

Image structure preserving denoising using generalized fractional time integrals

Eduardo Cuesta^a, Mokhtar Kirane^b, Salman A. Malik^b

^aDepartment of Applied Mathematics, E.T.S. of Telecommunication Engineers, University of Valladolid, Spain.

^bLaboratoire de Mathématiques, Image et Applications, Université de La Rochelle, Avenue M. Crépeau, 17042 La Rochelle Cedex, France.

Abstract

A generalization of the linear fractional integral equation $u(t) = u_0 + \partial^{-\alpha}Au(t)$, $1 < \alpha < 2$, which is written as a Volterra matrix-valued equation when applied as a pixel-by-pixel technique, has been proposed for image denoising (restoration, smoothing,...). Since the fractional integral equation interpolates a linear parabolic equation and a hyperbolic equation, the solution enjoys intermediate properties. The Volterra equation we propose is well-posed for all $t > 0$, and allows us to handle the diffusion by means of a *viscosity parameter* instead of introducing non linearities in the equation as in the Perona–Malik and alike approaches. Several experiments showing the improvements achieved by our approach are provided.

Keywords: Image processing, Fractional integrals and derivatives, Volterra equations, Convolution quadrature methods.

2010 MSC: 44A35, 44K05, 45D05, 65R20, 68U10, 94A08.

1. Introduction

Partial differential equations based methods for image processing (filtering, denoising, restorations, segmentation, edge enhancement/detection,...) have been largely studied in the literature (see [47] and references therein).

In that framework the first, and most investigated equation is the linear heat equation with homogeneous Neumann boundary condition

$$\begin{cases} \partial_t u(t, \mathbf{x}) = \Delta u(t, \mathbf{x}), & (t, \mathbf{x}) \in [0, T] \times \Omega, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \frac{\partial u}{\partial \eta}(t, \mathbf{x}) = 0, & (t, \mathbf{x}) \in [0, T] \times \partial\Omega, \end{cases} \quad (1)$$

where ∂_t , and Δ stand for the time derivative, and two-dimensional Laplacian operator, respectively, $\Omega \subset \mathbb{R}^2$ is typically a square domain, $\partial\Omega$ represents the boundary of Ω , $\partial/\partial\eta$ stands for the outward normal derivative, and u_0 is the noisy image from which the objective is to restore the original (ideal) image. Let us notice that $u(t, \mathbf{x})$ stands for the restored image at the time level t .

The interest for this model is due to the fact that the solution of (1) can be written as a convolution

$$u(t, \mathbf{x}) = \int_{\mathbb{R}^2} G_{\sqrt{2t}}(\mathbf{x} - \mathbf{y})u_0(\mathbf{y}) \, d\mathbf{y},$$

where G is the two-dimensional Gaussian kernel

$$G_\sigma(\mathbf{x}) := \frac{1}{2\pi\sigma^2} e^{-|\mathbf{x}|^2/2\sigma^2}.$$

Since convolution with a positive kernel is the basic tool in linear filtering, computing the solution of (1) is equivalent to Gaussian filtering in a classical way.

However, in this equation the diffusion is isotropic which, in the context of image processing, means that smoothing applies uniformly in the whole image and is equivalent in the direction of both coordinates axis, therefore independently of the image structure itself. This yields that in most of cases edges and corners are severely blurred disabling this filter for practical applications.

In view of this, an edge-preserving regularization, i.e. a non uniform diffusion model, will require more sophisticated approaches. One of this was proposed by Perona and Malik in [39] and reads as the non linear heat equation based problem

Email addresses: eduardo@mat.uva.es (Eduardo Cuesta), mokhtar.kirane@univ-lr.fr (Mokhtar Kirane), salman.malik@univ-lr.fr (Salman A. Malik)

$$\begin{cases} \partial_t u(t, \mathbf{x}) &= \operatorname{div} \left(c(|\nabla u(t, \mathbf{x})|^2) \nabla u(t, \mathbf{x}) \right), \\ u(0, \mathbf{x}) &= u_0(\mathbf{x}), \\ \frac{\partial u}{\partial \eta}(t, \mathbf{x}) &= 0, \quad \mathbf{x} \in \partial\Omega, \end{cases} \quad (2)$$

for $(t, \mathbf{x}) \in Q = [0, T] \times \Omega$. The diffusion coefficient $c : [0, +\infty) \rightarrow [0, +\infty)$, is chosen to be close to zero near edges and corners, that is, pixels where gradient is large. On the contrary, c should be large in pixels with low gradient variation. Functions satisfying these assumptions are commonly called *edge stopping functions*, and examples of such a function were firstly proposed by Perona and Malik, e.g.

$$c(s) = 1/(1 + s) \quad \text{or} \quad c(s) = e^{-s}. \quad (3)$$

Unfortunately, the *edge stopping functions* lead in general to backward–forward problems that turn out to be ill–posed. Numerical experiments carried out by some authors with these models (involving edge stopping functions) reported that no significant instabilities are observed; moreover, for large final times, restored images seem to preserve and enhance edges (see e.g. [26]). The reason for not observing any instability in Perona–Malik model is that in many cases the numerical scheme used does not correspond to their equation but rather to a time–regularized one which is a well–posed equation as reported by H. Amman [2]. Even in the case of the discretization corresponding to the equation (i.e. without regularizing) the only one artifact usually observed in the numerical experiments is the *staircasing effect*. This occurs when a standard spatial discretization of the equation is considered instead of the continuous one. In that case it has been proved in [44] that (2) becomes a well–posed system of nonlinear ordinary differential equations.

In the same way, some other approaches have been proposed as for example the ones based on the total variation of suitable functionals (see [43]). All these works have promoted the idea of replacing (2) by nearby equations keeping on the one hand the same image structure, and on the other hand, lying in a *reasonable* functionals space setting where the well–posedness can be guaranteed as well as the bounded variation, and further analytical and numerical properties. The first perturbed model was proposed in [11] where, for a suitable extension of u over \mathbb{R}^2 (e.g. by 0) denoted \tilde{u} , $c(|\nabla u|)$ is replaced by $c(|\nabla(G_\sigma * \tilde{u})|)$ (G_σ as defined above). In that case, for $u_0 \in L_2(\Omega)$, the regularized problem admits a unique solution in $C([0, T], L_2) \cap L_2((0, T), H^1)$ where H^1 is the Sobolev space defined as $H^1(\mathbb{R}^2) := \{f \in L_2(\mathbb{R}^2), \nabla f \in$

$L_2(\mathbb{R}^2)\}$. Despite of some features of this model, images become uniformly grey (for grey–scale images) in the long run, thus the information gets lost (see [2]). Variants of this approach have been studied by many authors (see e.g. [27, 28] and references therein).

Lately, further approaches have been proposed, e.g. by adding regularizing terms, like $\epsilon \Delta \partial_t u$, to the diffusion equation in (2) (see [6]), or for example higher order partial differential equations based regularization (see [21]), and in particular fourth order partial differential equations (see [37, 51]). However, despite of the practical results which seem to be quite good in most of the cases, some of them have not been closely studied from the analytical and the numerical view point.

