International Journal of Computer Mathematics
Vol. 90, No. 1, Jan 2013, 140–161

A New Variational Model for Removal of Combined Additive and Multiplicative Noise and a Fast Algorithm for Its Numerical Approximation

Noppadol Chumchob, Ke Chen, and Carlos Brito-Loeza

Variational image restoration models for both additive and multiplicative noise removal are rarely encountered in the literature. This paper proposes a new variational model and a fast algorithm for its numerical approximation to remove independent additive and multiplicative noise from digital images. Two previous works by [L. Rudin, S. Osher and E. Fatemi, *Phys. D*, 60 (1992), pp. 259-268] and [Z. Jin and X. Yang, *J. Math. Anal. Appl.*, 362 (2010), pp. 415-426] are used to develop the new model. As a result, developing a fast numerical algorithm is difficult because the associated Euler-Lagrange equation is highly nonlinear and standard unilevel iterative methods are not appropriate. To this end, we develop an efficient nonlinear multigrid approach via a robust fixed-point smoother. Numerical tests using both synthetic and realistic images not only confirm that our new model delivers quality results but also that the proposed numerical algorithm allows a very fast numerical realization of the model.

Keywords. Additive noise, image denoising, multiplicative noise, nonlinear multigrid, total variation, variational models.

1 Introduction

In many real world applications, images are usually contaminated with noise, either because of the image acquisition process, or because of naturally occurring phenomena. Therefore, the process of estimating the unknown image of interest from the available noisy image, known as image denoising or image restoration, plays an important role in various areas. Applications that require a restoration step range from art, astronomy, astrophysics, biology, chemistry, criminology, geophysics, physics, and other areas involving imaging techniques.

As is well known, the so-called additive noise (AN) model is commonly found in acquiring images via digital devices. Not surprisingly, most of the literatures mainly deal with this type of image formation models. In the AN model, one sought to recover an original (unknown) image $u : \Omega \subset \mathbb{R}^2 \to V \subset \mathbb{R}_0^+ = \mathbb{R}^+ \cup \{0\}$ from the noisy (known) image $z : \Omega \to V$ corrupted by some unknown AN η as follows:

$$z = u + \eta. \tag{1}$$

There are different types of noise. However, the so-called additive zero-mean Gaussian white noise has been attracted much attention and extensively investigated for many years. We refer the reader to the literature [5, 21] and

N. Chumchob is with the Department of Mathematics, Faculty of Science, Silpakorn University, Nakorn Pathom 73000, Thailand and Centre of Excellence in Mathematics, CHE, Si Ayutthaya Rd., Bangkok 10400, Thailand (e-mail: cnoppado@su.ac.th).
Ke Chen is with the Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, United Kingdom (e-mail:K.Chen@liv.ac.uk).
Carlos Brito-Loeza is with the Universidad Autónoma de Yucatán, Facultad de Matemáticas Anillo Periférico Norte, Tablaje Cat. 13615, Colonia Chuburná Hidalgo Inn, Mérida Yucatá (e-mail: carlos.brito@uady.mx).
All authors are with the Centre for Mathematical Imaging Techniques, Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, United Kingdom (www.liv.ac.uk/cmit; cmit@liv.ac.uk).

ISSN 1029-4937 print; ISSN 0020-7160 online © 2013 Taylor & Francis Ltd.
DOI:

references therein for an overview of the subject. Without loss of generality, we assume that $\Omega = [0,1]^2 \subset \mathbb{R}^2$ and $V = [0,1]$ for 2D gray intensity images throughout this paper.

According to [3, 4, 36–41, 45, 50, 52] and other references, there are many practical applications involving the so-called multiplicative noise (MN), also known as *speckle*. It is commonly found in the coherent imaging system, such as synthetic aperture radar (SAR) and sonar, and ultrasound and laser imaging. In the MN model, the original image has been corrupted by some unknown MN η as given by

$$z = u\eta. \tag{2}$$

Without loss of generality, we can assume that u and η in (2) are positive in the noise model. Unlike AN models, the noisy signals in the corrupted images are much more difficult to be removed, mainly not only because of the multiplicative nature between the noise and the original image, but also because of the noise distributions, which are generally non-Gaussian, commonly assumed to be Gamma and Rayleigh distributions.

Over the past decade, a wide variety of AN and MN removal methods have been proposed to recover the true image from the noise models given by (1) and (2). These methods include traditional filtering [3,41], wavelets techniques [8, 29], stochastic approaches [31], principal component analysis-based approaches [61,62], and variational methods [4,5,13,21,37–40,44,45,50]. It has long been known that finding u via (1) or (2) is a very ill-conditioned problem because the solution is highly sensitive to the noise η. The variational approaches with total variation (TV)-based regularization, in particular the first TV-based AN removal model by Rudin, Osher, and Fatemi [44] (also known as ROF model) and the first TV-based MN removal model by Rudin, Lions, Osher [45] (also known as RLO model), are among the most famous ones to offer superior image restoration quality. However, much improvement is still required.

From a practical point of view, a fundamental assumption concerning only pure AN or MN cannot be always true. For instance, recent studies by Hirakawa and Parks [36] and Lukin et al. [52] confirm this fact. In [36], the results show that

$$z = u + (k_0 + k_1 u)\eta, \tag{3}$$

where k_0 and k_1 are constants and η represents Gaussian white noise with variance 1, is the good noise model for Agilent Technologies camera evaluation board HDCP-2000 equipped with 300 K pixel CMOS sensor and capturing raw sensor data. In [52], the authors reported that the AN component cannot be neglected, especially for images formed by side look aperture radars (SLARs).

In order to restore the observed images corrupted by the combination of the AN and MN, a feasible option is to apply a two-step approach by first using an additive-restoration technique followed by a multiplicative-restoration one or the other way round. However, this two-step approach does not appear to be optimal due to increased high computational cost and approximation of (3). Ideally one would like to use only a one-step method. As far as we know, there are few image restoration techniques devoted to this type of the AN and MN noise removal problems. For example, in [36] a robust method for restoring the observed images corrupted with the additive, multiplicative, and mixed noise was introduced by constructing a so-called *image-patch model* from a linear combination of noisy patches and then fitting this model to the real-world image in the total least square sense. In [52] a discrete cosine transform method based on local adaptive filtering was developed to remove noise in automatic or blind manner. Although these former two methods offer good image quality, they are computationally expensive for high resolution images.

