

On the Improvement of Volterra Equation Based Filtering for Image Denoising

Eduardo Cuesta¹, Mokhtar Kirane², and Salman A. Malik²

¹Department of Applied Mathematics, Industrial Engineering School, University of Valladolid, Spain.
eduardo@mat.uva.es

²Laboratoire MIA, Université de La Rochelle, 17042 La Rochelle Cedex, France.
mokhtar.kirane@univ-lr.fr
salman.malik@univ-lr.fr

Abstract—This paper presents a simple but effective approach for the removal of additive white Gaussian noise from digital images. In our approach, a generalization of a linear heat equation, obtained by replacing time derivative to a fractional time derivative of order between 1 and 2 has been used and a pixel by pixel technique applied. The choice of order of fractional time derivative has been made for each pixel by using structure tensor of image, which allows us to control the diffusion process without introducing nonlinearities in equation as in classical approaches. The proposed model is well posed and numerical scheme adopted is stable. Several experiments showing improvement of our approach visually and in terms of SNR, PSNR are also provided.

Keywords: Fractional integrals and derivatives, Volterra equations, Structure tensor, Convolution quadrature methods.

1. Introduction

Among several techniques available, for digital image processing (filtering, denoising, restorations, segmentation, edge enhancement/detection,...), partial differential equations based techniques are one of them, which have been largely studied in the literature (see [1] and references therein). It is well known in signal processing that the convolution of a signal with a Gaussian kernel acts like a low pass filter. The convolution of the signal with a Gaussian is equivalent to computing the solution of the linear heat equation

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

where Δ denotes the two-dimensional Laplacian, in case of images $\Omega \subset \mathbb{R}^2$ is typically a square domain, $\partial\Omega$ represents the boundary of Ω , $\partial/\partial\eta$ stands for the outward normal derivative, and u_0 the original image. The effect of equation (1), on digital images is isotropic, which yields loss of information about edges, textures and corners in practical applications of image restoration (denoising).

An anisotropic model seems to be a suitable approach to guarantee an edge preserving restoration (denoising). This approach was initially proposed by Perona and Malik in [2]; it reads

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

for $(t, \mathbf{x}) \in Q = [0, T] \times \Omega$. The diffusion coefficient $c : [0, +\infty) \rightarrow [0, +\infty)$, is chosen to preserve the edges and corners meanwhile smooths the uniform regions. Therefore, diffusion coefficient c , defined such that it has values close to zero where we want to preserve the information of the image (normally near edges, corners etc where gradient values of pixels are large). On the contrary, c should be large in pixels with low gradient variation. The choice of such functions lead to backward-forward parabolic problems that are ill-posed (see [3]). But, the implementation of Perona-Malik model doesn't reveal any such property of instability. The reason as reported by H. Amman [4] is that the numerical scheme used by Perona and Malik does not correspond to their equation but rather to a time-regularized one which is well-posed.

There is a need of regularization of Perona-Malik model such that new model inhibits the same practical results but lying in a *reasonable* functional space setting where the well-posedness can be guaranteed as well as the bounded variation and further analytical and numerical properties. In this regard, several perturbed models proposed (see e.g. [5], [6], [7], [8]).

In this paper, we consider a recent work of authors (see [9]), in which a linear Volterra matrix-valued equation, obtained by pixel by pixel technique has been proposed for image denoising. Volterra matrix-valued equation is a generalization of Volterra equation

$$u(t) = u_0 + \partial^{-\alpha} Au(t), \quad (3)$$

where $\partial^{-\alpha}$ is the Riemann-Liouville fractional integral of order $1 < \alpha < 2$. The value of α (between 1 and 2) for each pixel is obtained by a viscosity parameter (defined in

[9]). The profile of the viscosity parameter determines the diffusion process on the images, furthermore changing the profile of the viscosity parameter can change the nature of the filter proposed in [9].

