Journal of Mathematical Imaging and Vision manuscript No.
(will be inserted by the editor)

Two Enhanced Fourth Order Diffusion Models for Image Denoising

Patrick Guidotti · Kate Longo

Received: date / Accepted: date

Abstract This paper presents two new higher order diffusion models for removing noise from images. The models employ fractional derivatives and are modifications of an existing fourth order partial differential equation (PDE) model which was developed by You and Kaveh as a generalization of the well-known second order Perona-Malik equation. The modifications serve to cure the ill-posedness of the You-Kaveh model without sacrificing performance. Also proposed in this paper is a simple smoothing technique which can be used in numerical experiments to improve denoising and reduce processing time. Numerical experiments are shown for comparison.

Keywords Nonlinear Diffusion · Fractional Derivatives · Image Denoising · Fourth Order

1 Introduction

Noise is an unavoidable component of digital image acquisition, and as such, noise removal has become a fundamental task of image processing. The problem is formulated by considering an image as a collection of intensity data about N pixels,

$$u(i) = v(i) + n(i), \qquad i = 1, ..., N.$$

The authors of this research were supported by the National Science Foundation under award number DMS-0712875.

Patrick Guidotti
Department of Mathematics
University of California at Irvine
Irvine, CA 92691-3875, USA
E-mail: gpatrick@math.uci.edu

Kate Longo
E-mail: klongo@math.uci.edu

In this paper, only grey scale images are considered, so u is a scalar valued function taking on quantized integer values between 0 and 255. u is an observed image, consisting of a "true" image v polluted with noise n. To denoise an image is to recover v from the observed image u, a theoretically impossible task. Indeed fine details of an image can be indistinguishable from noise and all known denoising methods can cause various degrees of blurring, staircasing, and other artifacts.

Noise reduction, in particular PDE-based noise reduction, has been a subject of much research since a seminal paper by Perona and Malik in 1990 [29] which introduced the then novel paradigm of using nonlinear diffusions for the task of denoising images. Their method improves upon the technique of linear diffusion used previously (and introduced by Witkin [33]) by reducing the diffusivity at locations of the image where edges are found by an appropriate edge detector. The latter is typically implemented by measuring the gradient $|\nabla u|$ of the image. The Perona-Malik model is based on the equation

$$u_t - \nabla \cdot (g(|\nabla u|)\nabla u) = 0, \tag{1.1}$$

where $g(\cdot)$ is a diffusivity such as

$$g(s) = \frac{1}{1 + c^2 s^2}\,,\ c > 0\,. \tag{1.2}$$

Discretizations of (1.1) have proved to be effective denoising tools. However, two significant problems have been observed. Firstly, equation (1.1) has been shown to be mathematically ill-posed [22], which makes it impossible to prove theoretical results about the behavior of algorithms based on that equation. Secondly, in numerical experiments, such algorithms tend to produce artifacts at edges, such as staircasing (when false edges are introduced) [22,32] and blocky, cartoonish effects

(when smooth edges are sharpened into corners) [34]. The literature abounds in attempts to cure these problems, including regularizing the edge detector $|\nabla u|$ [1,2, 5,8,18], and generalizing the ideas of Perona and Malik to higher order equations [30,6,14,31,20,21,34,25]. Of note is a fourth order generalization of (1.1) proposed by You and Kaveh which is the focus of this paper. Numerical experiments in the aforementioned papers and others show that many fourth order denoising models are successful at avoiding the staircasing and cartoonishness characteristic of second order models, and excel at preserving a natural look to images such as human faces. This success often comes at the cost of good performance in flat regions of images, and many fourth order denoising methods leave a kind of splotchy artifact in such regions (see for example experiments in [20,21, 25,10]). Additionally, equation (2.7) has been noted to produce a speckle artifact, which is removed by the authors of [34] in a post-processing step. Improvements of models such as (2.7) can be gained through the contributions of this paper. Proposed in this paper are two new PDE models which transplant ideas from regularized second order models [18] to the fourth order You-Kaveh model and which offer analytical as well as practical benefits. The new models utilize a fractional Laplacian operator and a fractional gradient, respectively, to detect edges and limit blurriness.

Processing time can be reduced and denoising results improved through the use of a novel smoothing step involving a simple manipulation of parameters. The proposed equations are discretized with a spectral scheme. With the combination of this numerical scheme and smoothing technique, we are able to obtain improved denoising results compared to previous experiments shown with (2.7) and its variants. Specifically, the splotchy artifact characteristic of fourth order denoising methods can be alleviated without sacrificing edge preservation, and speckling is not observed at all, with either (2.7) or the proposed modifications. The smoothing technique is not particular to the proposed fourth order equations; experiments are shown to demonstrate the ability of the smoothing step to improve the performance of a previously proposed second order PDE for noise removal, as well as to dramatically reduce the necessary processing time.

The paper is organized as follows. Key results from the literature about second and fourth order diffusions and their regularizations are summarized in section 2. In section 3, the behavior of the You-Kaveh model and the effects of various parameters are analyzed, and the smoothing technique is proposed and its ability to improve performance is demonstrated. Section 4 details the two proposed models, their performance is illustrated with numerical experiments in section 5, and the paper is concluded in section 6.