To the best of our knowledge, there are no variational models devoted to the problem of removing noise from images corrupted by mixed AN and MN. The difficulties are not only to design an effective image restoration model, but also to develop an efficient numerical solution for the associated large system of nonlinear equations arising from high resolution images. The aim of this project is to cover both aspects by designing a new TV-based image restoration model and developing a robust nonlinear multigrid (MG) method.

The outline of this paper is organized as follows. Section 2 reviews two effective models for removing AN and MN using TV regularization. Section 3 introduces our new variational model, followed by numerical techniques for solving the associated Euler-Lagrange equation in Section 4. Section 5 presents an efficient MG method following analysis the above proposed iterative solvers as MG smoothers. Experimental results from synthetic and realistic images illustrating the effectiveness of the new model and the efficiency of our MG method are shown in Section 6 before conclusions in Section 7.

2 Review of TV-based image restoration models

In the literature, TV-based regularization has been proven to be a very valuable tool for image restoration, and is used extensively in many practical applications. However, it has to combine with a suitably chosen data fitting term in leading to excellent image restoration results. This section first reviews the current state-of-the-art models for the AN and MN removal problems from which our new model was developed. As can be seen, each model includes the nonlinear fitting term able to offer very high quality of restored images.

2.1 *TV-based AN removal model*

Among all variational approaches for the AN removal problems, the classical ROF model [44] represented by

$$\arg\min_{u\in BV(\Omega)} \left\{ \int_\Omega |\nabla u| d\Omega + \tfrac{\alpha}{2} \int_\Omega (u-z)^2 d\Omega \right\} \tag{4}$$

is well known for preserving sharp edges in recovered images. Here $BV(\Omega)$ denotes the bounded variation space and $\alpha > 0$ is the regularization parameter deciding the amount of the noise to be removed. In the literature, a variety of fast numerical methods have been extensively studied, e.g. [1, 5, 14–19, 21, 22, 27, 28, 30, 47, 53–56, 60]. This makes the ROF model a very popular approach in the area of image restoration.

2.2 *TV-based MN removal model*

The MN noise models have been extensively studied over the last decades. However, in the literature the RLO model [45]

$$\arg\min_{u\in BV(\Omega)} \left\{ \int_\Omega |\nabla u| d\Omega + \alpha_1 \int_\Omega \frac{z}{u} d\Omega + \alpha_2 \int_\Omega (\frac{z}{u} - 1)^2 d\Omega \right\} \tag{5}$$

and the Aubert-Aujol (AA model) [4]

$$\operatorname*{arg\,min}_{u\in S(\Omega)}\left\{\int_\Omega |\nabla u|d\Omega+\alpha\int_\Omega(\log u+\frac{z}{u})d\Omega\right\} \tag{6}$$

are two main variational models, where $S(\Omega)=\{u\in BV(\Omega), u>0\}$ and α_1, α_2 are the weighted parameters. We note first that the fitting terms of RLO model (5) are under the assumption that the mean of the MN is equal to 1 and the variance is known. We also note that the fitting term of AA model derived from a maximum a posteriori (MAP) estimation is specifically devoted to recover the true image u from a noisy image z, corrupted by Gamma noise with mean equal to 1. As far as we know, the theoretical analysis of (5) has not yet been studied thoroughly, whereas some interesting analysis of (6) was done in [4].

Recently, Jin and Yang [39] applied the exponential transformation $u\to e^u$ introduced by [38] with the fitting term of AA model and proposed the following restoration model

$$\operatorname*{arg\,min}_{u\in BV(\Omega)}\left\{\int_\Omega |\nabla u|d\Omega+\alpha\int_\Omega(u+ze^{-u})d\Omega\right\}. \tag{7}$$

They proved the existence and uniqueness of a minimizer to the variational problem (7) in $BV(\Omega)$. To compute a numerical solution, the authors used an explicit method for the evolution equation corresponding to (7).

As can been seen, both models (5) and (7) are based on the TV regularization and only differ in their fitting terms. Other regularization terms may be used to improve their results and avoid the staircase effect; see higher order model in [13] and references therein.

3 The proposed variational image restoration model

We now introduce our new variational model for restoring images corrupted by both AN and MN. Assume that AN and MN are independent or, in other words, not related to each other. This situation is likely to happen in many real-life applications when they come from two different sources. For instance, synthetic aperture radar (SAR) images are usually corrupted by the MN as a consequence of image formation under coherent radiation [2] and the AN due to thermal vibrations in the electronic components of the image capture devices. Thus an appropriate image formation model for this case is

$$z=u+k_0\eta+k_1u\zeta, \tag{8}$$

where η and ζ are the AN and MN components, respectively.

Most multiplicative models usually assume that η is less significant than ζ and therefore dropping the first two terms from (8) leads to the MN model as given in (2). In this work we do not drop any term from (8) and propose the following minimization to remove both AN and MN:

$$\operatorname*{arg\,min}_{u\in BV(\Omega)}\{J[u]=\int_\Omega |\nabla u|d\Omega+\frac{\alpha_1}{2}\int_\Omega(u-z)^2d\Omega+\alpha_2\int_\Omega(u+ze^{-u})d\Omega\}, \tag{9}$$

where $\alpha_1,\alpha_2>0$ are the regularization parameters used to balance between the AN and MN fitting terms.

3.1 *Existence and uniqueness of the solution*

Note that existence of the solution can be proved easily using the standard arguments. To this end, we only need to show BV-coercivity, weakly lower semi-continuity of $J[u]$. The BV-coercivity of $J[u]$ with $u \in L^1(\Omega)$ follows provided $J[u] \to +\infty$ as $\|u\|_{BV} \to +\infty$, where $\|\cdot\|_{BV}$ is the total variation norm, and is analogous to $\int_\Omega |\nabla u| d\Omega$ for $u \in C_0^1(\Omega)$; see [1]. For (9), we have

$$\int_\Omega (u-z)^2 d\Omega \geq 0 \tag{10}$$

and

$$\int_\Omega (u + ze^{-u}) d\Omega \geq 0. \tag{11}$$

It follows that

$$\|u\|_{BV} \leq J[u]$$

up to a constant. Thus, the coercivity condition holds. Now, the weakly lower semi-continuity of (10) comes from the weakly lower semi-continuity of the norms on Banach spaces. The lower semi-continuity also holds for (11) as proven in [39] and for $\|u\|_{BV}$ as proven in [1]. Finally, with $J[u]$ being lower semi-continuous and BV-coercive, the existence of the solution is guaranteed by [1, Thm 3.1]. The solution is unique due to each term in $J[u]$ being convex, as also shown in [1,39].