The contribution of the present work is the proposal of an alternative approach for the selection of α for each pixel, i.e., a different profile for the viscosity parameter is proposed using the structure tensor of image. Previously, viscosity parameter (see [9]) allows us to handle the diffusion, as it exploits the nature of Volterra equation (3), which interpolates the linear heat equation and the linear wave equation (see [10]), for image denoising. Since the choice of α controls the diffusion process so the choice of α is crucial in implementation of the algorithm. The proposed method for choosing α , using structure tensor plays an important role and improve the results as seen in section (4). Well-posedness results, and a large variety of numerical discretizations of the proposed model have been closely studied by many authors, and therefore they are at our disposal for the experiments (see [11], [13], [14], [15] and references therein).

The rest of the paper is organized as follows: the next section presents the main idea of the paper i.e., selection of values of $1 < \alpha < 2$ for each pixel by using structure tensor of the image. In section 3 we provide the generalized matrix valued Volterra equation and its discretization in space and time. Section 4 illustrates the implementation of the proposed model and we provide some practical examples. Finally, section 5 concludes the work of this paper.

2. Volterra equations

The fractional calculus in image denoising has been firstly proposed in [11], where a generalization of the heat equation (1) is proposed which reads

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

where ∂_t^α stands for the Riemann–Liouville fractional time derivative of order $1 < \alpha < 2$; for an integrable function f is given by (see [12])

$$\partial_t^\alpha f(t) := \frac{1}{\Gamma(2-\alpha)} \frac{d^2}{dt^2} \int_0^t \frac{f(\tau)}{(t-\tau)^{\alpha-1}} d\tau. \quad (5)$$

The equation (4) is equivalent to the Volterra equation

$$u(t, \mathbf{x}) = u_0(\mathbf{x}) + \partial^{-\alpha} \Delta u(t, \mathbf{x}), \quad (6)$$

with homogeneous Neumann boundary condition along with $\partial_t u(0, \mathbf{x}) = 0$ and in equation (6), $\partial^{-\alpha}$, for $\alpha > 0$, stands for the Riemann–Liouville fractional integral of order $\alpha \in \mathbb{R}^+$

(see [12]) defined by

$$\partial^{-\alpha} f(t) := \frac{1}{\Gamma(\alpha)} \int_0^t \frac{f(\tau)}{(t-\tau)^{1-\alpha}} d\tau, \quad t > 0. \quad (7)$$

The main contribution of this work comes out as an alternative approach for the choice of α for each pixel. The choice of α proposed in [9], by using viscosity parameter, leaves some isolated pixels (see section 4) in the restored images. This is because, the noise plays its part while getting the value of α by viscosity parameter. Here, we use structure tensor (see [1]) for getting the eigenvalues of the image and use the well known fact that the eigenvalues near the edges, corners are greater than the eigenvalues of the smooth regions. This fact allows us to assign the value of α near 2 to the pixels which have large eigenvalues and 1 to contrary.

The structure tensor has been used for identifying several structures in images such as texture like flow, corners, T junctions etc, is defined as

$$J_\rho(\nabla u_\sigma) := K_\rho \star (\nabla u_\sigma \otimes \nabla u_\sigma) \quad \rho \geq 0, \quad (8)$$

where K_ρ is the Gaussian kernel, \star represents the convolution operator in space variable, $\nabla u_\sigma \otimes \nabla u_\sigma := \nabla u_\sigma \nabla u_\sigma^t$, ρ is the integration scale, σ is called local or noise scale. It is easy to calculate the expressions for eigenvectors and eigenvalues associated with structure tensor, for more details see [1]. The eigenvalues of the structure tensor provides us information about the edges corners of image, so the choice of α made as described above.

3. Space and time discretizations

Before we start spatial discretization of Laplacian, it is worthwhile to notice that $u(t, \mathbf{x})$ is transformed into a $MN \times 1$ vector-valued function $\mathbf{u}(t)$ whose components stand for the vector-arranged image pixels at time level t (M is length and N is the width of the image). Without loss of generality let $M = N$, for discretization of Laplacian in (4), a central difference scheme with a step size $h > 0$ has been considered. So, Δ transforms into a $M^2 \times M^2$ five-diagonals sparse matrix Δ_h , with the number of nonzero elements, $5M^2 - 4M$. The sparse nature of discretized Laplacian Δ_h is helpful in implementation as we store only nonzero elements for Δ_h , which makes algorithm robust when applied.