2 Background

Use of diffusion PDEs for noise removal can be traced back to the scale space method introduced by Witkin in 1983 [33]. This method smoothes a noisy image by convolving it with Gaussian kernels on a scale of variances. It is equivalent to considering a smoothed image as the solution to the linear heat equation

$$u_t = \nabla \cdot (\gamma \nabla u), \quad u(0) = u_0 \text{ (original image)}, \tag{2.1}$$

where the diffusion coefficient γ is constant. This is an effective smoothing method, but it cannot distinguish noise from edges, thus blurring the entire image. The method needs to be complemented by a second processing step which locates and reintroduces edges. In a 1990 paper [29], Perona and Malik propose their idea to preserve edges in the first step by replacing γ with an "edge detector", a nonlinear function which would inhibit diffusion across edges. They observe that edges in an image correspond to regions of high gradient, and thus consider $\gamma = g(|\nabla u|)$, where $g(\cdot)$ is chosen appropriately so as to slow diffusion (become small) when $|\nabla u|$ is large. One possibility for such a function suggested by Perona and Malik is given in (1.2). In their paper, they propose discrete equations which can be incorporated into a continuous PDE model for image processing. The Perona-Malik model can be viewed as an approximation of equation (1.1), which is therefore commonly referred to as the Perona-Malik equation in the literature. Analytically, PDE models based on (1.1) have been found to be ill-posed [22]. Despite, or rather because of this, the Perona-Malik model preserves sharp edges well, but still doesn't escape some practical drawbacks, including the creation of artifacts such as staircasing and blocky effects [22,32,34].

Attempts to overcome the problematic issues of Perona-Malik can be broadly classified into two categories: second order regularizations and relaxations of Perona-Malik, and higher order diffusions. Spatial regularizations have been considered, which involve smoothing the argument in the nonlinear edge detector function g with a C^∞ kernel G_σ, typically a Gaussian. See, for example, [1,8,27]. This leads to the equation

$$u_t - \nabla \cdot \big(g(|\nabla G_\sigma * u|)\nabla u\big) = 0\,, \text{ in } \Omega \text{ for } t > 0. \tag{2.2}$$

Other authors have proposed models in which the nonlinearity is regularized with a space-time convolution, namely [11,12,28]. Purely temporal regularizations are considered in [2,4,5].

An irony of the Perona-Malik model is that its greatest strengths, superior edge detection capabilities, are made possible by the nonlinearity $g(|\nabla u|)$, which is also responsible for its weaknesses, ill-posedness and artifacts. Consequently, regularizing this nonlinearity can cure ill-posedness, but often at the cost of good edge detection. This is particularly true of regularizations of the type in (2.2). Reintroducing a smooth kernel G_σ results in blurring, exactly what Perona and Malik try to avoid.

Guidotti in [18,17,16] proposes two different, "milder" regularizations characterized by the use of fractional derivatives in the edge detector function g. Fractional derivatives can be defined by first observing that

$$\partial_z = \mathcal{F}_z^{-1}\mathrm{diag}[(2\pi i n_z)_{n_z\in\mathbb{Z}}]\mathcal{F}_z,$$

where $\mathcal{F}_z$ is the partial Fourier transform with respect to $z = x, y$. Then the fractional partial derivative is defined by

$$\partial_z^\rho = \mathcal{F}_z^{-1}\mathrm{diag}\big[\big((2\pi n_z)^\rho e^{i\rho\frac{\pi}{2}\mathrm{sign}(n)}\big)_{n_z\in\mathbb{Z}}\big]\mathcal{F}_z,$$

for any $\rho \in \mathbb{R}^+$. The fractional gradient is finally given by

$$\nabla^\rho = \begin{bmatrix}\partial_x^\rho \\ \partial_y^\rho\end{bmatrix}.$$

Exponents of the positive definite Laplacian operator $-\Delta$ with periodic boundary conditions can be defined through its symbol:

$$(-\Delta)^\rho = \mathcal{F}^{-1}\mathrm{diag}[\big(4\pi^2|n|^{2\rho}\big)_{n\in\mathbb{Z}^2}]\mathcal{F},$$

for any $\rho \in \mathbb{R}^+$.

Fractional derivatives have been shown to be effective tools both for regularization [13] (where fractional diffusions are considered) and for edge detection [26] (where one dimensional signals are considered). The models proposed by Guidotti are based on the equations

$$u_t - \nabla\cdot\big(g(|\nabla^{1-\varepsilon}u|^2)\nabla u\big) = 0 \tag{2.3}$$

and

$$u_t - g\big([(-\Delta)^{1-\varepsilon}]^2\big)\Delta u = 0, \tag{2.4}$$

coupled with appropriate initial and boundary conditions, where $\varepsilon \in (0,1)$. It is shown in [17,16] that (2.4) and (2.3) are locally well-posed and that (2.3) admits characteristic functions of smooth sets as stationary solutions. Additionally, numerical experiments suggest that these two models produce significantly less staircasing than does (1.1) or other second order diffusion equations (such as TV minimization). Blocky effects (conferring a "cartoonish" feel to processed images), however, remain. See figures 3 and 9 for examples of the denoising effects of (2.3). Since $|\nabla^{1-\epsilon}u|$ is itself an effective edge detector, blurring is not observed, as it is with other regularized models such as (2.2).