Many image denoising models based on local average (also known as neighborhood filters) have been proposed for controlling the diffusion process. Let us mention some of the neighboring filters: Yaroslavsky neighborhood filter [50], sigma filter [30], bilateral filtering [45] are few examples. The faster diffusion is obtained in these filters by assigning appropriate weights where the neighborhood pixels have gray scale value close to one another (uniform region) and slower diffusion is obtained across the boundaries of the region, resulting in the preservation of edges in the image. All the local nonlinear filters (neighborhood filters) create artifact boundaries (staircase effect) in the restored images. In [8], the authors justify the phenomena of artifact boundaries by showing that the Yaroslavsky neighborhood filter has exactly the same qualitative behavior as Perona–Malik model [39]. The similar behavior of staircase for the bilateral filter was observed (see [4]).

Finally let us comment that one can face up to diffusion filtering in a more general framework as the one of *anisotropic* models. In fact most of models we mentioned above turn out to be *isotropic* since the diffusion is governed by means of a scalar–valued function c which allows to reflect the structure of the underlying image, but the diffusion turns out to be the same in the two orthogonal directions of coordinate axis. A generalization of such approaches consist in replacing c by a tensor which allows on the one hand to rotate the flux and so the diffusion orthogonal directions toward a suitable orientation, and on the other hand a suitable choice of the eigenvalues of the tensor can lead to different diffusions on each single direction (see e.g. [1, 5, 46, 48, 49] for more details). In the present work we focus on the scalar approach, i.e. no tensor diffusion are considered, keeping in mind that a tensorial approach in the framework of fractional calculus fits and will be the topic of our future works.

In this work, we present a new approach based on

fractional calculus which allows us to handle the diffusion, i.e. the smoothing in the image terminology, by means of a single parameter which plays the role of "viscosity" parameter in a linear partial differential equation. Let us mention the new features of our proposed model although satisfactory practical results are obtained (see section 6). First, the linear model we are proposing here is well-posed; since the final objective is the practical implementation, very efficient numerical discretizations can be applied for the proposed model because a large variety of them have been closely studied by many authors and they are at our disposal for the experiments in Section 6. Moreover, the model we propose enjoys the well-posedness for all positive time levels, i.e. for $t > 0$, on the contrary to what typically happens when nonlinear equations are involved. In fact, despite of the Perona-Malik model turn out to be a well-posed systems of nonlinear equations after discretizing in space as we described earlier, the existence of a unique solution can be locally guaranteed i.e. for a finite time interval (normally small). Therefore, when using the semidiscrete Perona-Malik model in practical applications some estimation of the final time should be done. Second, we have to highlight the simplicity of our model for controlling the diffusion process, which has been achieved by choosing a scalar function which allows a suitable choice of the viscosity parameter as we describe in Section 5. Moreover, due to the numerical scheme chosen for our experiments, some additional properties are guaranteed, in fact the positivity of the image seen in the continuous setting is also guaranteed in the discrete one (Th. 1, in Section 4).

This paper is organized as follows. In section 2, we recall some facts concerning fractional calculus, and the first approaches to image processing by using fractional calculus. Section 3 focusses on generalized fractional integrals based approach to image processing which is the main novelty of this work. Numerical discretizations are presented in Section 4. The discussion on the implementation, and practical experiments, are shown in Sections 5 and 6 respectively. Finally, we end with conclusions in Section 7.

2. Fractional calculus

Image filtering by means of fractional calculus is first considered in [13]. In that work, a generalization of the

heat equation (1) is proposed which reads as

$$\begin{cases} \partial_t^\alpha u(t, \mathbf{x}) = \Delta u(t, \mathbf{x}), & (t, \mathbf{x}) \in [0, T] \times \Omega \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \partial_t u(0, \mathbf{x}) = 0, & \mathbf{x} \in \Omega, \\ \frac{\partial u}{\partial \eta}(t, \mathbf{x}) = 0, & (t, \mathbf{x}) \in [0, T] \times \partial\Omega, \end{cases} \quad (4)$$

where ∂_t^α stands for the fractional time derivative of order $1 < \alpha < 2$ in the sense of Riemann–Liouville. Integrating in both sides, the problem (4) can be expressed as

$$u(t, \mathbf{x}) = u_0(\mathbf{x}) + \frac{1}{\Gamma(\alpha)} \int_0^t (t-s)^{\alpha-1} \Delta u(s, \mathbf{x}) ds,$$

also with homogeneous Neumann boundary condition, or in a compact format

$$u(t, \mathbf{x}) = u_0(\mathbf{x}) + \partial^{-\alpha} \Delta u(t, \mathbf{x}), \quad (5)$$

where $\partial^{-\beta}$, for $\beta > 0$, stands for the fractional integral of order $\beta \in \mathbb{R}^+$, in the sense of Riemann–Liouville.

Let us recall that, for $g : [0, +\infty) \rightarrow \mathbb{R}$, $g \in AC[0, +\infty)$ (the space of absolutely continuous functions) the integral of order $\beta \in \mathbb{R}^+$ in the sense of Riemann–Liouville is defined as the convolution integral

$$\partial^{-\beta} g(t) := \int_0^t k_\beta(t-s)g(s) ds, \quad t \geq 0, \quad (6)$$

where $k_\beta(t) := t^{\beta-1}/\Gamma(\beta)$, for $t > 0$ (see [29]). Now, the definition of the fractional derivative of order $\beta \geq 0$ is

$$\partial^\beta g(t) := \frac{d^m}{dt^m} \partial^{\beta-m} g(t), \quad t \geq 0,$$

where $m \in \mathbb{N}$, $m-1 < \beta \leq m$.

The interest of our model in the framework of image processing is due to the fact that, for $1 < \alpha < 2$, the problem (4) interpolates the linear (parabolic) heat equation (1) corresponding to $\alpha = 1$ (with no need of $\partial_t u(0, \mathbf{x}) = 0$), and the linear (hyperbolic) wave equation

$$\begin{cases} \partial_t^2 u(t, \mathbf{x}) = \Delta u(t, \mathbf{x}), & (t, \mathbf{x}) \in [0, T] \times \Omega \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \partial_t u(0, \mathbf{x}) = 0, & \mathbf{x} \in \Omega, \\ \frac{\partial u}{\partial \eta}(t, \mathbf{x}) = 0, & (t, \mathbf{x}) \in [0, T] \times \partial\Omega, \end{cases} \quad (7)$$

corresponding to $\alpha = 2$ (with zero initial velocity). Therefore, some properties of the solution of (4) are intermediate between the ones of (1) and (7) (see e.g.

[23, 24]). In particular, it can be easily proved that for the scalar equation (i.e., by replacing Δ in (4) with a complex λ with non-positive real part) the solution decays as $1/(1 + t^\alpha |\operatorname{Re}(\lambda)|)$, i.e., the solution is $o(t^{-\alpha})$, as $t \rightarrow +\infty$. In that case, the solution decays slower than the one of the scalar heat equation (with the same λ) for which the solution decays exponentially, and faster than the one of the scalar wave equation whose solution does not decay (but oscillates). The diffusion is now handled by the parameter α . We show below that the solution of the model we propose is well behaved.