3.2 *The Euler-Lagrange equation*

The minimizer of the energy functional J in (9) satisfies the Euler-Lagrange (EL) equation given by the following nonlinear partial differential equation (PDE):

$$-\mathcal{K}(u) + \alpha_1(u-z) + \alpha_2(1 - ze^{-u}) = 0 \text{ in } \Omega \tag{12}$$

subject to the natural boundary conditions:

$$\frac{\partial u}{\partial \boldsymbol{n}} = 0 \text{ on } \partial\Omega, \tag{13}$$

where $\mathcal{K}(u) = \nabla \cdot (\frac{\nabla u}{|\nabla u|_\beta})$, $|\nabla u|_\beta = \sqrt{|\nabla u|^2 + \beta}$, $\beta > 0$ is a small constant to avoid division by zero, and $\boldsymbol{n}$ is the unit outward normal vector on the image boundary $\partial\Omega$. We refer the reader to [39,44] for deriving (12).

4 Discretization of the EL equation and its numerical methods

In this section, we outline the finite difference (FD) discretization and discuss some numerical methods for (12).

4.1 *Finite difference discretization*

For sake of clarity, let the discrete domain be denoted by

$$\Omega_h = \left\{ \mathbf{x} \in \Omega | \mathbf{x} = (x_i, y_j)^\top = (\tfrac{(2i-1)h_x}{2}, \tfrac{(2j-1)h_y}{2})^\top, 1 \le i \le n_x,\ 1 \le j \le n_y \right\},$$

consisting of $n_x \times n_y$ cells of size $h_x \times h_y$ with grid spacing $\boldsymbol{h} = (h_x, h_y) = (1/n_x, 1/n_y)$ and let $(u^h)_{i,j} = u^h(x_i, y_j)$ denote the grid function. Applying finite difference (FD) approximations based on the cell-centered grid points to discretize (12), the discrete EL equation at a grid point (i, j) over Ω_h is given by

$$\underbrace{-\mathcal{K}^{\boldsymbol{h}} \left(u^{\boldsymbol{h}}\right)_{i,j} + \alpha_1(\left(u^{\boldsymbol{h}}\right)_{i,j} - \left(z^{\boldsymbol{h}}\right)_{i,j}) + \alpha_2(1 - \left(z^{\boldsymbol{h}}\right)_{i,j} e^{-(u^{\boldsymbol{h}})_{i,j}})}_{\mathcal{N}^{\boldsymbol{h}}(u^{\boldsymbol{h}})_{i,j}} = \left(g^{\boldsymbol{h}}\right)_{i,j} \tag{14}$$

with the following notation

$$\mathcal{K}^{\boldsymbol{h}} \left(u^{\boldsymbol{h}}\right)_{i,j} = \left[\frac{\delta_x^-}{h_x} \left(\frac{D\left(u^{\boldsymbol{h}}\right)_{i,j} \delta_x^+ (u^{\boldsymbol{h}})_{i,j}}{h_x} \right) + \frac{\delta_y^-}{h_y} \left(\frac{D\left(u^{\boldsymbol{h}}\right)_{i,j} \delta_y^+ (u^{\boldsymbol{h}})_{i,j}}{h_y} \right) \right].$$

Here $\left(g^{\boldsymbol{h}}\right)_{i,j} = 0$; on the finest grid in MG setting this notation is useful shortly. Without loss of generality, in this work we consider the simple case $n = n_x = n_y$ and $h = h_x = h_y = 1/n$ for all grid functions. This leads to

$$\begin{aligned}
-\mathcal{K}^h \left(u^h\right)_{i,j} &= (1/h^2)((\Sigma^h)_{i,j}(u^h)_{i,j} - (\overline{\Sigma}^h)_{i,j}(u^h)_{i,j}), \\
\left(\Sigma^h\right)_{i,j} (u^h)_{i,j} &= (D_1(u^h)_{i,j} + D_2(u^h)_{i,j} + 2D_3(u^h)_{i,j})(u^h)_{i,j}, \\
(\overline{\Sigma}^h)_{i,j}(u^h)_{i,j} &= D_1(u^h)_{i,j}(u^h)_{i-1,j} + D_2(u^h)_{i,j}(u^h)_{i,j-1}) + D_3(u^h)_{i,j}((u^h)_{i+1,j} + (u^h)_{i,j+1}), \\
D\left(u^h\right)_{i,j} &= (\sqrt{(\delta_x^+(u^h)_{i,j}/h)^2 + (\delta_y^+(u^h)_{i,j}/h)^2 + \beta})^{-1}, \\
D_1\left(u^h\right)_{i,j} &= D\left(u^h\right)_{i-1,j}, \quad D_2\left(u^h\right)_{i,j} = D\left(u^h\right)_{i,j-1}, \quad D_3\left(u^h\right)_{i,j} = D\left(u^h\right)_{i,j}, \\
\delta_{x_1}^{\pm}\left(u^h\right)_{i,j} &= \pm\left((u^h)_{i\pm1,j} - (u^h)_{i,j}\right), \quad \delta_{x_2}^{\pm}\left(u_l^h\right)_{i,j} = \pm\left((u^h)_{i,j\pm1} - (u^h)_{i,j}\right).
\end{aligned}$$

We note that the symbols h and $(\cdot, \cdot)^h$ will be dropped for simplicity in the following section.

4.2 *Time marching methods*

As is well known, a time marching method is one of convenient ways to solve the resulting EL equations like (12). The main idea is to introduce an artificial time variable t and compute the steady-state solution of the time-dependent nonlinear PDE of the form:

$$\partial_t u(\mathbf{x}, t) + \mathcal{N}(u(\mathbf{x}, t)) = g(\mathbf{x}).$$

In order to overcome the nonlinearity of $\mathcal{N}$, the so-called *Euler's explicit* *scheme* can be conveniently applied, and the iteration is then given by

$$\partial_t u(\mathbf{x}, t_{k+1}) = g(\mathbf{x}) - \mathcal{N}(u(\mathbf{x}, t_k)),\ \ k = 0, 1, 2, 3, ...,$$

where $u(\mathbf{x}, t_0)$ is some initial solution, typically $u(\mathbf{x}, t_0) = z(\mathbf{x})$.