Other researchers have opted to modify the Perona-Malik model by generalizing it to higher orders instead of by applying regularizations. Fourth order PDEs from the literature include those proposed in [31,34,6,25,20,21]. One of most frequently cited is the model derived by You and Kaveh in [34] from a variational formulation. It is shown in [35] that solving (1.1) is equivalent to minimizing the first order functional

$$\int_\Omega f(|\nabla u|)d\Omega, \tag{2.5}$$

where $f'(s) = sg(s)$, via the method of gradient descent. When the diffusivity function is picked as in (1.2), the Perona-Malik model is equivalent to minimizing

$$\frac{1}{2c^2}\int_\Omega \log\left(1 + c^2s^2\right)d\Omega.$$

You and Kaveh propose minimizing instead the second order functional

$$\int_\Omega f(\Delta u)d\Omega. \tag{2.6}$$

The gradient descent method performed on this second order functional yields a fourth order PDE

$$u_t = -\Delta\big(g(\Delta u)\Delta u\big). \tag{2.7}$$

This is a fourth order nonlinear diffusion equation in which diffusion is slowed near the location of edges. However, edges are identified by the Laplacian $|\Delta u|$, instead of the gradient $|\nabla u|$ used in (1.1). The rationale put forth in [34] for the use of fourth order equations with second order edge detection is that such equations admit a more flexible class of equilibria than second order equations (for which characteristic functions are stationary). This allows solutions to transiently relax to smoother states where sharp but smooth gradients are preserved as opposed to being further sharpened into jumps, thus avoiding the blocky effects associated with (1.1). The fourth order (2.7) exhibits similar analytical behavior to (1.1), and is likely also ill-posed [19].

One could consider instead the equation

$$u_t + \Delta\big[g(|\nabla u|)\Delta u\big] = 0, \tag{2.8}$$

which couples the fourth order diffusion of (2.7) with the edge detector of (1.1). It is an important observation that reversion to edge detection via the gradient as in (2.8) in the context of fourth order diffusions does not coincide with returning to a second order model.

Characteristic functions are in fact not equilibria for (2.8). Hajiaboli considers a model similar to (2.8) in [20] but with a modified diffusivity function g. There are many other approaches to noise removal which are not touched upon here. See [7] for a survey of other methods. Of particular interest to this discussion are higher order variational methods, for example, [9,10,3, 24]. The methods proposed here continue to compare well in the wider arena which also includes these other methods.

3 Numerical Considerations

3.1 Scheme

Equations such as (2.7) are commonly discretized using finite difference methods (see for example [34,20]). We instead employ a spectral method so that fractional derivatives, defined through the Fourier transform, can be easily incorporated into these models. The scheme used is the Krylov subspace spectral (KSS) method with block Lanczos iteration developed by Lambers in [23]. This is an explicit time-marching method which possesses a high order of accuracy as well as the stability of implicit methods. It is also useful for discretizing (2.3) and (2.4), second order PDEs using fractional derivatives.

The high accuracy of the KSS method comes from the fact that it computes each Fourier coefficient individually using an approximation that is, in some sense, optimal for it. Consequently, this method avoids the regularizing effects of the finite difference method, providing a more accurate representation of the behavior of the equations in question. Our experiments on (2.7), however, are largely comparable to those in [34,20]. One exception to this is a speckle artifact observed by You and Kaveh in experiments on (2.7), as well as by Hajiaboli on the same equation. You and Kaveh remove this artifact in a post-processing step in [34]. Our own experiments on (2.7) discretized using the KSS method do not reveal this speckle artifact, suggesting that the artifact is associated with the finite difference discretization of (2.7), rather than with the equation itself. This idea is strengthened by the results in [3], in which (2.7) is discretized with a pseudospectral method, and the speckle artifact is not observed. The authors do observe an artifact they refer to as speckling, but it is more akin to the effect of the bottom left image in figure 2, where dots appear all over the image. In the results shown in [34,20], speckles are concentrated along the edges of an image.

The KSS method is applied to tiles of an image of size 2^m pixels square, $m \in \mathbb{N}$. A border of 16 pixels is reflected across each side of the tile to emulate periodic boundary conditions and to avoid any boundary effects, and this border is removed after denoising. A very small time step is required with this method in order to avoid artifacts, in particular for fourth order equations and for larger values of m. This may necessitate a very long processing time to get sufficient noise removal, but the processing time can be shortened by processing an image in several stages with different parameters (see section 4).

In the following, unless otherwise noted, all experiments are performed on 256 pixel square images, and the diffusivity function g is chosen as in (1.2).