At the same time, fractional calculus was proposed for edge detection in [38], and later for image denoising in [3]. In these papers, the authors proposed anisotropic equations, where the anisotropy is handled by means of spatial fractional derivatives; however, the papers do neither include the study of the well-posedness of the problem nor the study of the behavior of the fully numerical discretization.

Finally, let us mention [16, 17, 18, 19, 20] where the fractional calculus, also applied to image processing, is understood as the fractional powers of the two-dimensional Laplacian, i.e. $(-\Delta)^\beta$, for $\beta > 0$, therefore in a different framework and with some different features.

3. Volterra equations

Despite of the fact that the approach (5) in Section 2 seems to be very promising, the diffusion (smoothing) is still uniform all over the whole image as in (1).

The above idea led us to a refinement which consists in splitting the whole image into sub-images, and apply (4) with different values of α for each sub-image (see [12]). However the approach we propose in this paper is more than a slight extension of the one in [12], from a technical viewpoint. Roughly speaking, in [12] the choice of each α is carried out by setting values close to 1 for the sub-images with lower mean gradient variation, and close to 2 for the sub-images with higher mean gradient variation. This approach does not provide satisfactory results as extended in the framework of image denoising despite of the fact that this approach has provided good results in some practical situations such as in satellite image classification (see [42]). It is mainly due to the fact that in most of cases the borders of each single sub-image keep clearly sharpened which is undesirable in image denoising.

However this idea suggested us a finer approach which is the main contribution of this work, and which intends to be the limit of the above situation, i.e., the

application of the fractional equation (4) with a different value of α (i.e. order of derivative) for each single pixel. The values of α are chosen according to the gradient variation at each single pixel as we discuss in Section 5.

Hereafter, we will consider gray-scale images since for colored images processing becomes a more difficult task. A naive way for colored images is to perform a similar methodology based on the one we present here but separately for each of the three layers (one per color) e.g. in the case of RGB format.

Let us start by taking a spatial discretization of the Laplacian in (4) based on a second order central difference scheme with mesh length $h > 0$. In such a way, Δ transforms into a $M^2 \times M^2$ pentadiagonal matrix Δ_h (see Fig. 1), and in the same fashion, $u(t, \mathbf{x})$ is transformed into a $M^2 \times 1$ vector-valued function $\mathbf{u}(t)$ which stands for the vector-arranged image pixels at time level t where M is the length and width of the image.

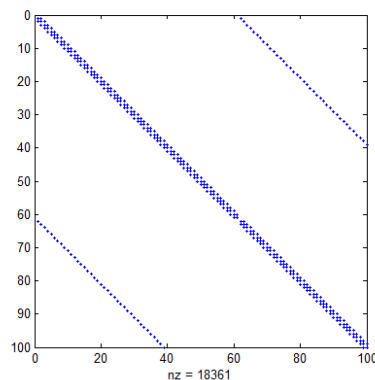


Figure 1: Sparsity pattern of the discretized Laplacian

Actually, since at present the most of signals (image, sound,...) are handled in digital format, this approach becomes natural.

As we commented above, the novelty of our approach consists in replacing the (only one) order derivative α of equation (4) with a different value of α for each single pixel of the image. This approach reads now as the linear Volterra matrix-valued equation

$$\mathbf{u}(t) = \mathbf{u}_0 + \int_0^t \mathbf{K}(t-s)\mathbf{u}(s) ds, \quad 0 \leq t \leq T, \quad (8)$$

where \mathbf{u}_0 is the vector-arranged initial data (noisy image), and the convolution kernel \mathbf{K} is defined as

$$\mathbf{K}(t) = I(t) \cdot \Delta_h$$

with

$$I(t) = \begin{bmatrix} \frac{t^{\alpha_1}}{\Gamma(\alpha_1 + 1)} & 0 & \dots & 0 \\ 0 & \frac{t^{\alpha_2}}{\Gamma(\alpha_2 + 1)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \frac{t^{\alpha_{M^2}}}{\Gamma(\alpha_{M^2} + 1)} \end{bmatrix}$$

and, $1 < \alpha_j < 2$, for $j = 1, 2, \dots, M^2$.

Let us notice that \mathbf{K} is a pentadiagonal matrix valued function, and since the Laplace transform of \mathbf{K} exists, the well-posedness of (8) is then guaranteed all over the positive real line (see e.g. [41]).

4. Time discretizations

4.1. Background

Time discretizations of Volterra equations as (8) have been largely studied in literature; let us mention, e.g., the convolution quadrature based methods (see [33, 34, 35]). In particular Runge–Kutta convolution quadrature methods (the convolution quadrature is based on classical Runge–Kutta methods) of that equations provide high order numerical methods jointly with good stability properties. In [36], these discretizations have been studied in the abstract setting of sectorial operators, i.e. for convolution kernels whose Laplace transform is of *sectorial type*. Let us recall that a complex function G is of *sectorial type* if there exist $0 < \theta < \pi/2$, $c \in \mathbb{R}$, and $\mu, M > 0$ such that G is analytic in the sector

$$S_\theta := \{\lambda \in \mathbb{C} : |\arg(\lambda - c)| < \pi - \theta\},$$

and

$$|G(\lambda)| \leq \frac{M}{|\lambda|^\mu}, \quad \lambda \in S_\theta.$$

Under these assumptions, the inverse Laplace transform of G can be written by means of the Bromwich formula as

$$g(t) = \frac{1}{2\pi i} \int_\gamma e^{t\lambda} G(\lambda) d\lambda,$$

where γ is a complex path connecting $-i\infty$ and $+i\infty$ parallel to the boundary of S_θ with increasing imaginary part.

The convergence of these methods has been recently extended to analytic semigroups (see [9]) where the only one requirement on the kernel is the existence of the Laplace transform (weaker assumption than the assumption of sectorial type). Since the Laplace transform of each function $t^{\alpha_j}/\Gamma(\alpha_j + 1)$ in (8) exists, and therefore

the Laplace transform of \mathbf{K} exists as a matrix-valued function, these methods turn out to be appropriate for our purposes. In this case, if $\tilde{\mathbf{K}}$ denotes the Laplace transform of \mathbf{K} , then the inverse Laplace transform can be written as

$$\mathbf{K}(t) = \frac{1}{2\pi i} \int_\gamma e^{t\lambda} \tilde{\mathbf{K}}(\lambda) d\lambda,$$

where $\gamma(r) = a + ri$, $-\infty < r < +\infty$, for $a \in \mathbb{R}^+$. Let us notice that now the whole path γ lies on the right-hand complex plane $\text{Re}(\lambda) \geq a$.

In order to set a suitable Runge–Kutta convolution quadrature method, we must take into account that the time regularity of the solution of (8) is constrained by the nature of the convolution kernel, in particular the continuity of their derivatives is guaranteed only up to the first order. Therefore, in this work, we will focus on the backward Euler convolution quadrature method, i.e., avoiding higher order schemes whose convergence will require more regularity on the solutions. This method will be sufficient to show the new features of our approach.

4.2. Convolution quadratures

For the sake of the readers convenience, we first recall the definition of the backward Euler convolution quadratures, and for the sake of the simplicity of the explanation we refer the readers to [9, 35] for the definition of Runge–Kutta convolution quadratures in the general case and further references.