For the time discretization we introduce a time-step $\tau > 0$, and then u is updated at the time step $k + 1$ by

$$u(\mathbf{x}, t_{k+1}) = u(\mathbf{x}, t_k) + \tau \left[g(\mathbf{x}) - \mathcal{N}(u(\mathbf{x}, t_k))\right]. \tag{15}$$

Therefore, the FD discretized version of equation (15) given in Section 4.1 can be represented at a grid point (i, j) by

$$(u^{[k+1]})_{i,j} = (u^{[k]})_{i,j} + \tau[(g)_{i,j} - \mathcal{N}(u^{[k]})_{i,j}]. \tag{16}$$

We note that this numerical scheme is easy to implement, but extremely slow to converge because the length of the time-step τ is required to be a very small number for stability reasons. For the second order PDEs like (16), $\tau \sim O(h^2)$.

In order to speed up the convergence of (16), we may apply the fully implicit scheme, and then $(u^{[k+1]})_{i,j}$ is updated by

$$(u^{[k+1]})_{i,j} = (u^{[k]})_{i,j} + \tau[(g)_{i,j} - \mathcal{N}(u^{[k+1]})_{i,j})]. \tag{17}$$

To overcome the nonlinearity of $\mathcal{N}$, we may globally linearize (17) respect to the $(k+1)$th time-step using the method of 'frozen coefficients' as well known for variational approaches related to the TV operator (see e.g., [6,15–17,22,30, 47,48]), and obtain the semi-implicit scheme as given by the following system of linear elliptic PDEs:

$$[1+\tau\alpha_1-\tau\mathcal{K}^{\text{lin}}(u^{[k]})_{i,j}](u^{[k+1]})_{i,j} = (u^{[k]})_{i,j}+\tau(\alpha_1(z)_{i,j}+\alpha_2((z)_{i,j}e^{-(u^{[k]})_{i,j}})-1), \tag{18}$$

where

$$-\mathcal{K}^{\text{lin}}(u^{[k]})_{i,j}(u^{[k+1]})_{i,j} = (1/h^2)((\Sigma^{[k]})_{i,j}(u^{[k+1]})_{i,j} - (\overline{\Sigma}^{[k]})_{i,j}(u^{[k+1]})_{i,j}).$$

Therefore, the update formula determined by a lexicographical ordering in a matrix-vector form can be written as

$$\mathbf{u}^{[k+1]} = ((1 + \tau\alpha_1)\mathbf{I}+\tau\mathcal{K}^{\text{lin}}(\mathbf{u}^{[k]}))^{-1}(\mathbf{u}^{[k]} + \tau(\alpha_1\mathbf{z} - \alpha_2(1 - \mathbf{z}e^{-\mathbf{u}^{[k]}})), \tag{19}$$

where $\mathbf{I}$ is the identity matrix. Note that the additive operator splitting (AOS) scheme by [57] can be applied with (19) in speeding up this time marching scheme.

4.3 *Fixed point methods*

As is well known, fixed point (FP) methods are successfully applied in solving the EL equations related to the TV minimization; see e.g. [6, 10–13, 20, 23–26, 30, 37, 47, 48, 55]. In this section we present two different FP methods in solving the EL equation (12).

4.3.1 ***Global fixed-point (GFP) method.*** For the first FP method, the nonlinear terms $1/|\nabla u|_\beta$ and e^{-u} represented in (12) or $D\,(u)_{i,j}$and $e^{-(u)_{i,j}}$ in (14) may be linearized or frozen globally at a previous FP step ν. This yields the resulting linearized system

$$\mathbf{N}[u^{[\nu]}]u^{[\nu+1]} = \mathbf{G}[u^{[\nu]}], \quad \nu = 0, 1, 2, ... \tag{20}$$

where

$$\mathbf{N}[u^{[\nu]}] = -\nabla \cdot (\tfrac{\nabla}{|\nabla u^{[\nu]}|_\beta}) + \alpha_1 \mathbf{I}$$

and

$$\mathbf{G}[u^{[\nu]}] = g + \alpha_1 z + \alpha_2 (z e^{-u^{[\nu]}} - 1),$$

typically $u^{[0]} = z$. Classified by its ingredients, we shall name this FP method the global FP (GFP) method.

As a common way to solve (20) for each GFP or outer step ν, we use the successive over-relaxation (SOR) method with the relaxation parameter $\omega \in (0, 2)$ and then the new step at a grid point (i, j) is given by

$$(u^{[\nu+1,k+1]})_{i,j} = (1 - \omega)\, u^{[\nu+1,k]})_{i,j} + \omega (\mathbf{N}[u^{[\nu]}])^{-1}_{i,j} (\mathbf{G}[u^{[\nu,k+1/2]}])_{i,j}, \tag{21}$$

where

$$(\mathbf{N}[u^{[\nu]}])^{-1}_{i,j} = \tfrac{1}{(1/h^2)(\Sigma^{[\nu]})_{i,j} + \alpha_1} \tag{22}$$

and

$$(\mathbf{G}[u^{[\nu,k+1/2]}])_{i,j} = (g)_{i,j} + \alpha_1 (z)_{i,j} + \alpha_2 ((z)_{i,j} e^{-(u^{[\nu]})_{i,j}} - 1) + (1/h^2)(\overline{\Sigma}^{[\nu]})_{i,j} (u^{[\nu]})_{i,j}. \tag{23}$$

Here the superscripts k, $k + 1/2$, and $k + 1$ denote the current, intermediate and new approximations computed by the SOR method, respectively.

Obviously, for each GFP step ν the linearized system (20) is strictly or irreducibly diagonally dominant. This guarantees the existence of a unique solution and global convergence of the SOR iterations [43, 46]. Moreover, the GFP method shows the interaction between the outer iteration that overcomes the nonlinearity of the discrete operator $\mathcal{N}$ in (14) at each outer step ν and the SOR method that solves the resulting linear system of equations at each corresponding inner step k. Instead of solving the linearized system (20) with very high precision, the SOR method or inner iteration can perform only a *few iterations* to obtain an approximate solution at each outer step ν. This is likely the so-called *inexact lagged-diffusivity FP* method which have been widely used for solving other problems in image processing applications related to the TV operator; see e.g. [6,10–13,20,23–26,30,37,47,48,55]. This procedure leads to a slight difference of convergence in the GFP scheme when it is used as a stand-alone solver, whereas the computational costs significantly reduce. Moreover, the relaxation parameter ω has a strong influence on the convergence speed. We usually use $\omega > 1$, typically $\omega = 1.5$, because it results in speeding up the convergence by many orders of magnitude faster than those of the Gauss-Seidel (GS) approach ($\omega = 1$). We also note that other basic iterative techniques such as line relaxation or preconditioned conjugate gradient (PCG) method may also be used as an inner solver but they are computationally more expensive than the SOR method and therefore not recommended.