3.2 Noise Removal Versus Blurring

In any diffusion based noise removal process there is a trade-off between sufficient noise removal and the preservation of edges. Diffusion in (2.7) is controlled by the diffusivity function g. Both of the diffusivity functions proposed by Perona and Malik (i.e., (1.2)) depend on a parameter c which determines the size of edges, that is, a location of an image is considered to be an edge if $|\nabla u|$ (or another choice of edge detector) is greater than $1/c$ at that location, and it is not an edge if the edge detector is less than $1/c$. Consequently, this parameter serves as a sort of measure of the trade-off between noise removal and edge preservation. When c is too small, noise and edges are both smoothed quickly; when c is too large, too much noise is mistaken for edges and never smoothed away.

However, with fourth order equations such as (2.7), within the range of medium-sized c's, a splotchy artifact comes into play. When c is medium-small, after a small number of iterations most noise is eliminated and edges are well preserved, but low-frequency components of the noise are still present, leaving a splotchy effect over flat areas of the image. Many more iterations are required to smooth these splotches away, but c is now too small to preserve edges in the face of so much smoothing. When c is chosen medium-large, splotches can be smoothed away while still preserving edges, but this requires a great number of iterations, at a high computational cost.

Similar behavior can also be observed with second order equations such as (1.1) and (2.3), but only at time scales smaller than are typically used for discretization. Thus, splotchiness is not usually observed with second order equations. Fourth order equations often need to be discretized with a smaller time step than is required for second order equations. The time scale frequently used for fourth order equation is small enough to expose the splotchy behavior described above, and hence

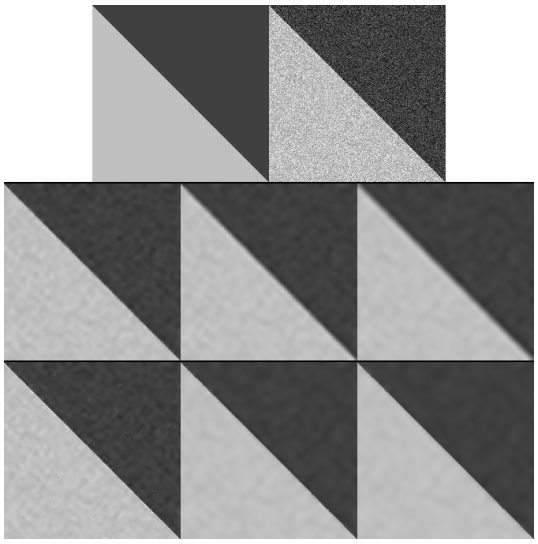

Fig. 1 A noisy two-tone image is denoised with equation (2.7) with a smaller c for 500, 1000, and 2000 iterations (second row), and with a larger c for 1000, 3000, and 5000 iterations (third row).

this artifact is observed in many experiments on fourth order models elsewhere in the literature, in particular, [34, 20, 21, 25].

This phenomenon is illustrated in figure 1 on a simple image with 20% Gaussian noise, denoised with (2.7) with two different values of c. The first row shows the original clean and noisy images. The second row, left to right, shows the evolution under (2.7) with a smaller $c = 10^{-6}$ and time step $h = 5 * 10^{-12}$, for 500, 1000, and 2000 iterations. The third row, left to right, shows the evolution with the larger $c = 3 * 10^{-6}$ and the same time step for 1000, 3000, and 5000 iterations. With the smaller c, the edge is blurred before noise is sufficiently removed. With the larger c, the edge is preserved for a large number of iterations, allowing for the splotchiness to be mostly smoothed away.

The edge in the simple image in figure 1 is able to withstand the large number of iterations necessary to eliminate splotches because it is very sharp. Smaller edges in natural images are not so robust. Splotchiness can be lessened, but at the cost of the blurring of fine details of an image, so it may be necessary to retain some splotchiness to avoid excessive blurring.

3.3 Artifacts and Proposed Smoothing Step

The numerical performance of (2.7) is sensitive to the choice of parameters. Artifacts appear when the parameter c and the time step are too large. This is unfortunate, since a small c can lead to blurriness, and a small time step can lead to a long processing time.

A small choice of c is not able to sufficently remove noise without blurring edges. However, by using a small c as well as a very small time step, high-frequency components of noise can be removed in just a few iterations with little blurring of edges (shown in the middle left image in figure 2). This slightly smoothed image, which is not noisy so much as it is splotchy, is less prone to artifacts than the initial noisy image. We then continue the denoising process on the smoothed image, using a larger time step and a larger c to preserve edges. If this larger c was used on the initial image, it would need to be used with a smaller time step to avoid artifacts, necessitating a large number of iterations to achieve sufficient noise removal. Additionally, since high frequency components of noise are removed in the first step, the remaining noise can be more effectively removed in the second step, and in less time.