Let $\tau > 0$ be the time step of the discretization. The convolution integral in (8) reads

$$\begin{aligned} \int_0^t \mathbf{K}(t-s)\mathbf{u}(s) ds &= \int_0^t \frac{1}{2\pi i} \int_\gamma e^{\lambda(t-s)} \tilde{\mathbf{K}}(\lambda) d\lambda \mathbf{u}(s) ds \\ &= \frac{1}{2\pi i} \int_\gamma \tilde{\mathbf{K}}(\lambda) Y(\lambda, t) d\lambda, \end{aligned}$$

where $Y(\lambda, t)$ stands for the solution of the ordinary differential equation

$$\mathbf{y}'(t) = \lambda \mathbf{y}(t) + \mathbf{u}(t), \quad 0 \leq t \leq T, \quad \text{with } \mathbf{y}(0) = 0. \quad (9)$$

The backward Euler convolution quadrature is obtained as

$$\int_0^{t_n} \mathbf{K}(t_n - s)\mathbf{u}(s) ds \approx \frac{1}{2\pi i} \int_\gamma \tilde{\mathbf{K}}(\lambda) Y_n(\lambda) d\lambda,$$

where $t_n = n\tau$, and $Y_n(\lambda)$ stands for the approximation of $Y(\lambda, t_n)$ reached by the backward Euler method applied to (9). Therefore, the convolution quadrature reads

$$\int_0^{t_n} \mathbf{K}(t_n - s)\mathbf{u}(s) ds \approx \sum_{j=0}^n \mathbf{Q}_{n-j}^{(\alpha)} \mathbf{u}(t_j),$$

where the weights $\mathbf{Q}_j^{(\alpha)}$'s are defined in terms of the backward Euler characteristic polynomials quotient $\rho(z)/\delta(z) = z/(z-1)$ evaluated in the variable $\xi = 1/z$. In fact, such weights turn out to be the coefficients of

$$\tilde{\mathbf{K}}\left(\frac{1-\xi}{\tau}\right) = \sum_{j=0}^{+\infty} \mathbf{Q}_j^{(\alpha)} \xi^j.$$

which explicitly written read

$$\mathbf{Q}_j^{(\alpha)} = \tau^\alpha \begin{bmatrix} \binom{\alpha_1}{j} & 0 & \dots & 0 \\ 0 & \binom{\alpha_2}{j} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \binom{\alpha_{M^2}}{j} \end{bmatrix} \cdot \Delta_h, \quad (10)$$

for $j = 0, 1, 2, \dots$, as $M^2 \times M^2$ matrices. (see [9, 14, 35] for more details).

4.3. Numerical method. Convergence

Let \mathbf{u}_n be the approximation of $\mathbf{u}(t_n)$, for $n \geq 0$. Then the time discretization of (8) by means of the backward Euler convolution quadrature method reads

$$\mathbf{u}_n = \mathbf{u}_0 + \sum_{j=1}^n \mathbf{Q}_{n-j}^{(\alpha)} \mathbf{u}_j, \quad n \geq 1,$$

and keeping in mind the practical implementation, since the matrix Δ_h is not singular, the unique n -th approximation is reached by solving the linear system

$$(I - \mathbf{Q}_0^{(\alpha)}) \mathbf{u}_n = \mathbf{u}_0 + \sum_{j=1}^{n-1} \mathbf{Q}_{n-j}^{(\alpha)} \mathbf{u}_j, \quad n \geq 1. \quad (11)$$

In the abstract setting considered in [9], optimal error bounds are obtained. In particular, for the backward Euler based method, the first order is reached by assuming the existence and boundedness of the second derivative of the solution. However, since the second derivative of the solution of (8) is merely integrable but not continuous, these results cannot be directly applied. In our case, if one takes into account the nature of the convolution kernel, then the stability proven in [14] jointly with Theorem 3.1 in [33] allow us to guarantee that the method is convergent of first order.

Besides, in [9] an interesting result is also proven which becomes even more interesting when applications fit into the framework of image processing. For the readers convenience we recall this result in the case of the backward Euler convolution quadrature method we apply in this paper.

Theorem 4.1. *If \mathbf{u} is the solution of (8), and \mathbf{u}_n , for $n \geq 0$, is the numerical solution yielded by (11), then there exists a probability density $\rho_{n,\tau} : [0, +\infty) \rightarrow \mathbb{R}$ such that*

$$\mathbf{u}_n = \int_0^{+\infty} \mathbf{u}(s) \rho_{n,\tau}(s) ds, \quad n \geq 1. \quad (12)$$

The interest of this theorem is that, since $\rho_{n,\tau}$ is a probability density, i.e., positive and satisfying

$$\int_0^{+\infty} \rho_{n,\tau}(s) ds = 1,$$

if \mathbf{u} is for example positive, then the representation (12) guarantees the positivity of \mathbf{u}_n . In other words, the numerical solution (11) preserves, among other properties, the positivity of the continuous solution.

Moreover, let us point out that in [9] it is proven that the density of probability in (12) does not depend on the equation itself but only on the numerical method. In particular, for the method considered here (see [14, 10]), there is an explicit expression for such a density

$$\rho_{n,\tau}(t) = \frac{1}{\tau(n-1)!} \left(\frac{t}{\tau}\right)^{n-1} e^{-t/\tau}, \quad n \geq 1.$$

Finally, we can mention some other very efficient methods to discretize (8) as for example the ones based on the discretization of the inverse Laplace transform (see [32]), the ones based on the adaptive fast and oblivious convolutions (see [31]), or the collocation methods (see [7] and references therein).

5. Implementation

In this section we will discuss some facts concerning the implementation of (11) itself.

First of all, the choice of α 's should be done according to the idea of preserving edges and corners and removing noise. Therefore, in view of the discussion in Section 2, pixels where the gradient is large should be associated with values of α close to 2. On the contrary, pixels with lower gradients should be associated with values of α close to 1. Let us notice that the practical computation of the gradient variation turns out to be very simple in a discrete setting as the one we consider in the spatial variables.

However some remarks have to be taken into account:

- On the one hand, avoiding singular situations can be yielded in both-sides values of α , i.e. $\alpha = 1$ and $\alpha = 2$, at least from the numerical point of view, we will not reach these values when setting α 's. In particular, in Section 6, for $j = 1, 2, \dots, M^2$, $\alpha_j \in [1 + \epsilon, 2 - \epsilon]$ with $\epsilon = 10^{-3}$.

- On the other hand, since extreme situations appear such as *isolated* noisy pixels (see Figure 3 where a gray-scale image is shown as a three-dimensional surface), a particular choice of α 's is expected; in particular for those pixels, values close to 1 will be associated. On the contrary, near edges and corners (very high gradient variation) no smoothing should be required, therefore α 's close to 2 will be set for that pixels.

According to this criteria, in this work, the setting of α values follows a profile distribution as in Figure 2.