Finally, the numerical implementation to compute one iteration of the proposed GFP method (20) based on the SOR method (21) can be summarized as follows:

Algorithm 4.1 Our Proposed GFP method

Denote by

v	the restored image
z	the noisy image
g	the RHS (right-hand side) term of (14)
α_1	the weighted parameter of the ROF fitting term in (9)
α_2	the weighted parameter of the JY fitting term in (9)
ω	relaxation parameter
MAX_{SOR}	the maximum number of SOR iterations

$$[v] \leftarrow GFP\,(v, z, g, \alpha_1, \alpha_2, \omega, MAX_{SOR})$$

```
• Use input arguments to compute $(\mathbf{N}[v])_{i,j}^{-1}$ (22) and
  the first three terms of $(\mathbf{G}[v])_{i,j}$ in (23) for all $1 \le i, j \le n$
• Perform SOR steps for solving (20)
    – for $k = 1 : MAX_{SOR}$ do
        – for $i = 1 : n$ do
            – for $j = 1 : n$ do
                    – Compute the last term of $(\mathbf{G}[v])_{i,j}$ in (23)
                    – Updating $(v^{[k+1]})_{i,j}$ using (21)
            – end
        – end
    – end
```

4.3.2 ***Local fixed-point (LFP) method.*** Apart from global linearization, the alternative approach for solving the nonlinear discrete systems like (14) is to use methods using only *local* linearization; see e.g. [6, 10–13, 47, 48] and references therein. The main idea is to solve a single nonlinear equation in the given nonlinear system for a (single) unknown using a numerical method of nonlinear equations in one variable.

More precisely, consider the corresponding nonlinear equation for the unknowns $(u)_{i,j-1}, (u)_{i-1,j}, (u)_{i,j}, (u)_{i+1,j}$ $(u)_{i,j+1}$ given by (14) as follows:

$$(1/h^2)((\Sigma)_{i,j}(u)_{i,j}-(\overline{\Sigma})_{i,j}(u)_{i,j})+\alpha_1((u)_{i,j}-(z)_{i,j})+\alpha_2(1-(z)_{i,j}\,e^{-(u)_{i,j}}) = (g)_{i,j}\,.$$

Therefore, at the kth iteration a nonlinear GS step is given by

$$\begin{aligned}&(1/h^2)((\Sigma^{[k+1]})_{i,j}(u^{[k+1]})_{i,j} - (\overline{\Sigma}^{[k+1]})_{i,j}(u^{[k+1]})_{i,j})\\ &\qquad +\alpha_1((u^{[k+1]})_{i,j} - (z)_{i,j}) + \alpha_2(1 - (z)_{i,j}\,e^{-(u^{[k+1]})_{i,j}}) = (g)_{i,j}\,,\end{aligned} \tag{24}$$

where

$$\begin{aligned}(\Sigma^{[k+1]})_{i,j}(u^{[k+1]})_{i,j} &= (D_1(u^{[k+1]})_{i,j} + D_2(u^{[k+1]})_{i,j} + 2D_3(u^{[k+1]})_{i,j})(u^{[k+1]})_{i,j},\\ (\overline{\Sigma}^{[k+1]})_{i,j}(u^{[k+1]})_{i,j} &= D_1(u^{[k+1]})_{i,j}(u^{[k+1]})_{i-1,j} + D_2(u^{[k+1]})_{i,j}(u^{[k+1]})_{i,j-1})\\ &\quad + D_3(u^{[k+1]})_{i,j}((u^{[k+1]})_{i+1,j} + (u^{[k+1]})_{i,j+1}).\end{aligned}$$

If the nonlinear terms $D_*(u^{[k+1]})_{i,j}$ and $e^{-(u^{[k+1]})_{i,j}}$ are simply replaced by $D_*(u^{[k]})_{i,j}$ and $e^{-(u^{[k]})_{i,j}}$, we obtain the so-called *Gauss-Seidel-fixed point* or *Gauss-Seidel-Picard* relaxation and we shall name this numerical scheme the local FP (LFP) method. As a result, we found experimentally that this relaxation method is inefficient in leading to fast convergence. An improvement can be simply obtained by using a few more steps of FP iterations with respect to the relaxation parameter $\omega \neq 1$ (typically $MAX_{FP} = 2$ and $\omega = 1.6$, where MAX_{FP} denotes the maximum number of FP iterations) as follows:

$$(u^{[k+1]})_{i,j} = (1-\omega)\,(u^{[k]})_{i,j} + \omega(\overline{u}^{[k+1]})_{i,j}, \tag{25}$$

where

$$(\overline{u}^{[k+1]})_{i,j} = \frac{(\mathbf{G}[u^{[\nu]}])_{i,j} + (1/h^2)((\overline{\Sigma}^{[k;\nu]})_{i,j}(u^{[k+1/2;\nu]})_{i,j})}{(1/h^2)(\Sigma^{[k;\nu]})_{i,j} + \alpha_1}, \tag{26}$$

$$(\mathbf{G}[u^{[\nu]}])_{i,j} = (g)_{i,j} + \alpha_1\,(z)_{i,j} + \alpha_2((z)_{i,j}\,e^{-(u^{[\nu]})_{i,j}} - 1), \tag{27}$$

$$\begin{aligned}(\overline{\Sigma}^{[k;\nu]})_{i,j}(u^{[k+1/2;\nu]})_{i,j} &= D_1(u^{[k;\nu]})_{i,j}(u^{[k+1]})_{i-1,j} + D_2(u^{[k;\nu]})_{i,j}(u^{[k+1]})_{i,j-1}\\ &\quad D_3(u^{[k;\nu]})_{i,j}((u^{[k]})_{i+1,j} + (u^{[k]})_{i,j+1})\end{aligned} \tag{28}$$

