||u^{k+1} - u^k||_2^2/u^{k+1} > \epsilon$ **do**
3    $\quad u^{k+1} = b - \rho \left(l_\alpha^T * (z^k - u^k) - l_\alpha^T * l_\alpha * u^k \right)$
4    $\quad z^{k+1} = S_{\lambda/\rho} \left(l_\alpha * u^{k+1} + v^k \right)$
5    $\quad v^{k+1} = v^k + l_\alpha * u^{k+1} - z^{k+1}$
6 **end**

In algorithm 2, l_α is one of the middle rows of L_α. It is a FIR filter. In 1D, l_α and its transpose l_α^T are convolved with signals. In 2D, we generate two kernels using l_α for the x and y directions as shown Fig. 2. l_α^T is identical to l_α as they are symmetric around the center. For images, we use an
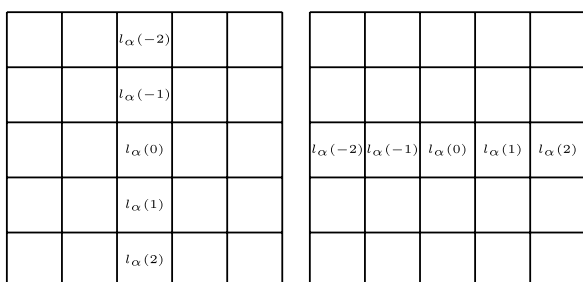
FIGURE 2. 5×5 2D masks for filtering images in (30) (H_1, H_2).

Algorithm 3 ADMM for Fractional ℓ_1 Trend Filter Using FFT

Result: u^{k+1}
1 $u^0 = b, z^0, v^0 = 0$;
2 **while** $||u^{k+1} - u^k||_2^2/u^{k+1} > \epsilon$ **do**
3    $\quad u^{k+1} = \mathcal{F}^{-1} \left\{ \dfrac{\mathcal{F}\{b\} + \rho \mathcal{F}\{l_\alpha\} \odot \mathcal{F}\{z^k - v^k\}}{1 + \rho \mathcal{F}\{l_\alpha\}^2} \right\}$
4    $\quad z^{k+1} = S_{\lambda/\rho} \left(L_\alpha u^{k+1} + v^k \right)$
5    $\quad v^{k+1} = v^k + L_\alpha u^{k+1} - z^{k+1}$
6 **end**
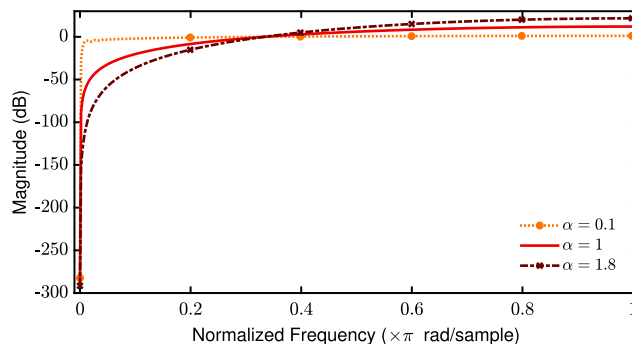
FIGURE 3. Frequency response of fractional Laplacian l_α.

Algorithm 4 Empirical Procedure for Determining the Length N of the Fractional-Order Laplacian Filter l_α

Input: α
Output: l_α, N
1 threshold $= 10^{-3}$;
2 $N = 3$;
3 $l_\alpha =$ use (14) ;
4 **while** $\sum \iota_{(|l_\alpha| > \text{threshold})} = N$ **do**
5    $\quad N \leftarrow N + 1$
6    $\quad$ Calculate l_α using (14)
7 **end**

---

an-isotropic extension of the 1D model as follows

$$\min_u ||u - b||_2^2 + \lambda \left(|H_1 * u|_1 + |H_2 * u|_1 \right) \qquad (30)$$

4) DISCUSSION

We note that the fractional Laplacian in (14), being purely discrete, is a FIR filter of order N. To determine the order of the filter, we need to exploit the structure of the Laplacian. The Laplacian filter is even symmetric and has a positive and relatively high middle coefficient surrounded on both sides by negative coefficients which decay towards zero approaching both ends. Smaller values of α correspond to filters with sharper roll-off in the frequency domain and as a result the filters are generally longer in the discrete domain.

To determine the length of the filter l_α, we propose an empirical procedure that iteratively increases the length
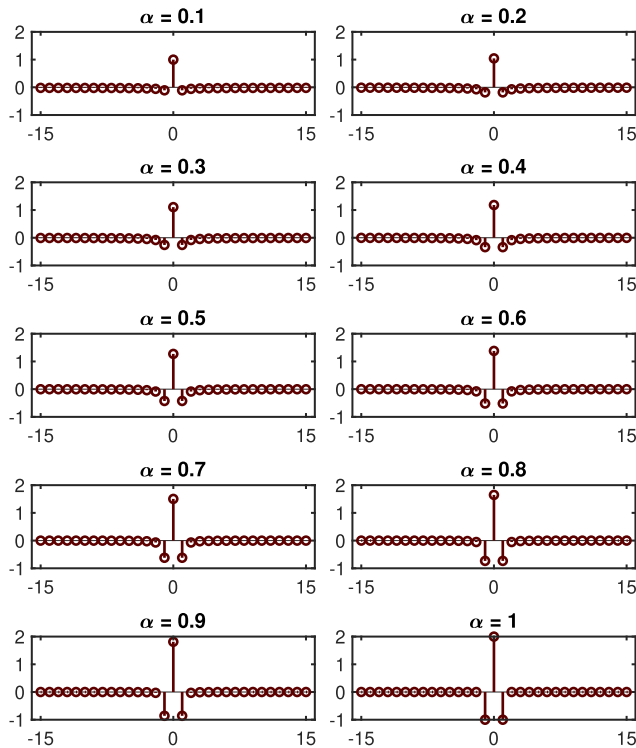
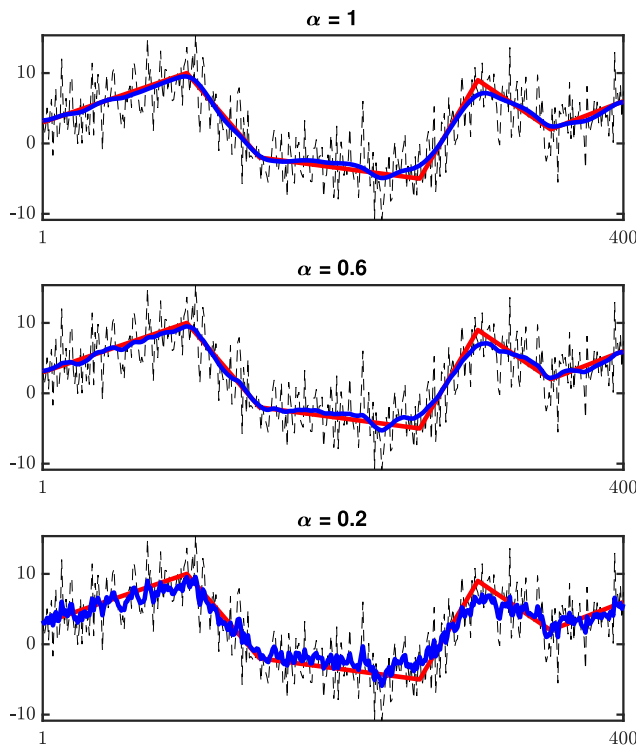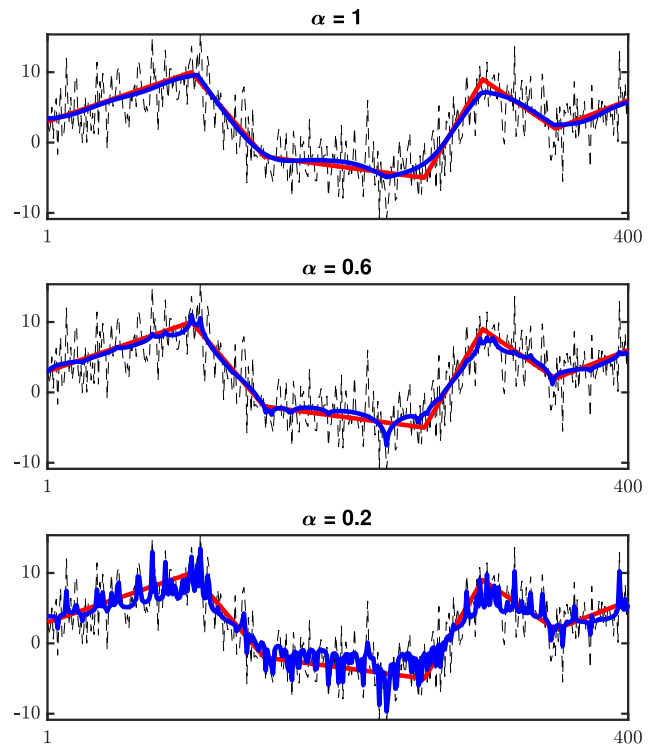**FIGURE 4.** Impulse response of fractional Laplacian l_α.