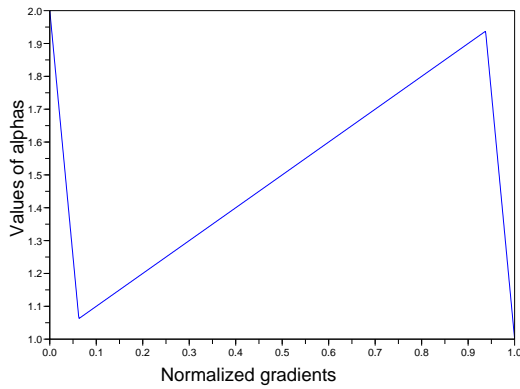


Figure 2: Profile distribution of α 's.

Notice that this simple and naive procedure allows one to establish different settings of distributions depending, e.g., on the characteristics of each image.

- From a computational point of view, the number of different values of α 's should be limited, otherwise if one admits a number of α 's as large as the number of pixels, the implementation becomes unavailable in practical cases. In fact, in this work, we set $\alpha \in \{1 + j/N, 1 \leq j \leq N\}$, for a fixed integer N which in Section 6 turns out to be $N = 100$.

Let us also mention that in (10) a fixed number of weights $\mathbf{Q}_j^{(\alpha)}$ are computed once for all for each α_j . Moreover, the practical computation of that weights has been carried out by means of the Fast Fourier Transform as in [9], therefore saving a noticeable *run-time*.

Another fact of interest concerns the measure of goodness of an implementation or procedures in image filtering, restoration, and in general, to measure the

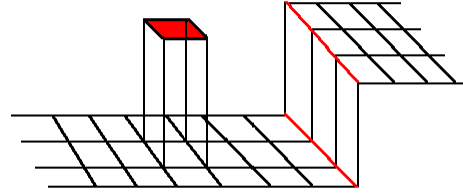


Figure 3: Three dimensional representation of (expected) isolated noisy pixels for a gray-scale image.

quality of a processed image. Despite of in many cases a visual analysis turn out to be sufficient to determine the goodness of a methodology when applied to image processing, in this work, we consider two numerical criteria, SNR and $PSNR$, which have been largely used in literature (see e.g [15, 22, 40]), and which are commonly applied to determine the quality of a processed image in the sense commented above (filtering, restoration,...). In fact, SNR and $PSNR$ stand for the Signal to Noise Ratio and Peak Signal to Noise Ratio, respectively, and the unit for both of these ratios are decibels (dB). To be more precise, SNR of a restored image R compared to an ideal image I is defined as

$$SNR = 10 \cdot \log_{10} \left(\frac{\text{var}(I)}{\text{var}(I - R)} \right),$$

where $\text{var}(x)$ stands for the variance of the vector x . Here I and R are considered as vector arranged gray-scale images, as in Section 3, with 256 gray levels. In the same way, $PSNR$ is defined as

$$PSNR = 10 \cdot \log_{10} \left(\frac{\sum_{i,j} 255^2}{\sum_{i,j} (I_{i,j} - R_{i,j})^2} \right),$$

where $I_{i,j}$, $R_{i,j}$ are the pixel values of I and R respectively. Notice that in restoration problems, we have a corrupted image, and try to restore the ideal image which in general is not available, but for the calculation of SNR and $PSNR$ (from above formulas) ideal image is required. For the experiments in Section 6, we take an image (ideal image) and we perturb that image by adding up some noise. This image is then used for restoration; therefore SNR and $PSNR$ can be computed for the restored images.

6. Practical results

In this section, we show the improvements in the restorations provided by our approach. To this end, we perform some experiments where a noisy image is evolved by using the Perona–Malik model (2) with $c(s) = e^{-s}$ as commented in (3) (PM), the contrast parameter (λ) is chosen such that $1.5 * \sigma < \lambda < 2 * \sigma$ as suggested in [25] (where σ is the noise variation) and the model (8) we propose (VE).

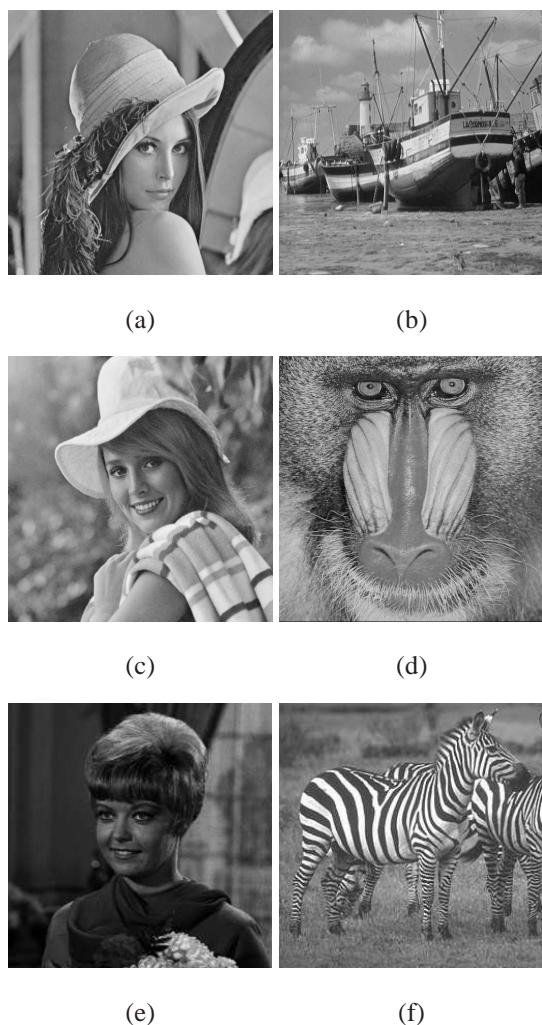


Figure 4: Images for the first experiment: (a) Lena, (b) Boats, (c) Elaine, (d) Baboon, (e) Lady, (f) Zebra.

First experiment:

In the first experiment the six images shown in (Figure 4) are considered for the experimental validation of the proposed method. Each original image from the six

images (Figure 4) has been perturbed by additive Gaussian noise of variance ranging from 10 to 30. In Table 1, we show the results (in terms of SNR) yielded by the restoration carried out with the mentioned procedures.

In fact, in view of Table 1, it can be observed that similar results are reached with (PM) and (VE). However we should recall that (VE) stands for a linear model whose well-posedness is guaranteed on the contrary to what happens with (PM).

For further close observation of the procedures applied to the images of Lena (Figure 4.(a)) and Elaine (Figure 4.(c)), with $\sigma = 25$ a part of such image is considered for all denoising models; in particular, a 200×200 part of the image has been considered (see Figure 5). The restored images by Perona–Malik model for both Lena and Elaine images, preserves edges/corners (Figure 5.(c) and Figure 5.(g)) but smooths very strongly the *flat* areas which causes lost of information regarding the texture of the image. Also, strong artifacts at the edges and corners has been observed as reported in [8]. The (VE) denoises the image (Figure 5.(j) and Figure 5.(h)), also by preserving edges/corners of the image; but here smoothing in flat areas is not so strong as with (PM) and it preserves the structure of the image without creating any artifacts.

In Table 2 *PSNR* analysis for the six images of Figure 4 with the variance of noise from 10 to 30 is given. The values of the table shows that the proposed method performs more or less in the same manner. In fact for the images from Figure 4.(a) to Figure 4.(e) the values of *PSNR* for the purposed method are slightly larger then the (PM) model. For the image of zebra (PM) model has values of *PSNR* larger then (VE) model.