The equation (6), applied for the choice of α , for each pixel reads now as the linear Volterra matrix-valued equation given by

$$\mathbf{u}(t) = \mathbf{u}_0 + \int_0^t \mathbf{K}(t-s) \mathbf{u}(s) ds, \quad 0 \leq t \leq T, \quad (9)$$

where \mathbf{u}_0 is the initial image, and the convolution kernel \mathbf{K} is defined as

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

with $I(t)$ be diagonal matrix with diagonal entries $t^{\alpha_i} / \Gamma(\alpha_i + 1)$ for $i = 1, 2, \dots, M^2$ and for each i , $1 < \alpha_i <$

2. Let us notice that \mathbf{K} is a five-diagonal matrix valued function and, since the Laplace transform of \mathbf{K} exists, the well-posedness of (9) is then guaranteed (see e.g. [13]).

For the time discretizations of Volterra equation (9), Runge–Kutta convolution quadrature method has been adopted, which provide high order numerical methods jointly with good stability properties. These methods have been studied extensively in literature, e.g., the convolution quadrature based methods (see [14], [15] and references therein).

Let $\tau > 0$ be the time step of the discretization and \mathbf{u}_n be the approximation of $\mathbf{u}(t_n, \mathbf{x})$, for $n \geq 0$. Then the discretization of (9) by means of the backward Euler convolution quadrature method reads

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

where, $\mathbf{Q}_j^{(\alpha)}$ is a $M^2 \times M^2$ diagonal matrix for each $j = 0, 1, 2, \dots$, containing convolution quadrature weights. The entries of $\mathbf{Q}_j^{(\alpha)}$ are given by $\left(\alpha_j^{(i)}\right)$, $i = 0, 1, 2, \dots, M^2$ (see [9] and reference therein, for more details). Since the matrix Δ_h is not singular, the unique n -th approximation is reached by solving the linear system

$$\left(I - \mathbf{Q}_0^{(\alpha)}\right) \mathbf{u}_n = \mathbf{u}_0 + \sum_{j=1}^{n-1} \mathbf{Q}_{n-j}^{(\alpha)} \mathbf{u}_j, \quad n \geq 1. \quad (10)$$

4. Implementation and practical results

In discrete setting, for an efficient implementation of (10), some facts must be considered. For $\alpha = 1$ and $\alpha = 2$, singularities may occur from the numerical point of view. In order to avoid these possible singularities for practical purposes, we set $\alpha_j \in [1 + \epsilon, 2 - \epsilon]$ with $\epsilon = 10^{-3}$ for $j = 1, 2, \dots, M^2$.

For each α_j , a fixed number of convolution quadrature coefficients has been computed. Moreover, the practical computation of that weights has been carried out by means of the Fast Fourier Transform as in [16]. From a computational point of view, the number of different values of α 's should be limited, otherwise if one admits number of α 's as large as the number of pixels, the implementation becomes unavailable in practical cases.

The quality of processed image has been obtained by calculating signal to noise ratio (SNR) and peak signal to noise ratio ($PSNR$). The formula used for the SNR is

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

where R is the restored image, I is an ideal image, $\text{var}(x)$ stands for the variance of the vector x . The formula for $PSNR$ used 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.

We show some experiments where a noisy image is evolved by using the Volterra equation method (viscosity parameter for the choice of α) as in [9] (VEV), Perona–Malik model (2) with $c(s) = e^{-s}$ (PM) and the contrast parameter (λ) is chosen such that $1.5 * \sigma < \lambda < 2 * \sigma$ as suggested in [17] (where σ is the noise variation) and the model (9), with structure tensor method for choice of α (VES).

All the four images used for the experiments are of size 512×512 , i.e. on the spatial domain $\Omega = [0, 511] \times [0, 511]$ and are shown in Fig 1.(a-d). For all images, original (ideal) image has been perturbed by Gaussian white noise of variance ranging from 10 to 35 and resulting noisy images used as input for the evolution process of the three methods. In table (1), we show the noise level in terms of SNR , for the restoration of the noisy images yielded out with the above mentioned procedures. Notice that the proposed method (VES) has the best SNR (in decibels) for all the four images. Table (2), shows noise level of the processed images as a function of $PSNR$ (in decibels) level of the input noisy image, as in case of SNR for all four images the proposed method (VES) outperforms the (PM) and (VEV) denoising process.