FIGURE 5. Fractional ℓ_2 trend filtering of synthesized data. Dashed black is the synthetic data. Solid red is the original trend. Solid blue is the filter result with λ chosen to minimize the mean squared error between the filter output and the original trend.

of the filter until the first and last coefficients become lower than a user specified threshold as can be seen in algorithm 4.

FIGURE 6. Fractional ℓ_1 trend filtering of synthesized data. Dashed black is the synthetic data. Solid red is the original trend. Solid blue is the filter result with λ chosen to minimize the mean squared error between the filter output and the original trend.

Frequency and impulse response of fractional Laplacian of various α's is presented in Fig. 3

5) NUMERICAL EXAMPLES

We demonstrate the use of the fractional trend filter in 1D and 2D settings. In the 1D case, we generate a synthetic time-series data with a piece-wise linear trend and additive white Gaussian noise. In Fig. 5 and Fig. 6 synthesized data, synthesized trend and filters' results are reported for three values of $\alpha \in \{1, 0.6, 0.2\}$. From Fig. 5 and Fig. 6 we notice that different fractional orders of trend filtering result in trend estimates that belong to different function families. In the case of $\alpha = 1$ and ℓ_1 fractional trend filter, which corresponds to the original ℓ_1 trend filter, the trend estimates are close to piece-wise linear but that is not the case for $\alpha = 0.6$ or 0.2.

In the 2D case, we test fractional trend filter on the image denoising task. We conducted an experiment on six gray-scale images (cameraman, house, lena, peppers, pirate and blonde woman) shown in Fig. 7. We start with a ground truth image $\tilde{u}$ then we add noise to it with three noise levels $\sigma \in \{15, 25, 50\}$ forming an image b and we experiment with various values of $\alpha \in \{0.1 \ldots 1\}$ with the goal of recovering an image $\hat{u}$ as close as possible to ground truth image $\tilde{u}$. Each experiment was run for 10 times (noise instances) and we are reporting the average performance across the 10 runs. Performance is

FIGURE 7. Grayscale images (256 × 256) used in the denoising experiments. (a) cameraman, (b) house, (c) lena, (d) peppers, (e) pirate and (f) blonde woman.

FIGURE 8. Blocks signal (dashed black) filtered using fractional Laplacian l_α (solid red) with different values of α. From top to bottom are filtered signals with $\alpha \in \{0.1, 0.4, 0.7, 1\}$.

includes the Laplacian operator as a special case. Compared with the Laplacian operator, the fractional Laplacian operator permits the user to "tune" the parameter α to achieve the desired result. As such, in other cases we have achieved better results by tuning α.

To get an idea about the difference in performance between the presented fractional trend filters and the state-of-the-art in image denoising, we have compared with BM3D [95] provided with the underlying noise standard deviation (not an estimate) in Table (3).

B. IMAGE PROCESSING APPLICATIONS

1) 1D FILTERING

To get a better idea about the effect of using the proposed fractional Laplacian, we conduct a comparison on a test signal from the Wavelet Toolbox in MATLAB named blocks. The choice of this signal was made because; it is a piecewise constant signal with sharp edges, allowing for clearer demonstration of the impact of the filters on edges. Results of linear filtering in Fig. 8 demonstrate the long memory effect. The fractional Laplacian with smaller values of α have longer memory effect than the standard Laplacian.

2) IMAGE SHARPENING

One use case for fractional Laplacian is to increase the image sharpness as follows:

$$J = I + \gamma(H_\alpha * I) \qquad (32)$$

where γ is an amplification factor and H_α is the 2D isotropic fractional Laplacian kernel in Fig. 9 formed by adding two kernels similar to the ones in Fig. 2.

measured in the form of mean squared error:

$$\text{MSE} = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \left[\tilde{u}(i,j) - \hat{u}(i,j) \right]^2. \qquad (31)$$

In the case of fractional ℓ_1 trend filtering, the results can be found in Table (1). We notice a pattern in the results that for higher noise levels, better reconstruction is achievable with smaller α. It is important to note here that the best regularization parameter λ was chosen using an exhaustive search.

In the case of fractional ℓ_2 trend filtering, experimental results can be found in Table (2). Similar procedure to ℓ_1 case was conducted. The results here again show that there is a benefit in using a fractional Laplacian for higher levels of noise.

In some cases, such as the "Cameraman" image and the "House" image, it turns out that the setting $\alpha = 1$ (corresponding to the Laplacian operator) leads to the best results. We believe such results should not be regarded as a weakness of the fractional Laplacian operator. Instead, we believe that this is an advantage of the fractional Laplacian operator which

TABLE 1. Mean square error (MSE) of denoised image images for six images using fractional ℓ_1 trend filtering.

(a) Cameraman

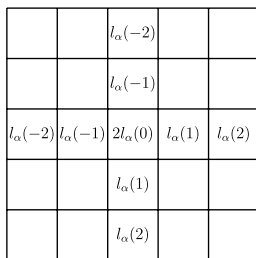| α	15	25	50
0.1	10.87	19.95	34.14
0.2	11.79	18.56	37.11
0.3	13.69	19.40	44.17
0.4	13.78	23.01	34.85
0.5	11.24	16.70	**31.75**
0.6	9.12	**14.70**	32.82
0.7	9.61	15.09	32.54
0.8	**8.39**	15.79	32.96
0.9	8.49	15.06	33.36
1	8.75	15.23	33.84

(b) House

α	15	25	50
0.1	14.65	15.58	20.15
0.2	11.66	12.75	18.32
0.3	9.90	11.15	**18.16**
0.4	9.03	10.45	20.00
0.5	**8.73**	10.40	24.98
0.6	8.80	11.10	35.01
0.7	9.26	13.54	27.04
0.8	10.90	18.74	21.82
0.9	10.91	14.54	24.00
1	9.72	12.09	26.45

(c) Lena

α	15	25	50
0.1	11.56	16.86	21.17
0.2	13.56	14.47	19.51
0.3	12.23	13.20	**19.37**
0.4	11.66	12.68	21.04
0.5	**11.49**	**12.56**	25.63
0.6	11.53	12.68	35.54
0.7	11.64	13.03	28.89
0.8	11.78	14.13	23.85
0.9	11.93	16.11	26.24
1	12.09	16.81	25.17

(d) Peppers

α	15	25	50
0.1	11.15	17.34	**25.06**
0.2	12.94	14.55	25.33
0.3	11.12	13.63	28.64
0.4	10.47	14.59	36.11
0.5	10.88	18.17	35.43
0.6	13.07	16.87	29.12
0.7	9.64	13.38	25.33
0.8	9.90	14.52	26.24
0.9	**8.23**	12.41	27.65
1	8.69	12.49	29.50

(e) Pirate

α	15	25	50
0.1	11.61	16.22	21.49
0.2	12.99	13.94	**20.39**
0.3	11.69	12.71	21.09
0.4	11.10	12.18	24.28
0.5	**10.91**	**12.07**	31.71
0.6	10.94	12.25	36.36
0.7	11.04	12.88	24.59
0.8	11.19	15.26	27.61
0.9	11.35	16.90	22.56
1	11.52	15.65	23.68

(f) Blonde

α	15	25	50
0.1	14.43	15.37	19.86
0.2	12.02	13.15	**18.57**
0.3	10.74	12.11	18.81
0.4	10.28	**12.05**	20.90
0.5	10.46	13.07	25.75
0.6	11.63	16.24	35.17
0.7	12.36	20.73	28.09
0.8	**9.96**	14.25	29.38
0.9	10.17	15.35	23.53
1	10.48	13.33	25.39

TABLE 2. Mean square error (MSE) of denoised image images for six images using fractional ℓ_2 trend filtering.