Consider the image of Lady (Figure 4.(e)) perturbed by an additive Gaussian noise of variance $\sigma = 25$ for further investigation. Notice that the size of the image is 256×256 and in Figure 6, we provide original image (Figure 6.(a)), noisy image (Figure 6.(b)), restored images by (PM) and (VE) in (Figure 6.(c)-Figure 6.(d)), respectively. Figure 6.(e) and Figure 6.(f) are the residual to (PM) model and (VE) model with the noisy image, i.e, Residual = Noisy image – restored image. The residual figure 6.(e) contain more structure part of the image as compare to the figure 6.(f), which also hints that the denoising by proposed (VE) method is structure preserving.

Second experiment:

As in the previous discussion of *SNR* and *PSNR* for (PM) and (VE), keep very close one to each other, the

Table 1: First experiment: SNR analysis

σ	10	15	20	25	30	10	15	20	25	30
Input SNR	13	10	7	5.6	4	13.3	9.88	7.4	5.5	4
Method	Lena (512 × 512)					Boats (512 × 512)				
(PM)	16.81	14.79	13.82	13.32	11.27	15.07	13.69	12.26	11.32	9.72
(VE)	17.31	15.00	14.46	13.73	12.78	15.74	14.10	12.65	11.57	10.32
Input SNR	13.2	9.7	7.3	5.3	3.8	12.5	9	6.5	4.5	3
Method	Elaine (512 × 512)					Baboon (512 × 512)				
(PM)	15.60	14.56	13.52	13.01	11.22	13.42	10.09	8.34	7.27	6.15
(VE)	16.32	14.74	13.71	13.52	11.96	12.88	10.29	8.51	7.49	6.89
Input SNR	12	8.73	6.34	4.5	3.1	13.79	10.25	7.81	5.88	4
Method	Lady (256 × 256)					Zebra (256 × 256)				
(PM)	16.06	14.23	12.50	11.90	9.94	15.18	12.38	10.19	9.13	7.49
(VE)	16.49	14.54	13.19	12.00	11.07	14.33	11.44	9.63	8.14	7.60

efficiency of our approach seems to be based on no more than an optical evidence. However, we will show that the efficiency of our approach is more than optical, and to this end, we will consider images where the texture plays a crucial role, and where the restoration procedures can be stressed.

The images we consider in these experiments are a 250×250 size image of a wood (Figure 7. (a)), and a 512×512 size naive image (Figure 7. (e)), both strongly perturbed with gaussian noise (ratios $SNR = 0.014$ and $PSNR = 12$, see Figure 7. (b), $SNR = 0.054$ and $PSNR = 16.2$, see Figure 7. (f), respectively). In these kind of images, the texture turns out to be more important than edges preservation, and among the numerical results in Table 3, a simple overview shows the goodness of our approach vs. (PM).

However, to be more precise in our analysis, Table 3 shows a numerical evidence of the efficiency of our method; in fact it must be highlighted that SNR and $PSNR$ are improved by using (VE) in comparison with (PM).

Table 3: Second experiment (textured images).

	SNR	PSNR	Figure
(PM)	1.8	12.1	Fig. 7.(c)
(VE)	4	14.4	Fig. 7.(d)
(PM)	5	21.2	Fig. 7.(g)
(VE)	6.8	22.6	Fig. 7.(h)

7. Conclusions and outlook

In the present work, we propose a partial differential equation based approach to image processing (filtering, denoising, enhancing,...) whose main novelty is that it fits into the framework of fractional calculus (derivatives and integrals) hence Volterra equations.

The interest of our work is twofold: On the one hand, the model we propose allows us to handle the smoothing by means of certain "viscosity" parameters which define the matrix-valued linear Volterra equation we propose. In other words, the smoothing is now handled by means of a linear partial equation, i.e. without introducing tricky nonlinear terms in the equation as many authors propose.

On the other hand, the model we propose fits into a closed mathematical setting, both from the analytical and the numerical point of view. To be more precise, the well-posedness of the Volterra equation we propose is as we said guaranteed all over the whole positive real line on the contrary what typically happens with nonlinear models, and numerical methods for its discretization have been largely studied in the literature.

As an additional interesting property of our proposal is that one can change the *profile* of the filter merely by changing the "viscosity" parameters setting, i.e. the distribution function we use. This allows the user to adapt easily the filter to each single image according to its characteristics.

Table 2: First experiment: PSNR analysis

σ	10	15	20	25	30	10	15	20	25	30
Input PSNR	26.66	24.5	22	20.2	18	28	24.6	22	20.27	18.7
Method	Lena (512 × 512)					Boats (512 × 512)				
(PM)	28.14	27.40	26.73	26.00	25.15	29.45	28.00	27.12	25.64	24.15
(VE)	30.00	29.33	28.37	27.08	25.94	30.49	28.42	27.25	26.32	25.02
Input PSNR	28	24.6	22.16	20.25	18.7	28	24.6	22	20.19	18.6
Method	Elaine (512 × 512)					Baboon (512 × 512)				
(PM)	29.11	28.50	26.73	25.94	23.61	28.44	24.91	22.62	22.57	21.75
(VE)	30.98	30.19	28.41	28.00	26.08	29.02	25.92	24.07	22.87	22.34
Input PSNR	28.24	24.88	22.49	20.69	19.23	28.24	24.7	22.27	20.3	18.78
Method	Lady (256 × 256)					Zebra (256 × 256)				
(PM)	32.06	30.29	28.46	28.00	27.04	29.64	26.83	24.63	22.05	19.95
(VE)	32.14	30.78	29.35	27.85	25.76	28.00	25.45	22.57	21.45	19.00

Acknowledgements

The research of the first author has been supported by DGI-MCI under project MTM2010-19510 (subprogram MTM) cofinanced by FEDER funds.