$$(\Sigma^{[k;\nu]})_{i,j} = D_1(u^{[k;\nu]})_{i,j} + D_2(u^{[k;\nu]})_{i,j} + 2D_3(u^{[k;\nu]})_{i,j}+, \tag{29}$$

$$D_1(u^{[k;\nu]})_{i,j} = (\sqrt{(((u^{[\nu]})_{i,j} - (u^{[k+1]})_{i-1,j})/h)^2 + (((u^{[k]})_{i-1,j+1} - (u^{[k+1]})_{i-1,j}))^2 + \beta})^{-1},$$

$$D_2(u^{[k;\nu]})_{i,j} = (\sqrt{(((u^{[k+1]})_{i+1,j-1} - (u^{[k+1]})_{i,j-1})/h)^2 + (((u^{[\nu]})_{i,j} - (u^{[k+1]})_{i,j-1}))^2 + \beta})^{-1},$$

$$D_3(u^{[k;\nu]})_{i,j} = (\sqrt{(((u^{[k]})_{i+1,j} - (u^{[\nu]})_{i,j})/h)^2 + (((u^{[k]})_{i,j+1} - (u^{[\nu]})_{i,j})/h)^2 + \beta})^{-1},$$

Finally our proposed method for performing one GS iteration with $\omega \neq 1$ (SOR iteration) can be summarized as follows:

Algorithm 4.2 Our Proposed LFP method

Denote by

v	the restored image
z	the noisy image
g	the RHS (right-hand side) term of (14)
α_1	the weighted parameter of the ROF fitting term in (9)
α_2	the weighted parameter of the JY fitting term in (9)
ω	relaxation parameter
MAX_{FP}	the maximum number of FP iterations

$$[v] \leftarrow LFP\,(v, z, g, \alpha_1, \alpha_2, \omega, MAX_{FP})$$

```
– for $i = 1 : n$ do
    – for $j = 1 : n$ do
        – Set $(v^{[\nu=0]})_{i,j} = (v)_{i,j}$
        – for $\nu = 0 : MAX_{FP}$ do
                – Compute $(\mathbf{G}[v^{[\nu]}])_{i,j}$ using (27)
                – Compute $(\overline{\Sigma}^{[\nu]})_{i,j}(v^{[\nu]})_{i,j}$ using (28)
                – Compute $(\Sigma^{[\nu]})_{i,j}$ using (29)
                – Compute $(\overline{v})_{i,j}$ using (26)
                – Set $(v^{[\nu+1]})_{i,j} = (\overline{v})_{i,j}$
        – end
        – Use $(v^{[\nu=0]})_{i,j}$, $(\overline{v})_{i,j}$ and (25) to compute $(v)_{i,j}$
    – end
– end
```

We have so far presented four numerical methods for solving (12). Obviously the explicit method in (16) is less efficient than the semi-implicit one in (19) because the time-step τ is required to be very small for stability reasons. From several tests on both synthetic and realistic images, we found first that both GFP and LFP methods are much faster than those of the semi-implicit time marching method in fulfilling the necessary condition for being a minimizer of the variational problem represented by (9), i.e. in achieving convergence, because the linear system has to be solved many times with changing the RHS term of in (19). We also found that there is no significant difference in their convergence behavior between GFP and LFP methods when each of them has been used as a stand-alone solver for solving (14).

Although either can be recommended as a unilevel method, it is not efficient as a MG method. Our next task is to select a suitable smoother from both methods in designing a convergent MG method for solving (14). The following section will be devoted to this topic.

5 A nonlinear multigrid method

MG techniques [9, 34, 51, 58, 59] have been proved to be very useful in the context of variational image processing for solving large systems of linear or nonlinear equations arising from high resolution images in real-life applications; see e.g. [6, 7, 10–13, 15, 23–26, 30, 47, 48, 53]. The basic idea of a MG method is to smooth high frequency components of the error of the solution on a fine grid by performing a few steps with a smoother (an iterative relaxation technique) such that a smooth error term can be well represented and approximated on a coarser grid. After a residual equation has been solved on the coarse grid, a coarse-grid correction is interpolated back to the fine grid and used to correct the fine grid approximation. Finally, the smoother is performed again in order to remove some new high frequency components of the error introduced by the interpolation. This is known as a two-grid cycle, and with recursive application it can be extended to a MG method.

As is well known, a working MG has three main components: (i) Smoothing via an iterative method; (ii) Restriction from a fine grid to a coarse grid; (iii) Interpolation from a coarse grid to a fine one. On the coarsest grid, an effective unilevel solver is used for accurate solution; here we shall use the semi-implicit method represented by (19). Without reducing the importance of the restriction and interpolation operators, the efficiency of every MG method strongly relies on the efficiency of the relaxation method, also known as *smoother*, used at each level in reducing (or smoothing) the high frequency components of the error. We shall first use the so-called *local Fourier analysis* (LFA) to decide which method (GFP or LFP) is better suited for our purpose before presenting an overall NMG algorithm. As it turns out, the LFP method is indeed the better method for being an effective smoother.

5.1 *Local Fourier analysis of GFP and LFP smoothers*

LFA is a valuable tool to analyze the smoothing properties of smoothers used in MG methods. Although LFA was originally developed for discrete linear operators with constant coefficients on infinite grids, it can also be applied to more general nonlinear equations with varying coefficients such as the discrete versions of (12). To this end, first an infinite grid is assumed to eliminate the effect of boundary conditions and second it is also assumed that the discrete nonlinear operator can be linearized (by freezing coefficients) and replaced locally by a new operator with constant coefficients [51]. This approach has proved to be very useful in the understanding of MG methods when solving nonlinear problems; see for instance [6, 7, 10–13, 15, 23–26, 32, 33, 35, 42, 49] for interesting examples and discussions.

For linear problems, iterative methods such as damped Jacobi or Gauss-Seidel (GS) methods are usually enough to rapidly reduce high frequencies of the underlying error. However for nonlinear problems, *non-standard smoothers* are often required and their efficiency in *smoothing* is the decisive factor in determining whether a given MG is convergent or not. For nonlinear and anisotropic problems like (12), developing such an effective smoother is by no means a trivial task. A quantitative measure of the smoothing efficiency for a given algorithm is the *smoothing factor* denoted by μ from a LFA and numerically computed for test problems, which is defined as the worst asymptotic error reduction, by performing one smoother step, of all high-frequency error components [51, 59].