(a) Cameraman

α	15	25	50
0.1	12.75	18.68	31.44
0.2	11.38	18.70	21.83
0.3	10.82	14.58	19.10
0.4	11.12	13.10	17.80
0.5	10.06	12.59	17.15
0.6	9.50	12.31	**16.87**
0.7	9.21	12.15	17.03
0.8	9.08	12.07	17.51
0.9	9.01	12.04	18.13
1	**8.97**	**12.03**	18.83

(b) House

α	15	25	50
0.1	12.17	17.18	28.01
0.2	11.12	15.83	19.42
0.3	10.29	11.94	16.45
0.4	8.32	10.66	14.83
0.5	7.56	9.95	13.94
0.6	7.21	9.51	**13.87**
0.7	6.99	9.25	14.47
0.8	6.84	**9.13**	15.35
0.9	6.75	9.23	16.33
1	**6.69**	9.45	17.31

(c) Lena

α	15	25	50
0.1	12.61	17.88	28.38
0.2	11.09	17.53	20.79
0.3	10.43	13.74	18.07
0.4	10.51	12.59	16.70
0.5	9.75	12.17	16.03
0.6	9.42	11.95	**15.96**
0.7	9.30	11.86	16.43
0.8	9.27	**11.83**	17.14
0.9	**9.26**	11.85	17.94
1	9.28	11.91	18.77

(d) Peppers

α	15	25	50
0.1	12.69	18.59	31.35
0.2	10.86	18.12	21.37
0.3	9.68	13.10	17.81
0.4	9.30	11.54	15.97
0.5	8.38	10.86	15.01
0.6	8.05	10.47	**14.77**
0.7	7.87	10.25	15.24
0.8	7.76	10.14	16.01
0.9	7.69	**10.14**	16.89
1	**7.66**	10.27	17.80

(e) Pirate

α	15	25	50
0.1	12.47	17.62	27.68
0.2	10.88	16.99	20.33
0.3	10.11	13.37	17.67
0.4	10.13	12.27	16.36
0.5	9.34	11.82	15.73
0.6	8.99	11.58	**15.67**
0.7	8.85	11.46	16.13
0.8	8.79	11.41	16.83
0.9	8.76	**11.41**	17.62
1	**8.75**	11.45	18.45

(f) Blonde

α	15	25	50
0.1	12.01	18.33	24.64
0.2	11.46	15.58	19.16
0.3	10.98	12.55	16.71
0.4	9.42	11.60	15.41
0.5	8.78	11.14	**14.78**
0.6	8.55	10.91	14.91
0.7	8.45	10.80	15.53
0.8	8.41	**10.76**	16.35
0.9	**8.39**	10.81	17.24
1	8.40	10.95	18.14

FIGURE 9. 5 × 5 2D isotropic fractional Laplacian (H_α).

TABLE 3. Mean square error (MSE) of denoised image images for six images using BM3D.

	Noise σ		
α	15	25	50
Cameraman	5.60	7.47	10.87
House	4.26	5.47	8.22
Lena	6.47	8.66	12.31
Peppers	5.38	7.29	11.05
Pirate	7.20	9.53	13.39
Blonde	5.88	7.90	11.66

Fig. 10 is a comparison of image sharpening performance between the standard Laplacian ($\alpha = 1$), the fractional Laplacian ($\alpha = 0.5$) and the guided filter [96]. The results in Fig. 10 clearly demonstrate that a sharper image was achieved using a fractional Laplacian with $\alpha = 0.5$ than the standard Laplacian ($\alpha = 1$), this results is expected as the fractional-order Laplacian captures more information than the standard Laplacian as can be seen in the impulse responses of Laplacian filters in Fig. 3. Sharpening using the guided filter is presented as a comparison with a

FIGURE 10. Image sharpening application. (a) is the original input image. (b) is sharpened image using the guided filter. (c) is sharpened image with $\alpha = 1$. (d) is sharpened image with $\alpha = 0.5$. Sharpened images are produced according to (32) with $\gamma = 2$. In the case of the guided filter, we boost the residual of filtering ($J = I + \gamma(I - GF(I))$).
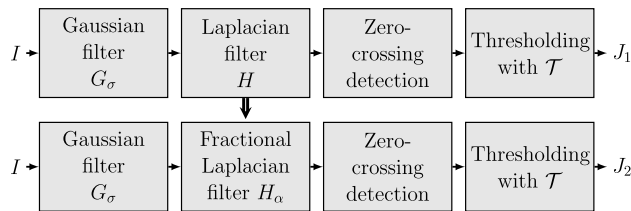

FIGURE 11. Block diagram of the Marr-Hildreth edge detector (top) and the extended version with our fractional Laplacian (bottom).

state-of-the-art edge-aware filter. Recall that the fractional Laplacian is a linear operator while the guided filter is nonlinear.

3) EDGE DETECTION

The literature on edge detection is rich and it is beyond the scope of this study to list and compare with all techniques in the literature however, we are examining the potential use of the proposed fractional Laplacian in this task. We begin with the traditional Marr-Hildreth edge detector [97]. A gray-scale image is first smoothed with a Gaussian filter to reduce the impact of noise and make the detection more robust. This is followed by Laplacian filtering step to find the edges. Filtering an image with the Laplacian kernel results in zero-crossings where edges are potentially located. Next, slopes at the zero-crossings are computed and finally a threshold $\mathcal{T}$ is applied to keep significant edges only. The algorithm is summarized in Fig. 11.
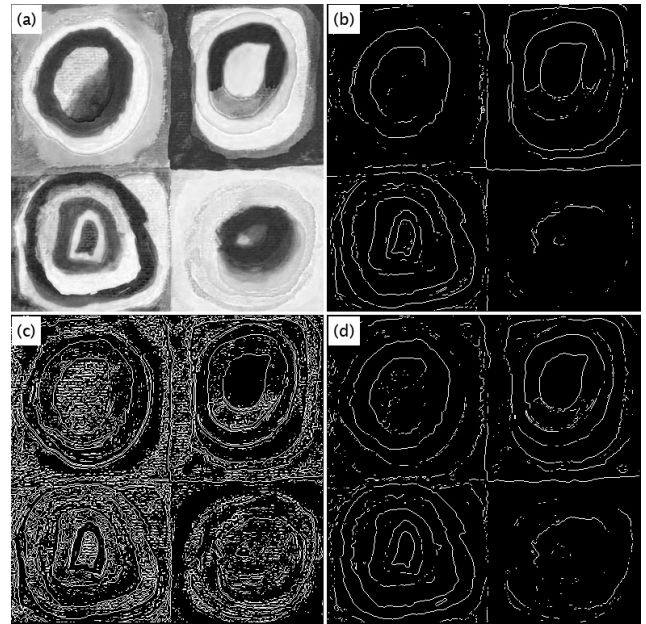
FIGURE 12. Fractional Marr-Hildreth edge detection. (a) is the original image. (b) is edge map produced using MATLAB's *edge* command (based on first order derivatives) using default values. (c) is edge map produced using the standard Marr-Hildreth. (d) is edge map produced using Fractional Marr-Hildreth with $\alpha = 0.2$. Gaussian smoothing with $\sigma = 1$ was used in this experiment.
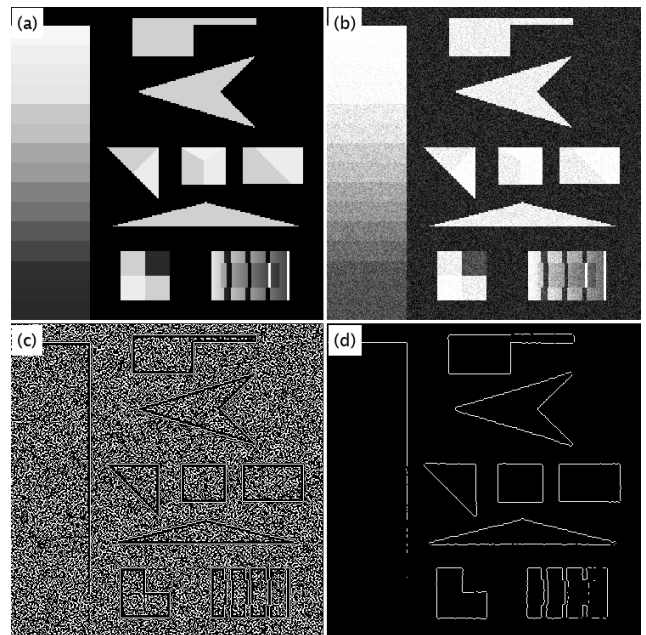
FIGURE 13. Noise Robustness of the fractional Marr-Hildreth edge detector. (a) is the original image. (b) is a noisy image formed by adding noise with $\sigma = 0.3$ to the original image. (c) is edge map produced using the standard Marr-Hildreth. (d) is edge map produced using Fractional Marr-Hildreth with $\alpha = 0.1$. Gaussian smoothing with $\sigma = 1$ was used in this experiment.