The visual inspection for the images of Barbara, boats and fingerprint has been made in Figure 2 by considering a zoomed small part of the restored images. The part of Barbara's image of size 100×100 is considered (Figure 2(a)). The noisy image with $\sigma = 30$ has been evolved by the three process (PM), (VEV) and (VES) (the proposed method). The zoomed part of noisy image is given in (Figure 2.(b)) the restoration by (PM) model Figure 2.(c) removes the noise by losing the structure of the image. Notice that (PM) very strongly smooths the regions of small pixel value variations corresponding to texture of image, hence losing information regarding texture and structure of the image whereas the denoising results obtained by (VEV) and (VES) preserves the structure of the image. For the image of boats (with $\sigma = 30$), a part of size 200×200 has been zoomed for the visual inspection, the original and noisy images are given in Figure 2.(f-g). Notice that (PM) model (Figure 2.(h)) not only smooths the region of small variation but also creates the artifact (staircase effect) at the edges. The restoration by (VEV) (Figure 2.(i)) doesn't create any staircase effect on edges but leaves some isolated pixels, where as the restoration by the proposed method (VES) (Figure 2.(j)) removes the noise by preserving the structure (texture) of the image and without leaving any isolated pixel and without creating any artifacts.

It has been reported in [9] that the Volterra equation (Eq. (9)) based filtering removes the noise by preserving the structure of the images and hence works well for the images

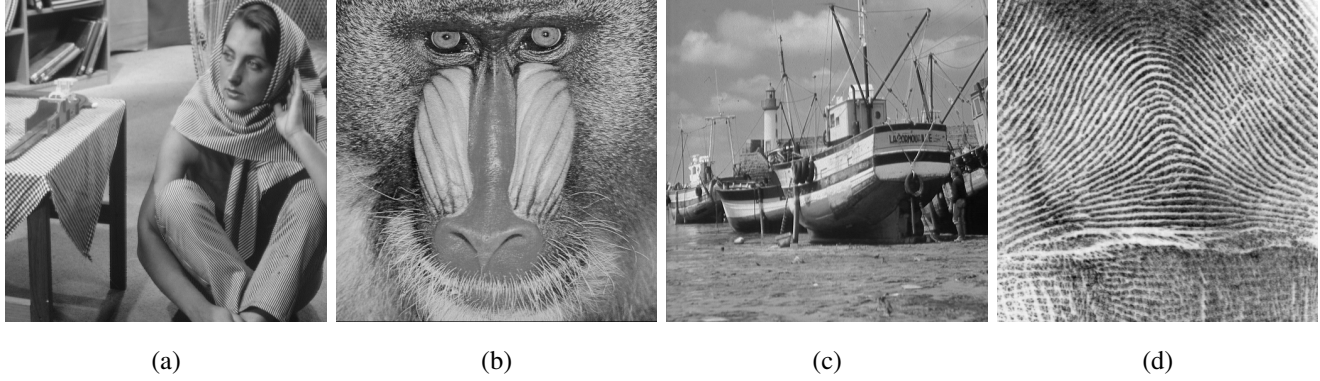


Fig. 1: Original image of (a) Barbara, (b) Baboon, (c) Boats, (d) Fingerprint.

Table 1: *SNR* analysis

σ	10	15	20	25	30	35	10	15	20	25	30	35
Input SNR	14.74	11.25	8.79	6.90	5.38	4.10	13.30	9.88	7.40	5.50	4.00	2.70
Method	Barbara (512×512)						Boats (512×512)					
(PM)	17.00	14.00	12.25	10.85	10.27	9.7	15.07	13.69	12.26	11.57	9.72	9.74
(VEV)	16.72	14.16	12.00	10.5	9.78	9.60	15.74	14.10	12.65	11.32	10.32	9.42
(VES)	17.31	15.20	13.95	11.85	10.80	10.30	16.84	14.88	13.50	12.52	11.81	10.69
Input SNR	14.00	10.60	8.20	6.30	4.80	3.50	12.50	9.00	6.50	4.50	3.00	1.78
Method	Fingerprint (512×512)						Baboon (512×512)					
(PM)	14.80	11.97	10.26	9.10	7.90	8.15	13.42	10.09	8.34	7.27	6.15	5.51
(VEV)	15.00	11.40	10.9	9.50	9.3	8	12.88	10.29	8.51	7.49	6.89	6.34
(VES)	17.00	15.11	13.60	12.59	11.80	10.52	13.87	11.01	9.50	8.52	7.65	7.28