Figure 2 compares the denoising effects of (2.7) on a natural image with and without this regularizing step. The first row shows the original clean image of Lena, and the image corrupted with 20% Gaussian noise. The second row shows the two steps of the smoothing process: first the noisy image is smoothed with a small $c = 10^{-8}$ and time step $h = 5 * 10^{-12}$ for 10 iterations; the resulting smoothed image is then denoised with a larger $c = 2 * 10^{-5}$ and a larger time step $h = 10^{-10}$ for 400 iterations. The third row shows denoising results without the smoothing step: the image denoised with $c = 3 * 10^{-6}$ and the larger time step $h = 10^{-10}$ for 200 iterations; and the image denoised with the same c but the smaller time step $h = 5 * 10^{-12}$ for 1000 iterations. Edge preservation in the two images in the middle and bottom right is comparable, but the image denoised with the smoothing step requires many fewer iterations (410 versus 1000), and shows less noise and splotchiness in flat areas. When a smaller c is used with the larger time step, as in the image in the bottom left, noise is not effectively removed before blurring sets in.

This smoothing technique can be applied to other denoising methods as well, such as second order PDEs. Figure 3 compares the effects of equation (2.3) with and without an intial smoothing step. On the left is an image of Lena corrupted with 20% Gaussian noise and denoised with (2.3) with $\varepsilon = 0.1$, $c = .003$, and time step $h = 5 * 10^{-6}$ for 400 iterations. On the right is the same noisy image denoised with the same equation first with a smoothing step using parameters $h = 5 * 10^{-7}$ and $c = 5 * 10^{-7}$ for 10 iterations, and followed by 100 iterations using the same parameters as with the image on the left. This discretization of this equation is not sensitive to parameter choice as (2.7) is, so we can utilize a large choice of c even without the smoothing step. As is shown, the smoothing step is still useful for im-

Fig. 2 Comparison of denoising effects of (2.7) with and without a smoothing step. First row (left to right): clean image, image with 20% Gaussian noise. Second row (left to right): noisy image smoothed first with a very small c and small time step for 10 iterations, this smoothed image denoised with a larger time step and larger c for 400 iterations. Third row (left to right): image denoised with the smaller c and larger time step for 200 iterations, image denoised with the smaller c and smaller time step for 1000 iterations.

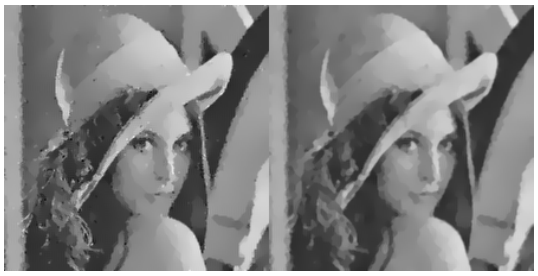

Fig. 3 Comparison of denoising effects of (2.3) with and without a smoothing step. The initial clean and noisy images are shown in figure 2.

proving performance (at the cost of the clarity of some finer features) and for dramatically reducing processing time (110 iterations versus 400).

3.4 Quantitative Evaluation

The parameters used in the experiments in section 5 (i.e., time step h, c, and stopping time) have been chosen by trial and error in order to get the best denoising results on the basis of appearance. While this is hardly an exact way to measure results, ultimately the important feature is the appearance of an image, even though it is a difficult task to find a quantity which can measure it reliably.

	Noisy Image	2nd Order	4th Order (1)	4th Order (2)
PSNR	27.83	**35.78**	34.57	35.24
SNR	2.65	**6.02**	5.63	5.72
ℓ_1-error	2.93e3	2.38e3	**1.67e3**	3.39e3
ℓ_2-error	2.32e3	**0.93e3**	1.07e3	0.99e3

Table 1 Quantitative values are shown for the three denoised images shown in figure 4, as well as for the original noisy image.

Other characteristics of a denoised image may be measured to quantify the success of the denoising method. These characteristics include peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), ℓ_1-error, and ℓ_2-error. All of these quantities measure a comparison between the denoised image and the original clean image. A stopping time could be determined by optimizing one of these quantities (as is often done, see, e.g., [20, 15]). However, it should not be assumed that the clean image is known, and the stopping time for a denoising model should be independent of the clean image.

Furthermore, these quantities do not necessarily give an accurate measure of the quality of a denoised image, as they may not adequately penalize certain undesirable properties, such as excessive noise, splotchiness, or blurriness. Table 1 lists the PSNR, SNR, ℓ_1-error and ℓ_2-error for three different denoising results on an image of Lena corrupted by 20% Gaussian noise, as well as for the initial noisy image. The three images are obtained by denoising with:

- (2.3) with $\varepsilon = 0.1$, $c = .003$, and time step $h = 5 * 10^{-6}$ for 300 iterations ("2nd Order");
- (2.7) with $c = 10^{-7}$ and time step $h = 5 * 10^{-12}$ for 10 iterations ("4th Order (1)"); and
- (2.7) with the previous parameters, followed by $c = 5 * 10^{-5}$ and time step $h = 10^{-11}$ for 350 iterations ("4th Order (2)").