We generalize Marr-Hildreth edge detector by replacing the Laplacian by its fractional-order generalization.

Fig. 12 illustrates the results that can be achieved using various values of α. We notice in Fig. 12 that more robust

FIGURE 14. α Shock filter of the top input image. Columns correspond to values of $\alpha \in \{0.6, 1\}$ from left to right. Rows correspond to values of $N \in \{1, 25, 100\}$ from left to right. The parameter λ was set to 0.1 for all results.
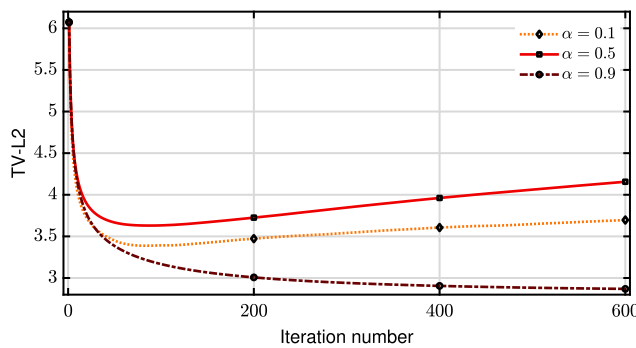
FIGURE 15. TV-L2 calculated for 600 iterations of the model in (36). Each curve corresponds to one value of $\alpha \in \{0.1, 0.5, 0.9\}$.

edge detection in terms of edge lines can be achieved with values of α other than 1.

Edge detectors based on the second order derivative are known to be sensitive to noise [98] which begs the question: does the fractional Laplacian suffer from sensitivity to noise as well? To this end, we used a synthetic image with various forms of edges that commonly exist in natural images, then we added Gaussian noise to it and finally processed it with the fractional Marr-Hildreth edge detector in Fig. 13.
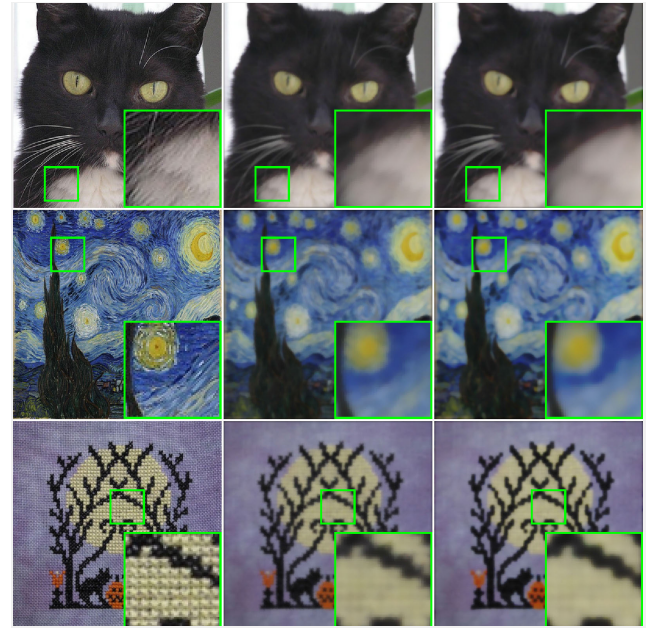
FIGURE 16. α Shock filter with stopping criterion 5. Left column are input image. Middle column is the output images of shock filter with $\alpha = 1$ (Standard shock filter). Right column is the output of shock filtering with $\alpha = 0.1$. The parameter h was set to 0.05 in this experiment.

From Fig. 13 we see that; the fractional Laplacian is less sensitive to noise as expected because it has more coefficients (memory property) than the standard Laplacian.

4) SHOCK FILTERING

Shock filtering was initially proposed by Osher and Rudin [99] for image enhancement but, the technique received a lot of interest from researchers. Shock filters are formulated as PDEs that are evolved over time to come up with the filtered image which is characterized to be piecewise constant. The basic formulation of a shock filter is as follows:

$$\frac{\partial u}{\partial t} = -\operatorname{sign}(\Delta u)|\nabla u| \qquad (33)$$

A more robust version utilizes a smoothing operator such as a Gaussian kernel, and is formulated as follows:

$$\frac{\partial u}{\partial t} = -\operatorname{sign}(\Delta(G_\sigma * u))|\nabla(G_\sigma * u)| \qquad (34)$$
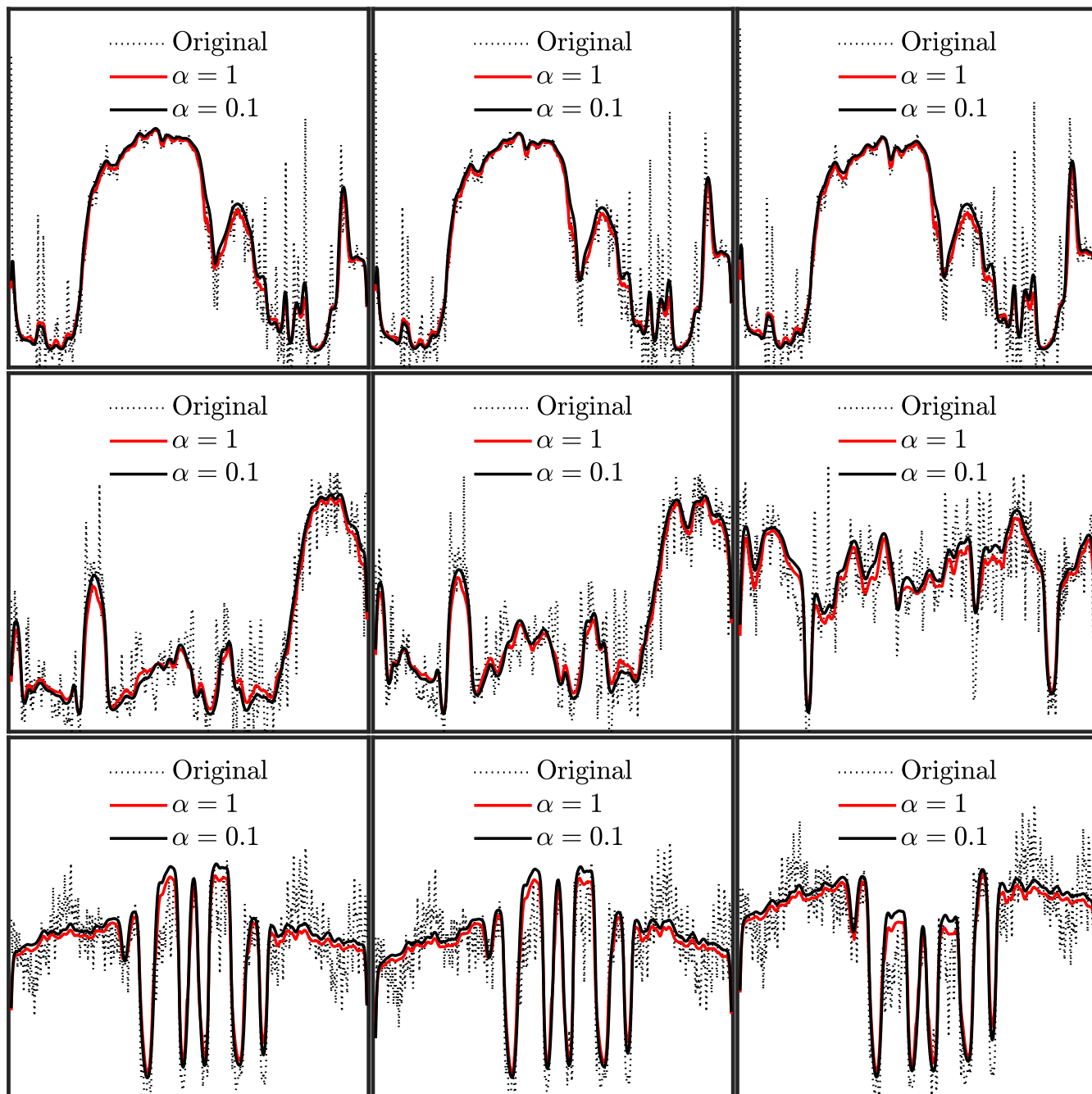
G_σ represents a two-dimensional Gaussian filter with σ being a smoothness parameter and is set to 1 in all experiments in this work.