where texture plays an important role (i.e. textured images). For this we consider the image of fingerprint (Figure 1.(d)) and a zoomed part of 150×150 has been investigated for the restoration process (Figure 2. (k-o)). The restoration by (PM) (Figure 2.(m)) as observed for the previous examples creates artifacts at edges and restoration by (VEV) (Figure 2.(n)) leave some isolated pixels. The proposed method (VES) (Figure 2.(o)) not only outperforms the two methods (PM), (VEV) visually but also in terms of *SNR* and *PSNR*.

In Figure 3.(a) we present the plot of *SNR* values of first 35 iterations for the restoration of a noisy image of Barbara ($\sigma = 25$) by (VEV) and (VES). Figure 3.(b) shows the plot of *SNR* values of first 45 iterations for the denoising (restoration) process of Baboon's image ($\sigma = 30$) by (VEV) and (VES). From both figures the improvement of the proposed method is evident. The difference of optimal stopping time for nonlinear and linear models restrict us to include *SNR* values for (PM) in the plots of Figure 3 (see [18]).

5. conclusion and future work

We have presented a novel approach for image denoising in the framework of fractional calculus using linear heat equation with fractional derivative in time of order $1 < \alpha < 2$, which is equivalent to matrix-valued Volterra equation. The assignment of α for each pixel of the image, explores the fact that large eigenvalues occur for edges and corner, where as structure tensor of the image has been used for getting the eigenvalues of the image. The practical results show improvements visually and in terms of *SNR*, *PSNR* when compared to the nonlinear Perona-Malik model and the model already proposed by authors in [9]. The proposed model is linear and well-posed. The numerical scheme used in discretization is convergent and preserves positivity of the solution. The extension of this work to the nonlinear models, involving structure tensor or fractional Laplacian (as nonlinearity in the model) will be the topic of our forthcoming papers.

References

- [1] J. Weickert, *Anisotropic Diffusion in Image Processing*, B.G. Teubner Stuttgart, 1998.