The denoised images are shown in figure 4. The "best" value for each quantity is shown in the table in bold (a larger PSNR or SNR is better, and a smaller ℓ_1-error or ℓ_2-error is better). The best values correspond either to "2nd Order", which still contains transient noise-related artifacts, or to the "4th Order (1)", which contains excessive splotchiness. "4th Order (2)" has, arguably, the best appearance, but this is not indicated by any of these measurements.

Fig. 4 The Lena image with 20% Gaussian noise added is denoised three ways. First row (left to right): "2nd Order", denoised with (1.1); "4th Order (1)", denoised with (2.7) for a short time; and "4th Order (2)", denoised with (2.7) for a long time. Second row: the method noise for each image. The initial clean and noisy images are the same as in figure 2

Method noise, the difference between the initial noisy image and the denoised image, is a gauge of the success of a denoising method that does not depend on the clean image. Ideally the method noise should resemble noise, and should not exhibit features of the original image. The method noise for the three denoised images is shown in figure 4. Method noise is a good indicator of blurriness in a denoised image, as when edges are smoothed out, they show up in the method noise. However, it may not be a good indicator of the best denoising results, as sharp edges may need to be partly sacrificed for effective noise removal.

4 Proposed Models

We propose two modifications to (2.7), each of which uses a different edge detector. The two new models are

$$\begin{cases} u_t + \Delta\Big(g\big(\big[(-\Delta)^{1-\varepsilon}u\big]^2\big)\Delta u\Big) = 0, & \text{in } \Omega \text{ for } t > 0, \\ u \text{ periodic}, & \text{for } t > 0, \\ u(0) = u_0, & \text{in } \Omega \text{ for } t = 0, \end{cases} \tag{4.1}$$

and

$$\begin{cases} u_t + \Delta\Big(g\big(|\nabla^{1-\varepsilon}u|^2\big)\Delta u\Big) = 0, & \text{in } \Omega \text{ for } t > 0, \\ u \text{ periodic}, & \text{for } t > 0, \\ u(0) = u_0, & \text{in } \Omega \text{ for } t = 0, \end{cases} \tag{4.2}$$

where Ω is the unit square and $\varepsilon \in (0, 1)$. For small ε, equation (4.1) offers at best a modest improvement to the performance of (2.7), but unlike (2.7), it can be shown to possess a unique short time solution [19]. While the the Laplacian $|\Delta u|$ in (2.7) is effective at detecting edges, the gradient $|\nabla u|$ is generally a better edge detector, especially on images with very sharp edges. It is therefore natural to replace the edge detector in (2.7) with the gradient, yielding equation (2.8). It is discussed in section 3.3 that (2.7) is prone to artifacts when a large time step and large values of c are chosen. Equation (2.8) is even more susceptible to artifacts, and an even smaller time step is required to avoid them, resulting in a longer processing time. Equation (4.2) uses instead a regularized gradient to detect edges, which alleviates some of the sensitivity to the choice of parameters. (4.2) can therefore be used with a larger time step than can (2.8), shortening processing time, while still providing effective edge preservation.

5 Numerical Experiments

It is shown in section 3.3 that the proposed smoothing technique can improve the performance of some PDE denoising models from the literature, including both second and fourth order models. In this section we illustrate the behavior of the two proposed fourth order equations implemented with the smoothing step, and compare the behavior with that of (2.7). The strength of equation (4.2) compared to the second order (2.3) is also demonstrated.

The following parameters are used in figures 5 and 9:

- Equation (2.7): Smoothed with time step $h = 5 * 10^{-12}$ and $c = 10^{-8}$ for 10 iterations, followed by $h = 10^{-10}$ and $c = 2 * 10^{-5}$.
- Equation (4.1), $\varepsilon = 0.1$: Smoothed with $h = 5 * 10^{-12}$ and $c = 10^{-7}$ for 5 iterations, followed by $h = 10^{-10}$ and $c = 5 * 10^{-5}$.
- Equation (4.2), $\varepsilon = 0.1$: Smoothed with $h = 5 * 10^{-12}$ and $c = 10^{-5}$ for 5 iterations, followed by $h = 5 * 10^{-11}$ and $c = 0.006$.
- Equation (2.3), $\varepsilon = 0.1$: Smoothed with $h = 5*10^{-7}$ and $c = 5 * 10^{-7}$ for 10 iterations, followed by $h = 5 * 10^{-6}$ and $c = 0.003$.

The number of iterations used in the second step varies depending on the image and the amount of noise.

Figures 5, 6, and 7 show the evolution of the three fourth order models – (2.7), (4.1), and (4.2) – on several noisy images, after 300 and 500 iterations following the smoothing step. Figure 8 shows contour plots of images denoised with the proposed models. The images denoised with (4.2) show somewhat improved edge preservation compared to the others, but they also exhibit somewhat blocky behavior. The images denoised with equations (2.7) and (4.1) show some cross-hatching artifacts which are not observed with (4.2). All three fourth order methods effectively remove splotches and other artifacts in flat regions. (These cross-hatching artifacts are also observed in experiments in [3] on (2.7) and on a modification of (2.7) involving fractional derivatives.

Fig. 5 Lena with 20% Gaussian noise denoised with (2.7) (first row), (4.1) (second row), and (4.2) (third row), at two different stopping times. The initial clean and noisy images are the same as in figure 2.