We extend the proceeding model to fractional order

$$\frac{\partial u}{\partial t} = -\operatorname{sign}(L_\alpha(G_\sigma * u))|\nabla(G_\sigma * u)| \qquad (35)$$

which can be solved using the explicit scheme [3, Appendix A]

$$u^{n+1} = u^n - \lambda \operatorname{sign}(L_\alpha(G_\sigma * u^n))|\nabla(G_\sigma * u^n)| \qquad (36)$$

FIGURE 17. Scan-lines captured from the RGB (left to right) channels of the images shown in Fig. 16. Using the fractional Laplacian results in more more abstract lines.

This extra parameter α gives more control over the filtering effect. In Fig. 14, we compare the fractional ($\alpha = 0.6$) and the standard Laplacian ($\alpha = 1$). The fractional Laplacian produces better object segmentation effect with sharper edges than the case with the standard Laplacian.

a: STOPPING CRITERION

For values of $0 < \alpha < 1$, time stepping (36) for a large amount of time results in images that are not pleasant.

To give a sense of what is happening, and because the shock filter results in piece-wise constant images, we compute a normalized anisotropic TV-L2 cost at every iteration. The TV-L2 is:

$$\text{TV-L2} : \{|u^n - u^0| + |\nabla_x u^n|_1 + |\nabla_y u^n|_1\}/N \qquad (37)$$

where N is the number of pixels in the image u. Fig. 15 demonstrates the measure TV-L2 for different runs of the shock filter at different values of α.
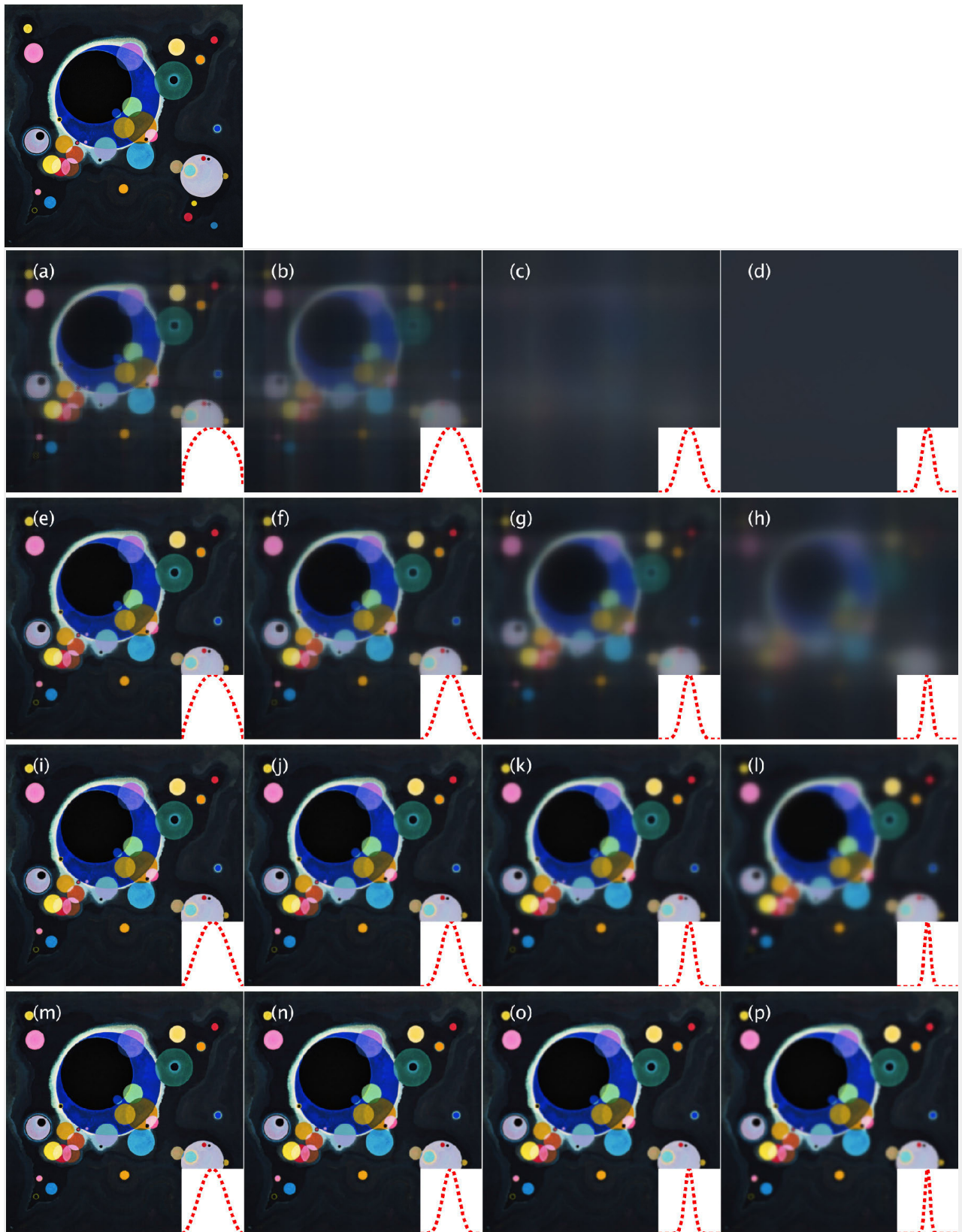
FIGURE 18. α scale-space filtered image on the top. Rows correspond to values of $\alpha \in \{0.2, 0.4, 0.7, 1\}$ from top to bottom. Columns correspond to values of $N \in \{1, 3, 10, 30\}$ from left to right. On the bottom right corner of each image is a plot of the α scale-space function in (42) with the corresponding values of α and N.

It is clear from Fig. 15 that the shock filter might not converge. To avoid the shock filtered result from diverging, we use TV-L2 as a stopping criterion as in Algorithm 5.

Algorithm 5 Fractional Shock Filter With Stopping Criterion

Input: α, I_{input}
Output: u^{n+1}
1 threshold $= 10^{-3}$;
2 $u^0 = I_{input}$;
3 **while** *TV-L2* (37) $>$ *threshold* **do**
4 | u^{n+1} using (36)
5 | $n \leftarrow n+1$
6 **end**

In Fig. 16, the impact of using different values of α in the proposed shock filter is presented for different images where, the number of iterations is determined by the stopping criterion.

One thing to notice in Fig. 16 is that the fractional shock filtering results in more abstract images and sharper edges. To further validate this observation, we plot in Fig. 17 a scan line from the three channels (RGB) for the four images in Fig. 16. In each subplot of Fig. 16, we have scan-lines of two cases to facilitate visual comparison.