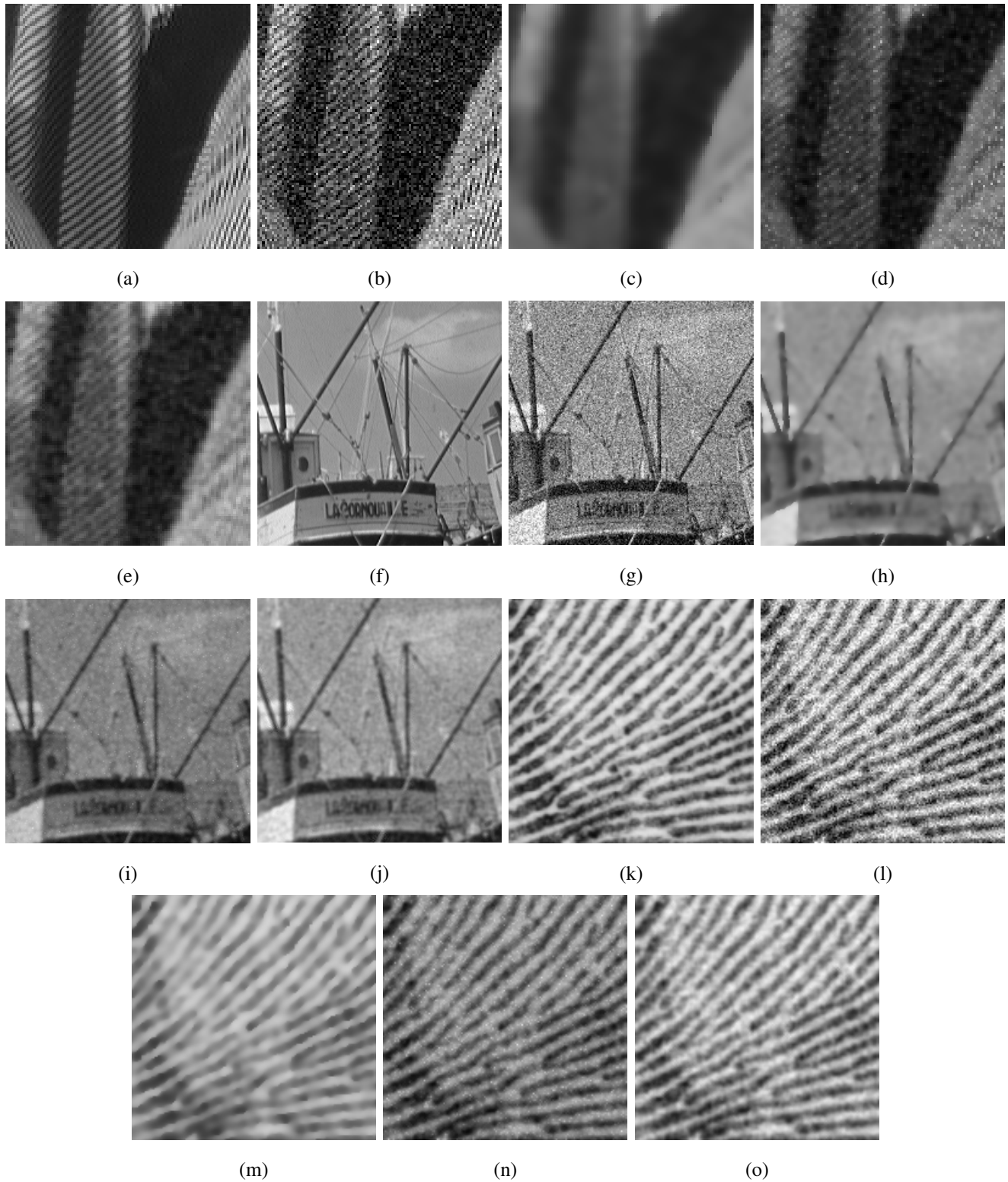


Fig. 2: (a) 100×100 zoomed part of original Barbara's image, (b) Zoomed part of Barbara's noisy image ($\sigma = 30$), (c) Restored zoomed part by (PM), (d) Restored zoomed part by (VEV), (e) Restored zoomed part by (VES), (f) 200×200 zoomed part of original Boats's image, (g) Zoomed part of Boats's noisy image ($\sigma = 30$), (h) Restored zoomed part by (PM), (i) Restored zoomed part by (VEV), (j) Restored zoomed part by (VES) (k) 150×150 part of Fingerprint's original image (l) Zoomed part of Fingerprint's noisy image ($\sigma = 30$) (m) Restored zoomed part by (PM) (n) Restored zoomed part by (VEV) (o) Restored zoomed part by (VES).

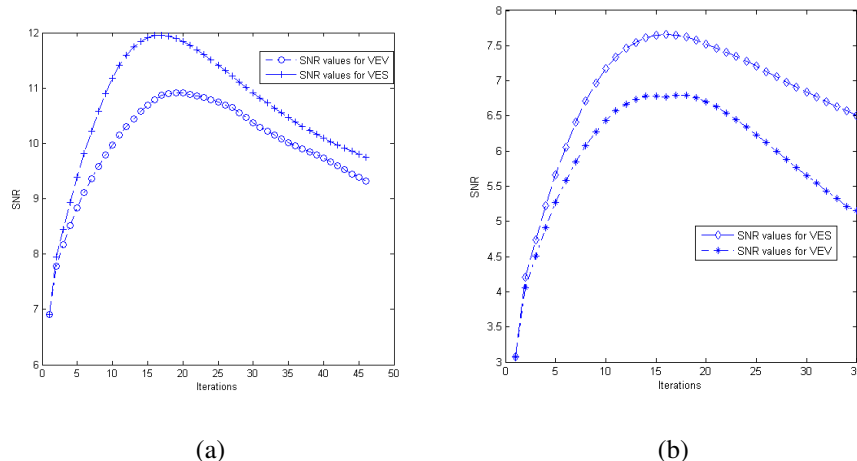


Fig. 3: (a) SNR values for the first 35 iterations of (VEV) and (VES) for the noisy image of Barbara with $\sigma = 25$, (b) SNR values for the first 45 iterations of (VEV) and (VES) for the noisy image of Baboon with $\sigma = 30$.