Below we shall compute the smoothing factor of GFP and LFP methods (as our GFP and LFP smoothers shortly) applied with the discrete linear equation $\overline{\mathbf{N}}[\overline{u}]u = \overline{\mathbf{G}}[\overline{u}]$. Here u and $\overline{u}$ denote the exact solution and the current approximation and

$$\overline{\mathbf{N}}[\overline{u}] = -\nabla \cdot (\tfrac{\nabla}{|\nabla \overline{u}|_\beta}) + \alpha_1 \mathbf{I}$$

and

$$\overline{\mathbf{G}}[\overline{u}] = g + \alpha_1 z + \alpha_2 (z e^{-\overline{u}} - 1)$$

the resulting discrete (linear) operators obtained from the linearization at $\overline{u}$ by freezing coefficients in (14) at some FP step over the infinite grid

$$\Omega^\infty = \{\mathbf{x} \in \Omega | \mathbf{x} = (x_i, y_j)^\top = (\tfrac{(2i-1)h}{2}, \tfrac{(2j-1)h}{2})^\top,\ i, j \in \mathbb{Z}^2\}.$$

Note that the symbols h and $(\cdot,\cdot)^h_{i,j}$ will be dropped for simplicity.

Let $\boldsymbol{\varphi}(\boldsymbol{\theta}, \mathbf{x}) = \exp(\mathbf{i}\boldsymbol{\theta}\mathbf{x}/h)$ be grid functions $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top \in \boldsymbol{\Theta} = (-\pi, \pi]^2$, $\mathbf{x} \in \Omega^\infty$, and $\mathbf{i} = \sqrt{-1}$. It is important to remark that due to the locality nature of LFA, our analysis applies to each grid point separately, i.e. μ is matrix with its (i, j) entry representing the smoothing factor for grid point $\xi = (i, j)$. Hence we define $\mu_{\text{loc}} = \mu(\xi)$ as the local smoothing factor and $\bar{\mu}_{\text{loc}}$ as the worst possible value of μ_{loc} over Ω^∞. Thus for GFP and LFP methods, we have

$$\bar{\mu}_{\text{loc}} = \max_{\xi \in \Omega_h^\infty} \mu_{\text{loc}}.$$

To determine μ_{loc} we consider the local discrete system $\overline{\mathbf{N}}(\xi)u(\xi) = \overline{\mathbf{G}}(\xi)$ centered and defined only within a small neighborhood of ξ. By using the splitting $\overline{\mathbf{N}}(\xi) = \overline{\mathbf{N}}^+(\xi) + \overline{\mathbf{N}}^-(\xi)$, it is possible to write the local inner iterations of GFP and LFP methods with $\omega = 1$ as

$$\overline{\mathbf{N}}^+(\xi)\overline{u}_{new}(\xi) + \overline{\mathbf{N}}^-(\xi)\overline{u}_{old}(\xi) = \overline{\mathbf{G}}(\xi) \tag{30}$$

where $\overline{u}_{old}(\xi)$ and $\overline{u}_{new}(\xi)$ stand for the approximations to $\overline{u}(\xi)$ before and after the inner smoothing step, respectively. Here

$$\overline{\mathbf{N}}^{+}(\xi) = \begin{bmatrix} 0 & 0 & 0 \\ -D_2(\overline{u}(\xi)) & \Sigma(\xi) & 0 \\ 0 & -D_1(\overline{u}(\xi)) & 0 \end{bmatrix} \text{ and } \overline{\mathbf{N}}^{-}(\xi) = \begin{bmatrix} 0 & -D_3(\overline{u}(\xi)) & 0 \\ 0 & 0 & -D_3(\overline{u}(\xi)) \\ 0 & 0 & 0 \end{bmatrix}.$$

By subtracting (30) from $\overline{\mathbf{N}}(\xi)u(\xi) = \overline{\mathbf{G}}(\xi)$ and defining $\overline{e}_{new}(\xi) = u(\xi) - \overline{u}_{new}(\xi)$ and $\overline{e}_{old}(\xi) = u(\xi) - \overline{u}_{old}(\xi)$ we obtain the local system of error equations

$$\overline{\mathbf{N}}^{+}(\xi)\overline{e}_{new}(\xi) + \overline{\mathbf{N}}^{-}(\xi)\overline{e}_{old}(\xi) = 0 \tag{31}$$

or

$$\overline{e}_{new}(\xi) = \mathbf{S}(\xi)\overline{e}_{old}(\xi) \tag{32}$$

where $\mathbf{S}(\xi) = -[\overline{\mathbf{N}}^{+}(\xi)]^{-1}[\overline{\mathbf{N}}^{-}(\xi)]$ is the amplification factor. The effect of $\mathbf{S}(\xi)$ on the grid functions $\boldsymbol{\varphi}(\boldsymbol{\theta}, \mathbf{x})$ within $\boldsymbol{\Theta}_{\text{high}} = \boldsymbol{\Theta}\backslash[-\pi/2, \pi/2]^2$ will determine the smoothing properties of GFP and LFP methods. Thus, $\overline{\mathbf{N}}^{+}(\xi, \boldsymbol{\theta})$ and $\overline{\mathbf{N}}^{-}(\xi, \boldsymbol{\theta})$ are defined by

$$\overline{\mathbf{N}}^{+}(\xi, \boldsymbol{\theta}) = \Sigma(\xi) - D_1(\xi)\exp(-\mathbf{i}\theta_1) - D_2(\xi)\exp(-\mathbf{i}\theta_2)$$

and

$$\overline{\mathbf{N}}^{-}(\xi, \boldsymbol{\theta}) = -D_3(\xi)(\exp(\mathbf{i}\theta_1) + \exp(\mathbf{i}\theta_2)).$$

and the local smoothing factor is

$$\mu_{\text{loc}} = \sup\{|\rho(\mathbf{S}(\xi, \boldsymbol{\theta}))| : \boldsymbol{\theta} \in \boldsymbol{\Theta}_{\text{high}}\} \tag{33}$$

where ρ indicates the spectral radius of

$$\mathbf{S}(\xi, \boldsymbol{\theta}) = -[\overline{\mathbf{N}}^{+}(\xi, \boldsymbol{\theta})]^{-1}[\overline{\mathbf{N}}^{-}(\xi, \boldsymbol{\theta})]. \tag{34}$$

Here we would say a FP-type smoother is effective if $\mu_{\text{loc}} \ll 1$.