5) α SCALE-SPACE
The fractional Laplacian could be used as replacement for the standard Laplacian in the heat equation, this is known as α scale-space model [100]:

$$\frac{\partial u}{\partial t} = -L_\alpha u \tag{38}$$

This leads to different diffusion effects and rates. The implementation of this diffusion can be carried out efficiently in the DCT domain. First, we write the explicit scheme [3]:

$$2u_{t+1} = u_t - L_\alpha u_t \tag{39}$$

$$u_{t+1} = u_t - M^T E^\alpha M u_t \tag{40}$$

$$\underbrace{M u_{t+1}}_{U_{t+1}} = \underbrace{M u_t}_{U_t} - \underbrace{M M^T}_{I} E^\alpha \underbrace{M u_t}_{U_t} \tag{41}$$

Consequently, the filtered signal at iteration n becomes:

$$U_n = (I - E^\alpha)^N U_0 \tag{42}$$

A demonstration of the effect of α scale-space is presented in Fig. 18. It is important to stress that this is an iterative linear filtering process. The filters are characterized as having low-pass response. The authors of [73] have explored combining multiple fractions of the Laplacian implemented in the Fourier transform domain.

IV. SUMMARY AND CONCLUSION
In this article, we have presented a new technique for constructing a fractional Laplacian using the DCT transform.

The proposed operator is a matrix operator which avoids the need for discretization, a necessary step for implementation, typically done in DSP-based constructions. We have also studied the trend filter and provided a new DCT-based implementation for it, which we used later in generalizing trend filters to the fractional-order. The proposed fractional Laplacian allowed us to make the generalization of another version of the trend filter namely the ℓ_1 trend filter to fractional-order.

To test the efficiency of the new operator, we have incorporated it in five applications that traditionally relied on the Laplacian operator. Firstly, we used the proposed fractional trend filters for image denoising and showed that for higher noise levels, the fractional-order Laplacians tend to produce better results than the standard Laplacian. Secondly, we used the proposed operator in image sharpening and stronger sharpening effect was achieved compared to the standard Laplacian. Thirdly, Marr-Hildreth scheme for edge detection was generalized and more robust edge detection was demonstrated. Fourthly, we showed that the use of fractional-order Laplacian in shock filtering resulted in better segmented images. Finally, we incorporated the proposed fractional Laplacian in the α scale-space scheme and showed that the fractional Laplacian resulted in faster diffusion.

Finally, we reiterate; the aim of this paper is to present an alternative way to define the fractional Laplacian operator, which, to our best knowledge, has not been published before. We demonstrate its potential applications in solving a wide range of problems from data modeling to image processing. We observe from the simulations, the performance is sometimes sub-optimal. This is a result of applying the fractional Laplacian operator in a direct and non-sophisticated fashion. Further improvement can be achieved by using the fractional Laplacian operator as a building block in some successful algorithms.

REFERENCES

[1] S. Das, *Functional Fractional Calculus for System Identification and Controls*. Berlin, Germany: Springer-Verlag, 2007.

[2] M. D. Ortigueira, *Fractional Calculus for Scientists and Engineers*. Amsterdam, The Netherlands: Springer-Verlag, 2011.

[3] P. Kornprobst and G. Aubert, *Mathematical Problems in Image Processing*. New York, NY, USA: Springer-Verlag, 2006.

[4] O. Lezoray and L. Grady, *Image Processing and Analysis With Graphs*. New York, NY, USA: Taylor & Francis, 2017.

[5] G. Cheung, E. Magli, Y. Tanaka, and M. K. Ng, "Graph spectral image processing," *Proc. IEEE*, vol. 106, no. 5, pp. 907–930, May 2018.

[6] S. G. Samko, A. A. Kilbas, and O. I. Marichev, *Fractional Integrals and Derivatives*, vol. 1993. Yverdon Yverdon-les-Bains, Switzerland: Gordon and Breach Science Publishers, 1993.

[7] J. T. Machado, V. Kiryakova, and F. Mainardi, "Recent history of fractional calculus," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, no. 3, pp. 1140–1153, 2011.

[8] A. Elwakil, "Fractional-order circuits and systems: An emerging interdisciplinary research area," *IEEE Circuits Syst. Mag.*, vol. 10, no. 4, pp. 40–50, 4th Quart., 2010.

[9] R. Magin, M. D. Ortigueira, I. Podlubny, and J. Trujillo, "On the fractional signals and systems," *Signal Process.*, vol. 91, no. 3, pp. 350–371, Mar. 2011.

[10] Q. Yang, D. Chen, T. Zhao, and Y. Chen, "Fractional calculus in image processing: A review," *Fractional Calculus Appl. Anal.*, vol. 19, no. 5, pp. 1222–1249, Jan. 2016.

[11] S. Samko, "Fractional integration and differentiation of variable order: An overview," *Nonlinear Dyn.*, vol. 71, no. 4, pp. 653–662, Mar. 2013.

[12] C. Pozrikidis, *The Fractional Laplacian*. London, U.K.: Chapman & Hall, 2016.

[13] M. Ortigueira, "An introduction to the fractional continuous-time linear systems: The 21st century systems," *IEEE Circuits Syst. Mag.*, vol. 8, no. 3, pp. 19–26, 3rd Quart., 2008.

[14] E. C. de Oliveira and J. A. T. Machado, "A review of definitions for fractional derivatives and integral," *Math. Problems Eng.*, vol. 2014, pp. 1–6, Jun. 2014.

[15] J. L. Vázquez, "The mathematical theories of diffusion: Nonlinear and fractional diffusion," in *Nonlocal and Nonlinear Diffusions and Interactions: New Methods and Directions*. Cham, Switzerland: Springer, 2017, pp. 205–278.

[16] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, no. 3, pp. 1005–1028, Jan. 2009.

[17] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. M. Meerschaert, M. Ainsworth, and G. E. Karniadakis, "What is the fractional Laplacian? A comparative review with new results," *J. Comput. Phys.*, vol. 404, Mar. 2020, Art. no. 109009.

[18] B. Mathieu, P. Melchior, A. Oustaloup, and C. Ceyral, "Fractional differentiation for edge detection," *Signal Process.*, vol. 83, no. 11, pp. 2421–2432, Nov. 2003.

[19] Y. Pu, W. Wang, J. Zhou, Y. Wang, and H. Jia, "Fractional differential approach to detecting textural features of digital image and its fractional differential filter implementation," *Sci. China F, Inf. Sci.*, vol. 51, no. 9, pp. 1319–1339, Sep. 2008.

[20] C. B. Gao, J. L. Zhou, J. R. Hu, and F. N. Lang, "Edge detection of colour image based on quaternion fractional differential," *IET Image Process.*, vol. 5, no. 3, pp. 261–272, Apr. 2011.

[21] W. X. Wang, W. S. Li, and X. Yu, "Fractional differential algorithms for rock fracture images," *Imag. Sci. J.*, vol. 60, pp. 103–111, Apr. 2012.

[22] J.-L. Chen, C.-H. Lin, Y.-C. Du, and C.-H. Huang, "Combining fractional-order edge detection and chaos synchronisation classifier for fingerprint identification," *IET Image Process.*, vol. 8, no. 6, pp. 354–362, Jun. 2014.

[23] C. Chi and F. Gao, "Palm print edge extraction using fractional differential algorithm," *J. Appl. Math.*, vol. 2014, pp. 1–7, Apr. 2014.

[24] M. Ö. Arısoy and Ü. Dikmen, "Edge enhancement of magnetic data using fractional-order-derivative filters," *Geophysics*, vol. 80, no. 1, pp. J7–J17, Jan. 2015.

[25] A. Acharya, U. Mukherjee, and C. Fowlkes, "On image segmentation using fractional gradients-learning model parameters using approximate marginal inference," May 2016, *arXiv:1605.02240*. [Online]. Available: https://arxiv.org/abs/1605.02240

[26] P. Amoako-Yirenkyi, J. K. Appati, and I. K. Dontwi, "A new construction of a fractional derivative mask for image edge analysis based on Riemann–Liouville fractional derivative," *Adv. Difference Equ.*, vol. 2016, no. 1, p. 238, Dec. 2016.

[27] S. Kumar, R. Saxena, and K. Singh, "Fractional Fourier transform and fractional-order calculus-based image edge detection," *Circuits, Syst., Signal Process.*, vol. 36, no. 4, pp. 1493–1513, Apr. 2017.