The model proposed in [3] is not closely related, mathematically speaking, to (4.1), but is in the same spirit.)

The proposed fourth order equations, like (2.7) and other fourth order models, improve upon second order models in the sense that they avoid the cartoonish look common to many images denoised by second order models. This benefit is particularly evident the the presence of large amounts of noise. Figure 9 contrasts the performance of the fourth order equation (4.2) and the second order (2.3) on images with 20% and 30% Gaussian noise added. The second row shows the denoising results with (2.3) after 100 and 200 iterations in the second step, and the third row shows the results with (4.2) after 400 and 800 iterations (the rest of the parameters are as given above). Results on the image with 20% noise are in the left column, and with 30% in the right column. The second order method requires few iterations to achieve satisfactory denoising results, but the fourth order method is better able to avoid a cartoonish appearance and preserves a more natural look.

Fig. 6 Cameraman with 20% Gaussian noise denoised with (2.7) (second row), (4.1) (third row), and (4.2) (fourth row), at two different stopping times. The initial clean and noisy images are shown in the first row.

6 Conclusion

The fourth order diffusion PDE (2.7) proposed in [34] has been analyzed, and a smoothing technique has been proposed which reduces preocessing time and improves the performance of this model and others like it. Two modifications of (2.7) have been proposed which utilize fractional derivatives for edge detection. Numerical experiments have been shown comparing the performance of the two new models with that of (2.7) as well as a second order method.

Acknowledgements The authors would like to express their gratitude to James Lambers for his assistance in the numerical aspect of this research.

Fig. 7 Peppers with 20% Gaussian noise denoised with (2.7) (second row), (4.1) (third row), and (4.2) (fourth row), at two different stopping times. The initial clean and noisy images are shown in the first row.

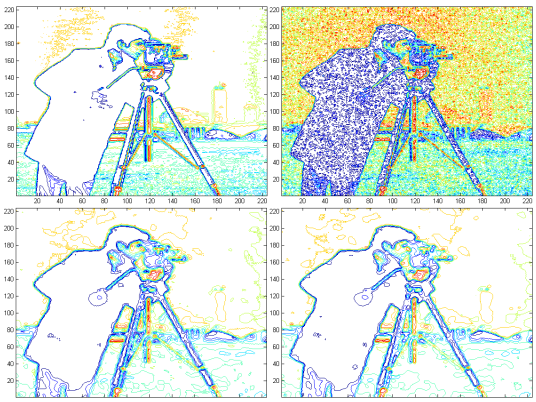

Fig. 8 Contour plots of the clean cameraman image (top left), the noisy image (top right), the image denoised with (4.1) (bottom left), and the image denoised with (4.2) (bottom right).

Fig. 9 Denoising results using (2.3) and (4.2) on an image with 20% (left column) and 30% (right column) Gaussian noise. First row: initial noisy images. Second row: results denoising with (2.3). Third row: results denoising with (4.2). The initial clean image can be seen in figure 2.

References

1. Alvarez, L., Lions, P.L., Morel, J.M.: Image selective smoothing and edge-detection by non-linear diffusion. II. SIAM J. Numer. Anal. **29**(3), 845–866 (1992)
2. Amann, H.: Time-Delayed Perona-Malik Problems. Acta Math. Univ. Comenianae **LXXVI**, 15–38 (2007)
3. Bai, J., Feng, X.C.: Fractional Order Anisotropic Diffusion for Image Denoising. IEEE Transactions on Image Processing **16**(10), 2492–2502 (2007)
4. Belahmidi, A.: Equations aux dérivées partielles appliquées à la restoration et à l'agrandissement des images. Ph.D. Thesis. Université Paris-Dauphine, Paris (2003)
5. Belahmidi, A., Chambolle, A.: Time-delay regularization of anisotropic diffusion and image processing. M2AN Math. Model. Numer. Anal. **39**(2), 231–251 (2005)
6. Bertozzi, A., Greer, J.: Low-Curvature Image Simplifiers: Global Regularity of Smooth Solutions and Laplacian Limiting Schemes. Communications on Pure and Applied Mathematics **LVII**, 0764–0790 (2004)
7. Buades, A., Coll, B., Morel, J.: A Review of Image Denoising Algorithms, with a New One. Multiscale Modeling and Simulations **4**(2), 490–530 (2005)
8. Catté, F., Lions, P.L., Morel, J.M., Coll, T.: Image selective smoothing and edge-detection by non-linear diffusion. SIAM J. Numer. Anal. **29**(1), 182–193 (1992)
9. Chan, T., Marquina, A., Mulet, P.: High-order total variation-based image restoration. SIAM J. Sci. Comput. **22**(2), 503–516 (electronic)