References

- [1] Aja-Fernández, S., García, R.L., Li, D.T.X.. Tensors in Image Processing and Computer Vision. Springer-Verlag, London, 2009.
- [2] Amann, H.. Time-delayed Perona-Malik type problems. Proceedings of the International Conference on Differential Equations (Bratislava, Check Republic 25-29 July 2005), K Mikula, et al (eds), Comenius University Press 2005::15–38.
- [3] Bai, J., Feng, X.C.. Fractional anisotropic diffusion for image denoising. IEEE Trans Image Process 2007;16:2492–2502.
- [4] Barash, D.. A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. IEEE Trans Pattern Anal Machine Intell 2002;24(6):844–856.
- [5] Bartels, S., Prohl, A.. Stable discretizations of scalar and constrained vectorial Perona-Malik equation. Interfaces Free Bound 2007;9(4):431–453.
- [6] Bellettini, G., Fusco, G.. A regularized Perona-Malik functional: some aspects of the gradient dynamics. Proceedings of the International Conference of Differential Equations (Hasselt, Belgium 22-26 July 2003), F Dumortier, H Broer, et al (eds), World Scientific 2003::639–644.
- [7] Brunner, H.. Collocation Methods for Volterra Integral and Related Functional Equations. Cambridge Monographs Applied and Computational Mathematics. Cambridge University Press, 2004.
- [8] Buades, A., Coll, B., Morel, J.M.. The staircasing effect in neighborhood filters and its solution. IEEE Trans on Image Process 2006;15(6):1499–1505.
- [9] Calvo, M., Cuesta, E., Palencia, C.. Runge-Kutta convolution quadrature methods for well-posed equations with memory. Numer Math 2007;107:589–614.
- [10] Calvo, M.P., Cuesta, E., Palencia, C.. Backward Euler method as a positivity preserving method for abstract integral equations of convolution type. In: Oustaloup, A.e., editor. Proceedings of Fractional Differentiation and its Applications '06. 2006. .
- [11] Catte, F., Lions, P.L., Morel, J.M., Coll, T.. Image selective smoothing and edge detection by nonlinear diffusion. SIAM J Numer Anal 1992;29(1):182–193.
- [12] Cuesta, E.. Some advances on image processing by means of fractional calculus. In: Nonlinear Science and complexity. World Scientific; volume II; 2008. p. (in press).
- [13] Cuesta, E., Finat, J.. Image processing by means of a linear integro-differential equation. IASTED 2003;438–442.
- [14] Cuesta, E., Palencia, C.. A numerical method for an integro-differential equation with memory in Banach spaces. SIAM J Numer Anal 2003;41:1232–1241.
- [15] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.. Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space. In: IEEE Int. Conf. Image Process., IICIP 2007. 2007. .
- [16] Didas, S.. Denoising and enhancement of digital images, variational methods, integrodifferential equations, and wavelets. Ph.D. thesis; Saarland University; 2008.
- [17] Didas, S., Burgeth, B., Imiya, A., Weickert, J.. Regularity and scale-space properties of fractional high order linear filtering. Scale Space and PDE Methods in Computer Vision Lecture Notes in Computer Vision 2005;3459:13–25.
- [18] Didas, S., Steidl, G., Weickert, J.. Discrete multiscale wavelet shrinkage and integrodifferential equations. In: Scheelkens, P., Ebrahimi, T., Cristóbal, G., Truchetet, F., editors. Optical and Digital Image Processing–Photonic Europe. Proceedings of SPIE. Bellingham; volume 7000; 2008. p. 7000S–1–7000S–12.
- [19] Didas, S., Steidl, G., Weickert, J.. Integrodifferential equations for continuous multiscale wavelet shrinkage: The discrete case. Technical report 214; Department of Mathematics, Saarland University; 2008.
- [20] Didas, S., Weickert, J.. Integrodifferential equations for continuous multiscale wavelet shrinkage. Inverse Problems and Imaging 2007;1(1):47–62.
- [21] Didas, S., Weickert, J., Burgeth, B.. Properties of higher order

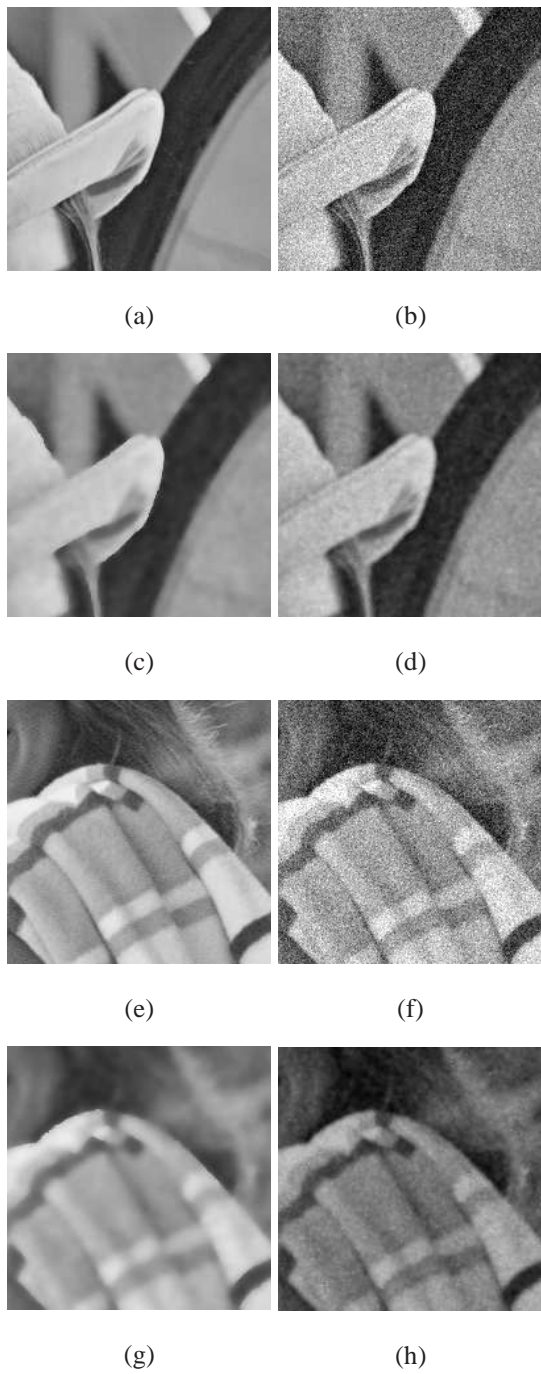


Figure 5: Zoomed 200×200 part of the Lena and Elaine images: (a) Original image of Lena, (b) Noisy image perturbed by Gaussian noise ($\sigma = 25$), (c) Restoration by (PM), (d) Restoration by (VE), (e) Original image of Elaine, (f) Noisy image perturbed by Gaussian noise ($\sigma = 25$), (g) Restoration by (PM), (h) Restoration by (VE).

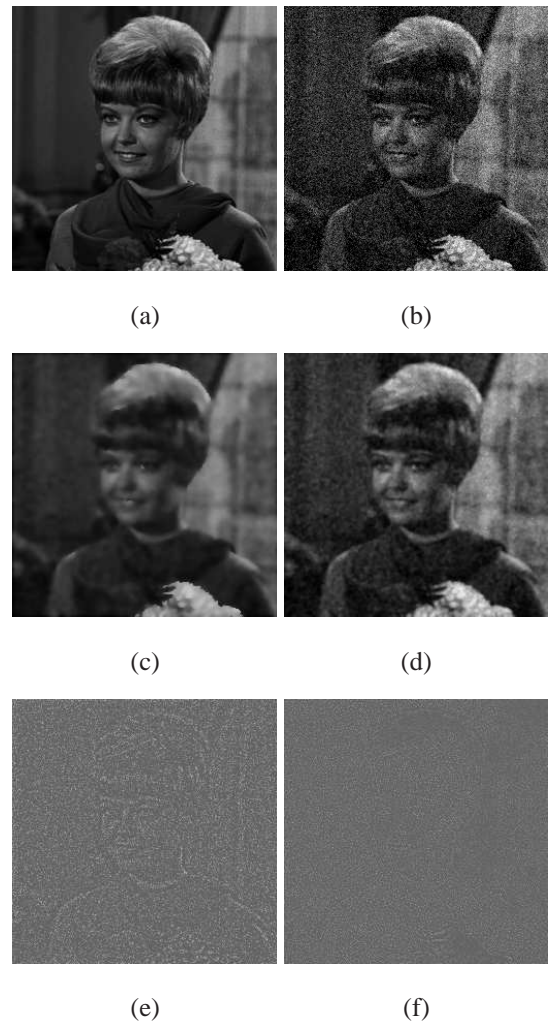
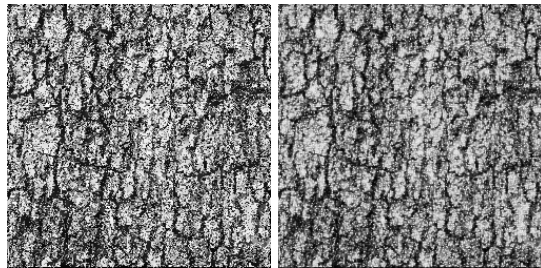