[28] C. Georgescu, "Improved edge detection algorithms based on a Riesz fractional derivative," in *Image Analysis and Recognition*, A. Campilho, F. Karray, and B. ter Haar Romeny, Eds. Cham, Switzerland: Springer, 2018, pp. 201–209.

[29] A. Nandal, H. Gamboa-Rosales, A. Dhaka, J. M. Celaya-Padilla, J. I. Galvan-Tejada, C. E. Galvan-Tejada, F. J. Martinez-Ruiz, and C. Guzman-Valdivia, "Image edge detection using fractional calculus with feature and contrast enhancement," *Circuits, Syst., Signal Process.*, vol. 37, no. 9, pp. 3946–3972, Sep. 2018.

[30] J. E. Lavín-Delgado, J. E. Solís-Pérez, J. F. Gómez-Aguilar, and R. F. Escobar-Jiménez, "A new fractional-order mask for image edge detection based on Caputo–Fabrizio fractional-order derivative without singular kernel," *Circuits, Syst., Signal Process.*, vol. 39, no. 3, pp. 1419–1448, Mar. 2020.

[31] R. A. Z. Daou, F. El Samarani, C. Yaacoub, and X. Moreau, "Fractional derivatives for edge detection: Application to road obstacles," in *Smart Cities Performability, Cognition, & Security*. Cham, Switzerland: Springer, 2020, pp. 115–137.

[32] J. You, S. Hungnahally, and A. Sattar, "Fractional discrimination for texture image segmentation," in *Proc. Int. Conf. Image Process.*, vol. 1, Oct. 1997, pp. 220–223.

[33] Y. Pu, "Fractional calculus approach to texture of digital image," in *Proc. 8th Int. Conf. Signal Process. (ICSP)*, vol. 2, Nov. 2006, pp. 1–5.

[34] Y.-F. Pu, J.-L. Zhou, and X. Yuan, "Fractional differential mask: A fractional differential-based approach for multiscale texture enhancement," *IEEE Trans. Image Process.*, vol. 19, no. 2, pp. 491–511, Feb. 2010.

[35] M. Xu, J. Yang, D. Zhao, and H. Zhao, "An image-enhancement method based on variable-order fractional differential operators," *Bio-Med. Mater. Eng.*, vol. 26, no. s1, pp. S1325–S1333, Aug. 2015.

[36] S. Hemalatha and S. M. Anouncia, "G-L fractional differential operator modified using auto-correlation function: Texture enhancement in images," *Ain Shams Eng. J.*, vol. 9, no. 4, pp. 1689–1704, Dec. 2018.

[37] X. Luo, T. Zeng, W. Zeng, and J. Huang, "Comparative analysis on landsat image enhancement using fractional and integral differential operators," *Computing*, vol. 102, no. 1, pp. 247–261, Jan. 2020.

[38] S. Khanna and V. Chandrasekaran, "Fractional derivative filter for image contrast enhancement with order prediction," in *Proc. IET Conf. Image Process. (IPR)*, Jul. 2012, pp. 1–6.

[39] K. Kaur, N. Jindal, and K. Singh, "Improved homomorphic filtering using fractional derivatives for enhancement of low contrast and non-uniformly illuminated images," *Multimedia Tools Appl.*, vol. 78, no. 19, pp. 27891–27914, Oct. 2019.

[40] Q. Dai, Y.-F. Pu, Z. Rahman, and M. Aamir, "Fractional-order fusion model for low-light image enhancement," *Symmetry*, vol. 11, no. 4, p. 574, Apr. 2019.

[41] J. Bai and X.-C. Feng, "Fractional-order anisotropic diffusion for image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2492–2502, Oct. 2007.

[42] M. Janev, S. Pilipović, T. Atanacković, R. Obradović, and N. Ralević, "Fully fractional anisotropic diffusion for image denoising," *Math. Comput. Model.*, vol. 54, nos. 1–2, pp. 729–741, Jul. 2011.

[43] E. Cuesta, M. Kirane, and S. A. Malik, "Image structure preserving denoising using generalized fractional time integrals," *Signal Process.*, vol. 92, no. 2, pp. 553–563, Feb. 2012.

[44] S. Larnier and R. Mecca, "Fractional-order diffusion for image reconstruction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2012, pp. 1057–1060.

[45] Y.-F. Pu, P. Siarry, J.-L. Zhou, and N. Zhang, "A fractional partial differential equation based multiscale denoising model for texture image," *Math. Methods Appl. Sci.*, vol. 37, no. 12, pp. 1784–1806, Aug. 2014, doi: 10.1002/mma.2935.

[46] Y.-S. Zhang, F. Zhang, B.-Z. Li, and R. Tao, "Fractional domain varying-order differential denoising method," *Opt. Eng.*, vol. 53, no. 10, Apr. 2014, Art. no. 102102.

[47] X. Yin, S. Zhou, and M. A. Siddique, "Fractional nonlinear anisotropic diffusion with p-Laplace variation method for image restoration," *Multimedia Tools Appl.*, vol. 75, no. 8, pp. 4505–4526, Apr. 2016.

[48] W. Zhang, J. Li, and Y. Yang, "Spatial fractional telegraph equation for image structure preserving denoising," *Signal Process.*, vol. 107, pp. 368–377, Feb. 2015.

[49] Y.-F. Pu, N. Zhang, Y. Zhang, and J.-L. Zhou, "A texture image denoising approach based on fractional developmental mathematics," *Pattern Anal. Appl.*, vol. 19, no. 2, pp. 427–445, May 2016.

[50] Y.-F. Pu, P. Siarry, A. Chatterjee, Z.-N. Wang, Z. Yi, Y.-G. Liu, J.-L. Zhou, and Y. Wang, "A fractional-order variational framework for retinex: Fractional-order partial differential equation-based formulation for multi-scale nonlocal contrast enhancement with texture preserving," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1214–1229, Mar. 2018.

[51] Y.-F. Pu, N. Zhang, Z.-N. Wang, J. Wang, Z. Yi, Y. Wang, and J.-L. Zhou, "Fractional-order Retinex for adaptive contrast enhancement of under-exposed traffic images," *IEEE Intell. Transp. Syst. Mag.*, early access, Feb. 7, 2019, doi: 10.1109/MITS.2018.2889706.

[52] Z. Guo, W. Yao, J. Sun, and B. Wu, "Nonlinear fractional diffusion model for deblurring images with textures," *Inverse Problems Imag.*, vol. 13, no. 6, pp. 1161–1188, 2019.

[53] Z. Ren, "Fractional-order bidirectional diffusion for image upsampling," *J. Electron. Imag.*, vol. 21, no. 2, May 2012, Art. no. 023006.

[54] X. Yin, S. Chen, L. Wang, and S. Zhou, "Fractional-order difference curvature-driven fractional anisotropic diffusion equation for image super-resolution," *Int. J. Model., Simul., Sci. Comput.*, vol. 10, no. 1, Feb. 2019, Art. no. 1941012.

[55] J. Zhang and K. Chen, "A total fractional-order variation model for image restoration with nonhomogeneous boundary conditions and its numerical solution," *SIAM J. Imag. Sci.*, vol. 8, no. 4, pp. 2487–2518, 2015.

[56] H. Li, Z. Yu, and C. Mao, "Fractional differential and variational method for image fusion and super-resolution," *Neurocomputing*, vol. 171, pp. 138–148, Jan. 2016.

[57] A. Ullah, W. Chen, and M. A. Khan, "A new variational approach for restoring images with multiplicative noise," *Comput. Math. Appl.*, vol. 71, no. 10, pp. 2034–2050, May 2016.

[58] Y.-F. Pu, "Fractional-order euler-Lagrange equation for Fractional-order variational method: A necessary condition for Fractional-order fixed boundary optimization problems in signal processing and image processing," *IEEE Access*, vol. 4, pp. 10110–10135, 2016.

[59] A. Abirami, P. Prakash, and K. Thangavel, "Fractional diffusion equation-based image denoising model using CN–GL scheme," *Int. J. Comput. Math.*, vol. 95, nos. 6–7, pp. 1222–1239, Jul. 2018.