Table 2: $PSNR$ analysis

σ	10	15	20	25	30	35	10	15	20	25	30	35
Input PSNR	28.12	24.6	22.18	20.29	18.76	17.50	28	24.6	22	20.27	18.7	17.10
Method	Barbara (512×512)						Boat (512×512)					
(PM)	30.14	26.40	24.60	23.10	22.00	20.54	29.45	28.00	27.12	25.64	24.15	23.48
(VEV)	30.00	27.33	25.08	24.03	23.15	21.26	30.49	28.42	27.25	26.32	25.02	23.43
(VES)	32.00	28.76	26.37	25.00	24.34	23.74	31.57	29.61	28.14	27.05	25.08	24.00
Input PSNR	28.13	24.6	22.2	20.3	19	17	28	24.6	22	20.19	18.6	17.18
Method	Fingerprint (512×512)						Baboon (512×512)					
(PM)	28.81	25.91	24.21	23.05	21.87	20.01	28.44	24.91	22.62	22.57	21.75	20.46
(VEV)	29.17	27.33	25.37	24.00	22.42	20.9	29.02	25.92	24.07	22.87	22.34	20.82
(VES)	31.23	29.01	27.62	26.60	25.8	24.5	28.38	25.08	24.05	23.62	22.58	21.67

- [2] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. on Pattern Anal. and Mach. Intell.*, vol. 12, no. 7, pp. 629–639, July 1990.
- [3] S. Kichenassamy, "The persona-malik paradox," *SIAM J. Appl. Math.*, vol. 147, pp. 1328–1342, 1997.
- [4] H. Amann, "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, pp. 15–38, 2005.
- [5] F. Catte, P. L. Lions, J. M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numer. Anal.*, vol. 29, no. 1, pp. 182–193, 1992.
- [6] J. Kačur and K. Mikula, "Solution of nonlinear diffusion appearing in image smoothing and edge detection," *Appl. Numer. Math.*, vol. 17, no. 1, pp. 47–59, 1995.
- [7] J. Kačur and K. Mikula, "Slow and fast diffusion effects in image processing," *Computation and Visualization in Science*, vol. 3, pp. 185–185, 2001.
- [8] P. Guidotti and J. V. Lambers, "Two new nonlinear nonlocal diffusions for noise reduction," *J. Math. Imaging Vis.*, vol. 33, pp. 25–37, 2009.
- [9] E. Cuesta, M. Kirane, and S. A. Malik, "Image structure preserving denoising using generalized fractional time integrals," *preprint* <http://hal.archives-ouvertes.fr/hal-00437341/en/>.
- [10] Y. Fujita, "Integro-differential equation which interpolates the heat equation and the wave equation," *Osaka J. Math.*, vol. 27, pp. 319–327, 1990.
- [11] E. Cuesta and J. Finat, "Image processing by means of a linear integro-differential equation," *IASTED*, pp. 438–442, 2003.
- [12] A. A. Kilbas, H. M. Srivastava, and J. J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Elsevier, 2006.
- [13] J. Pruss, *Evolutionary Integral Equations and Applications*, Birkhäuser Verlag, Basel, 1993.
- [14] Ch. Lubich, "Convolution quadrature revisited," *BIT*, vol. 44, pp. 503–514, 2004.
- [15] Ch. Lubich and A. Ostermann, "Runge-kutta methods for parabolic equations and convolution quadrature," *Math. Comput.*, vol. 60, pp. 105–131, 1993.
- [16] M.P. Calvo, E. Cuesta, and C. Palencia, "Runge-kutta convolution quadrature methods for well-posed equations with memory," *Numer. Math.*, vol. 107, pp. 589–614, 2007.
- [17] G. Gerig, O. Kubler, R. Kikinis and A. Jolesz "Nonlinear Anisotropic Filtering of MRI Data," *IEEE Transactions on medical imaging* **11**, No. 2, 221–232 (1992)
- [18] P. Mrázek and M. Navara "Selection of Optimal Stopping Time for Nonlinear Diffusion Filtering," *Int. journal of Computer Vision*. **52**, 189–203 (2003)