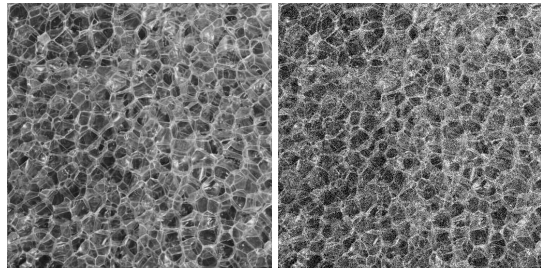
Figure 6: Analysis of Lady image: (a) Original image of Lady, (b) Noisy image perturbed by Gaussian noise ($\sigma = 25$), (c) Restoration by (PM), (d) Restoration by (VE), (e) Residual to (PM) restoration, (f) Residual to (VE) restoration.



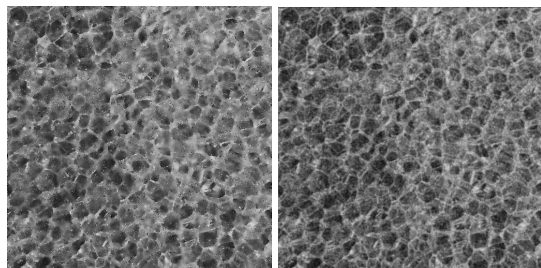
(a) (b)



(c) (d)



(e) (f)



(g) (h)

Figure 7: Denoising of textured images: (a) Original image of wood, (b) Noisy image perturbed by gaussian noise, (c) (PM), (d) (VE) (e) Original naive image, (f) Noisy image perturbed by gaussian noise, (g) (PM), (h) (VE).

nonlinear diffusion filtering. *J Math Imaging Vis* 2009;35:208–226.

[22] Emmanuel, J.L.S., C, E.J., Donoho, D.L., Wavelet, A., Denoising, I. The curvelet transform for image denoising. *IEEE Transactions on Image Processing* 2000;11:670–684.

[23] Fujita, Y. Integro-differential equation which interpolates the heat equation and the wave equation. *Osaka J Math* 1990;27:319–327.

[24] Fujita, Y. Integro-differential equation which interpolates the heat equation and the wave equation (ii). *Osaka J Math* 1990;27:797–804.

[25] Gerig, G., Kubler, O., Kikinis, R., Jolesz, A. Nonlinear anisotropic filtering of mri data. *IEEE Transactions on medical imaging* 1992;11(2):221–232.

[26] Hamza, A.B., Krim, H., Unal, G.B. Unifying probabilistic and variational estimation. *IEEE Signal Process Mag* 2002;2:37–47.

[27] Kačur, J., Mikula, K. Solution of nonlinear diffusion appearing in image smoothing and edge detection. *Appl Numer Math* 1995;17(1):47–59.

[28] Kačur, J., Mikula, K. Slow and fast diffusion effects in image processing. *Comput Visual Sci* 2001;3:185–195.

[29] Kilbas, A.A., Srivastava, H.M., Trujillo, J.J. *Theory and Applications of Fractional Differential Equations*. Elsevier, 2006.

[30] Lee, J.S. Digital image smoothing and the sigma filter. *Comput Vis Graph Image Process* 1983;24:253–269.

[31] López-Fernández, M., Lubich, C. Adaptive, fast, and oblivious convolution in evolution equations with memory. *SIAM J Sci Comp* 2008;30:1015–1037.

[32] López-Fernández, M., Palencia, C. On the numerical inversion of the Laplace transform of certain holomorphic mappings. *Appl Numer Math* 2004;51:289–303.

[33] Lubich, C. Convolution quadrature and discretized operational calculus I. *Numer Math* 1988;52:129–145.

[34] Lubich, C. Convolution quadrature and discretized operational calculus II. *Numer Math* 1988;52:413–425.

[35] Lubich, C. Convolution quadrature revisited. *BIT* 2004;44:503–514.

[36] Lubich, C., Ostermann, A. Runge–Kutta methods for parabolic equations and convolution quadrature. *Math Comput* 1993;60:105–131.

[37] Lysaker, M., Lundervold, A., Tai, X.C. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Trans Image Process* 2003;12(12):1579–1590.

[38] Mathieu, B., Melchior, P., Oustaloup, A., Ceyral, C. Fractional differentiation for edge detection. *Signal Process* 2003;83:2421–2432.

[39] Perona, P., Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans on Pattern Anal and Mach Intell* 1990;12(7):629–639.

[40] Portilla, J., Simoncelli, E.P. Image denoising via adjustment of wavelet coefficient magnitude correlation. In: *Proceedings of the 7th ICIP*. IEEE Computer Society; 2000. p. 10–13.

[41] Pruss, J. *Evolutionary Integral Equations and Applications*. Birkhäuser Verlag, Basel, 1993.

[42] Quintano, C., Cuesta, E. Improving satellite image classification by using fractional type convolution filtering. *Int J Appl Earth Obs* 2010;12:298–301.

[43] Rudin, L., Osher, S., Fatemi, E. Nonlinear total variation based noise removal algorithm. *Phys D* 1992;60:259–268.

[44] Solina, F., Kropatsch, W., Klette, R., (Eds.), R.B., editors. *A semidiscrete nonlinear scale–space theory and its relation to the Perona–Malik paradox*. *Advances in computer vision*, Springer, Wien, 1–10, 1997.

[45] Tomasi, C., Manduchi, R. Bilateral filtering for grey and color

- images. In: Proceedings of the 6th International Conference on Computer Vision, New Delhi, India. 1998. p. 839–846.
- [46] Weickert, J.. A review of nonlinear diffusion filtering. volume 1252. Springer, Berlin, 1997.
- [47] Weickert, J.. Anisotropic Diffusion in Image Processing. B.G. Teubner Stuttgart, 1998.
- [48] Weickert, J.. Coherence enhancing diffusion filtering. Int J of Computer Vision 1999;31:111–127.
- [49] Weickert, J., Brox, T.. Diffusion and regularization of vector- and matrix-valued images. Inverse Problems Image Analysis and Medical imaging Contemporary Mathematics 2002;313:251–268.
- [50] Yaroslavsky, L.P. Digital Picture Processing. An Introduction. Springer-Verlag, New York, 1985.
- [51] You, Y.L.. Fourth-order partial differential equations for noise removal. IEEE Trans Image Process 2000;9(10):1723–1730.