[60] M. Chen, Y.-F. Pu, and Y.-C. Bai, "A fractional-order variational residual CNN for low dose CT image denoising," in *Intelligent Computing Theories and Application*, D.-S. Huang, V. Bevilacqua, and P. Premaratne, Eds. Cham, Switzerland: Springer, Jul. 2019, pp. 238–249.

[61] X. Jia, S. Liu, X. Feng, and L. Zhang, "FOCNet: A fractional optimal control network for image denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6047–6056.

[62] R. H. Chan and H.-X. Liang, "Truncated fractional-order total variation model for image restoration," *J. Oper. Res. Soc. China*, vol. 7, no. 4, pp. 561–578, Dec. 2019.

[63] A. Kumar, M. O. Ahmad, and M. N. S. Swamy, "A framework for image denoising using first and second order fractional overlapping group sparsity (HF-OLGS) regularizer," *IEEE Access*, vol. 7, pp. 26200–26217, 2019.

[64] J.-J. Mei, Y. Dong, and T.-Z. Huang, "Simultaneous image fusion and denoising by using fractional-order gradient information," *J. Comput. Appl. Math.*, vol. 351, pp. 212–227, May 2019.

[65] Q. Wang, J. Ma, S. Yu, and L. Tan, "Noise detection and image denoising based on fractional calculus," *Chaos, Solitons Fractals*, vol. 131, Feb. 2020, Art. no. 109463.

[66] M. R. Chowdhury, J. Zhang, J. Qin, and Y. Lou, "Poisson image denoising based on fractional-order total variation," *Inverse Problems Imag.*, vol. 14, no. 1, pp. 77–96, 2020.

[67] X. Yang and B. Guo, "Fractional-order tensor regularisation for image inpainting," *IET Image Process.*, vol. 11, no. 9, pp. 734–745, Sep. 2017.

[68] J. Zhang and K. Chen, "Variational image registration by a total fractional-order variation model," *J. Comput. Phys.*, vol. 293, pp. 442–461, Jul. 2015.

[69] S. G. Bardeji, I. N. Figueiredo, and E. Sousa, "Optical flow with fractional order regularization: Variational model and solution method," *Appl. Numer. Math.*, vol. 114, pp. 188–200, Apr. 2017.

[70] P. Ghamisi, M. S. Couceiro, J. A. Benediktsson, and N. M. F. Ferreira, "An efficient method for segmentation of images based on fractional calculus and natural selection," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12407–12417, Nov. 2012.

[71] P. Ghamisi, M. S. Couceiro, F. M. L. Martins, and J. A. Benediktsson, "Multilevel image segmentation based on fractional-order darwinian particle swarm optimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, pp. 2382–2394, May 2014.

[72] F. Guo, H. Peng, B. Zou, R. Zhao, and X. Liu, "Localisation and segmentation of optic disc with the fractional-order darwinian particle swarm optimisation algorithm," *IET Image Process.*, vol. 12, no. 8, pp. 1303–1312, Aug. 2018.

[73] S. Didas, B. Burgeth, A. Imiya, and J. Weickert, "Regularity and scale-space properties of fractional high order linear filtering," in *Proc. Int. Conf. Scale-Space Theories Comput. Vis.*, in Lecture Notes in Computer Science. Berlin, Germany: Springer, 2005, pp. 13–25.

[74] P. D. Romero and V. F. Candela, "Blind deconvolution models regularized by fractional powers of the Laplacian," *J. Math. Imag. Vis.*, vol. 32, no. 2, pp. 181–191, Oct. 2008.

[75] P. D. Tafti, R. Delgado-Gonzalo, A. F. Stalder, and M. Unser, "Fractal modelling and analysis of flow-field images," in *Proc. IEEE Int. Symp. Biomed. Imaging, From Nano Macro*, Apr. 2010, pp. 49–52.

[76] P. Gatto and J. S. Hesthaven, "Numerical approximation of the fractional Laplacian via *hp*-finite elements, with an application to image denoising," *J. Sci. Comput.*, vol. 65, pp. 249–270, Oct. 2015.

[77] H. Antil and S. Bartels, "Spectral approximation of fractional PDEs in image processing and phase field modeling," *Comput. Methods Appl. Math.*, vol. 17, no. 4, pp. 661–678, Jan. 2017.

[78] H. Antil, Z. Di, and R. Khatri, "Bilevel optimization, deep learning and fractional Laplacian regularization with applications in tomography," *Inverse Problems*, 2020, doi: 10.1088/1361-6420/ab80d7.

[79] B. Maundy, A. S. Elwakil, and T. J. Freeborn, "On the practical realization of higher-order filters with fractional stepping," *Signal Process.*, vol. 91, no. 3, pp. 484–491, Mar. 2011.

[80] T. J. Freeborn, B. Maundy, and A. Elwakil, "Second order approximation of the fractional Laplacian operator for equal-ripple response," in *Proc. 53rd IEEE Int. Midwest Symp. Circuits Syst.*, Aug. 2010, pp. 1173–1176.

[81] A. M. AbdelAty, A. S. Elwakil, A. G. Radwan, C. Psychalinos, and B. J. Maundy, "Approximation of the fractional-order Laplacian s^α as a weighted sum of first-order high-pass filters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 65, no. 8, pp. 1114–1118, Aug. 2018.

[82] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley, MA, USA: Cambridge Press, 1996.

[83] G. Strang, "The discrete cosine transform," *SIAM Rev.*, vol. 41, no. 1, pp. 135–147, 1999.

[84] V. Britanak, P. C. Yip, and K. R. Rao, "Discrete cosine and sine transforms: General properties, fast algorithms and integer approximations," in *Discrete Cosine and Sine Transforms*, V. Britanak, P. C. Yip, and K. R. Rao, Eds. New York, NY, USA: Academic, 2006, pp. 16–50.

[85] G. Deng, "Fast algorithm for zero-phase linear filter using discrete cosine transform," *Electron. Lett.*, vol. 55, no. 10, pp. 621–623, May 2019.

[86] C.-C. Tseng and S.-L. Lee, "Computation of partial fractional derivative of digital image using discrete cosine transform," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 2828–2831.

[87] C.-C. Tseng, S.-C. Pei, and S.-C. Hsia, "Computation of fractional derivatives using Fourier transform and digital FIR differentiator," *Signal Process.*, vol. 80, no. 1, pp. 151–159, Jan. 2000.

[88] M. Kumar and T. K. Rawat, "Design of fractional order differentiator using type-III and type-IV discrete cosine transform," *Eng. Sci. Technol., Int. J.*, vol. 20, no. 1, pp. 51–58, Feb. 2017.

[89] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, "ℓ_1 trend filtering," *SIAM Rev.*, vol. 51, no. 2, pp. 339–360, 2009.

[90] B. Bruder, T.-L. Dao, J.-C. Richard, and T. Roncalli, "Trend filtering methods for momentum strategies," *SSRN Electron. J.*, 2011, doi: 10.2139/ssrn.2289097.

[91] R. J. Hodrick and E. C. Prescott, "Postwar U.S. Business cycles: An empirical investigation," *J. Money, Credit Banking*, vol. 29, no. 1, p. 1, Feb. 1997.

[92] W. K. Pratt, "Generalized Wiener filtering computation techniques," *IEEE Trans. Comput.*, vol. C-21, no. 7, pp. 636–641, Jul. 1972.

[93] B. Fischer and J. Modersitzki, "Curvature based image registration," *J. Math. Imag. Vis.*, vol. 18, no. 1, pp. 81–85, 2003.

[94] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[95] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[96] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.

[97] D. Marr and E. Hildreth, "Theory of edge detection," *Proc. Roy. Soc. London B. Biol. Sci.*, vol. 207, pp. 187–217, Feb. 1980.

[98] R. Gonzalez, *Digital Image Processing*. New York, NY, USA: Pearson, 2018.

[99] S. Osher and L. I. Rudin, "Feature-oriented image enhancement using shock filters," *SIAM J. Numer. Anal.*, vol. 27, no. 4, pp. 919–940, Aug. 1990.

[100] R. Duits, L. Florack, J. de Graaf, and B. ter Haar Romeny, "On the axioms of scale space theory," *J. Math. Imag. Vis.*, vol. 20, no. 3, pp. 267–298, May 2004.

• • •