(2000). DOI 10.1137/S1064827598344169. URL http://dx.doi.org/10.1137/S1064827598344169
10. Chan, T.F., Esedoglu, S., Park, F.E.: A fourth order dual method for staircase reduction in texture extraction and image restoration problems. UCLA CAM report **05-28** (April 2005)
11. Chen, Y., Bose, P.: On the incorporation of time-delay regularization into curvature-based diffusion. J. Math. Imaging Vision **14**(2), 149–164 (2001)
12. Cottet, G.H., Ayyadi, M.E.: A volterra type model for image processing. IEEE Trans. Image Processing **7**, 292–303 (1998)
13. Didas, S., Burgeth, B., Imiya, A., Weickert, J.: Regularity and Scale-Space Properties of Fractional High Order Linear Filtering. In: Scale Space and PDE Methods in Computer Vision, *Lecture Notes in Computer Science*, vol. 3459. Springer Berlin/Heidelberg (2005)
14. Didas, S., Weickert, J., Burgeth, B.: Stability and local Feature Enhancement of Higher Order Nonlinear Diffusion Filtering. Pattern Recognition **3663**, 451–458 (2005)
15. Didas, S., Weickert, J., Burgeth, B.: Properties of higher order nonlinear diffusion filtering. J. Math. Imaging Vision **35**(3), 208–226 (2009). DOI 10.1007/s10851-009-0166-x. URL http://dx.doi.org/10.1007/s10851-009-0166-x
16. Guidotti, P.: A new Nonlocal Nonlinear Diffusion of Image Processing. Journal of Differential Equations **246**(12), 4731–4742 (2009)
17. Guidotti, P.: A New Well-posed Nonlinear Nonlocal Diffusion. Nonlinear Analysis Series A: Theory, Methods, and Applications **72**, 4625–4637 (2010)
18. Guidotti, P., Lambers, J.: Two New Nonlinear Nonlocal Diffusions for Noise Reduction. Journal of Mathematical Imaging and Vision **33**(1), 25–37 (2009)
19. Guidotti, P., Longo, K.: Well-Posedness for a Class of Fourth Order Diffusions for Image Processing. To appear in Nonlinear Differential Equations and Applications
20. Hajiaboli, M.: A Self-governing Hybrid Model for Noise Removal. In: Advances in Image and Video Technology, *Lecture Notes in Computer Science*, vol. 5414. Springer Berlin/Heidelberg (2008)
21. Hajiaboli, M.: An Anisotropic Fourth-Order Partial Differential Equation for Noise Removal. In: Scale Space and Variational Methods in Computer Vision, *Lecture Notes in Computer Science*, vol. 5567. Springer Berlin/Heidelberg (2009)
22. Kichenassamy, S.: The Perona-Malik paradox. SIAM J. Appl. Math. **57**(5), 1328–1342 (1997)
23. Lambers, J.V.: Enhancement of Krylov Subspace Spectral Methods by Block Lanczos Iteration. Electronic Transactions on Numerical Analysis **31**, 86–109 (2008)
24. Li, F., Shen, C., Fan, J., Shen, C.: Image restoration combining a total variational filter and a fourth-order filter. Journal of Visual Communication and Image Representation **18**(4), 322–330 (2007)
25. Lysaker, M., Lundervold, A., Tai, X.: Noise Removal Using Fourth Order Differential Equations with Applications to Medical Magnetic Resonance Images in Space-Time. IEEE Transaction on Image Processing **12**(12), 1579–1590 (2003)
26. Mathieu, B., Melchior, P., Outstaloup, A., Ceyral, C.: Fractional Differentiation for Edge Detection. Signal Processing **83**, 2421–2432 (2003)
27. Nitzberg, M., Shiota, T.: Nonlinear image smoothing with edge and corner enhancement. Tech. Report 90-2. Harvard University, Cambridge, MA (1990)
28. Nitzberg, M., Shiota, T.: Nonlinear image filtering with edge and corner enhancement. IEEE Trans. Pattern Anal. and Machine Intelligence **14**, 826–833 (1992)
29. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions Pattern Anal. Machine Intelligence **12**, 161–192 (1990)
30. Tumblin, J., Turk, G.: LCIS: A boundary hierarchy for detail-preserving contrast reduction. In: Proceedings of the SIGGRAPH 1999 Annual Conference on Computer Graphics, August 8-13, 1999, Los Angeles, CA, USA, Siggraph Annual Conference Series, pp. 83–90. ACM Siggraph, Addison-Wesley, Longman (1999)
31. Wei, G.: Generalized Perona-Malik equation for image restoration. IEEE Signal Processing Letters **6**(7), 165–167 (1999)
32. Weickert, J.: Anisotropic Diffusion in Image Processing. ECMI Series. Teubner Verlag, Stuttgart (1998)
33. Witkin, A.P.: Scale-space filtering. In: Proc. IJCAI, pp. 1021–1019. Karlsruhe (1983)
34. You, Y., Kaveh, M.: Fourth order partial differential equations for noise removal. IEEE Transaction on Image Processing **9**(10), 1723–1730 (2000)
35. You, Y., Xu, W., Tannenbaum, A., Kaveh, M.: Behavioral analysis of anisotropic diffusion in image processing. IEEE Transaction on Image Processing **5**(11), 1539–1553 (1996)