We can analyze the smoothing factor for $1 \neq \omega \in (0, 2)$ in (21) and (25) by the LFA in the similar way to the case $\omega = 1$. Here

$$\overline{\mathbf{N}}^{+}(\xi, \boldsymbol{\theta}) \quad \text{and} \quad \overline{\mathbf{N}}^{-}(\xi, \boldsymbol{\theta})$$

are given respectively by

$$\overline{\mathbf{N}}^{+}(\xi, \boldsymbol{\theta}) = \Sigma(\xi) - \omega D_1(\xi)\exp(-\mathbf{i}\theta_1) - \omega D_2(\xi)\exp(-\mathbf{i}\theta_2) \tag{35}$$

and

$$\overline{\mathbf{N}}^{-}(\xi, \boldsymbol{\theta}) = (1-\omega)\Sigma(\xi) - \omega D_3(\xi)(\exp(\mathbf{i}\theta_1) + \exp(\mathbf{i}\theta_2)). \tag{36}$$

The effectiveness of the above two smoothers (i.e. GFP and LFP methods) is now tested by computing their smoothing factors for the particular examples as shown in Figure 1 (a) and (d). Table 1 summarizes the smoothing factors by the GFP and LFP smoothers. Clearly, the LFP smoother is much better than the GFP smoother.

Smoothers	Example 1	Example 2
GFP $(\omega, MAX_{SOR}) = (1.5, 2)$	0.7837	0.5824
LGP $(\omega, MAX_{FP}) = (1.6, 2)$	0.8258	0.6052

Table 1. Smoothing factors $\bar{\mu}_{\text{loc}}$ by the GFP and LFP smoothers after 5 relaxation steps with $h = 1/128$ and $\beta = 10^{-5}$ for Example 1 and Example 2 as shown in Figure 1 (a) and (d), respectively. Note that both images were corrupted by the combination of Gaussian and Gamma noise by the parameter $(k_0, k_1) = (50, 0.25)$. Clearly, the GFP smoother performs much better than the LFP smoother.

5.2 *A nonlinear multigrid algorithm*

Full approximation scheme based nonlinear multigrid (FAS-NMG) method has become an efficient approach for solving nonlinear problems, in particular image processing applications. Here we have to solve the nonlinear PDE (14) i.e.

$$-\mathcal{K}^{\boldsymbol{h}}\left(u^{\boldsymbol{h}}\right)_{i,j} + \alpha_1(\left(u^{\boldsymbol{h}}\right)_{i,j} - \left(z^{\boldsymbol{h}}\right)_{i,j}) + \alpha_2(1 - \left(z^{\boldsymbol{h}}\right)_{i,j} e^{-(u^{\boldsymbol{h}})_{i,j}}) = \left(g^{\boldsymbol{h}}\right)_{i,j}$$

which is again denoted (for the purpose of applying MG) by

$$\mathcal{N}^h(u^h) = g^h$$

where the nonlinear partial differential operator $\mathcal{N}^h$ is given by the left-hand side of (14) and $g^h = 0$ on the finest grid.

Let $\overline{u}^h$ be the current approximation of u^h after a few smoothing iterations in a **pre-smoothing step** on a fine-grid problem where we denote by u^h the exact solution of (14). Then, the algebraic error $\boldsymbol{e}^h$ of the solution is given by $e^h = u^h - \overline{u}^h$. The residual equation system is given by

$$\mathcal{N}^h(\overline{u}^h + e^h) - \mathcal{N}^h(\overline{u}^h) = g^h - \mathcal{N}^h(\overline{u}^h) = r^h.$$

In order to correct the approximated solution $\overline{u}^h$ on the fine grid, one needs to compute the error e^h. However, the computation of e^h is prohibitively expensive and cannot be computed directly on the fine grid. Since high frequency components of the error in the pre-smoothing step have already been removed by the smoother, we can transfer the following nonlinear system to the coarse grid as follows:

$$\underbrace{\mathcal{N}^h(\overline{u}^h + e^h)}_{\mathcal{N}^h(u^h)} = \underbrace{r^h + \mathcal{N}^h(\overline{u}^h)}_{g^h} \qquad \to \qquad \underbrace{\mathcal{N}^H(\overline{u}^H + e^H)}_{\mathcal{N}^H(u^H)} = \underbrace{r^H_{\hat{l}} + \mathcal{N}^H(\overline{u}^H)}_{g^H} \tag{37}$$

where $H = 2h$ is the new cell size $H \times H$ and $g^H \neq 0$ on the coarse grid. After the nonlinear residual equation on the coarse grid (37) has been solved with a method of our choice, the coarse-grid correction $e^H = u^H - \overline{u}^H$ is then interpolated back to the fine grid e^h that can now be used for updating the approximated solution $\overline{u}^h$ of the original system on the fine grid $\overline{u}^h_{new} = \overline{u}^h + e^h$ (**coarse-grid correction step**). The last step for a FAS-NMG method is to perform the smoother again to remove high frequency parts of the interpolated error (**post-smoothing step**).

We now define our MG components as follows. The GFP or LFP method represented in Section 4.3 is applied as the MG smoother. Standard coarsening is used in computing the coarse-grid domain Ω_H by doubling the grid size in each space direction, i.e. $h \to 2h = H$. For intergrid transfer operators between

Ω_h and Ω_H, the averaging and bi-linear interpolation techniques are used for the restriction and interpolation operators denoted respectively by I_h^H and I_H^h; see the details in [9,34,51,58,59]. In order to compute the coarse-grid operator of $\mathcal{N}^h(u^h)$ given by the left hand side of (14), a so-called *discretization coarse grid approximation* (DCA) is performed [9,51,59]. The idea is to rediscretize the Euler-Lagrange system directly.

Finally, the pseudo-code implementation of our FAS-NMG method can be summarized in the following algorithm:

Algorithm 5.1 FAS-NMG Algorithm

Denote the FAS-NMG parameters as follows:

v^h	the restored image
z^h	the noisy image
g^h	the RHS (right-hand side) term of (14)
α_1	the weighted parameter of the ROF fitting term in (9)
α_2	the weighted parameter of the JY fitting term in (9)
ω	relaxation parameter
MAX_S	the maximum number of iterations using by a smoother
ν_1	pre-smoothing steps on each level
ν_2	post-smoothing steps on each level
μ	the number of MG cycles on each level ($\mu = 1$ for V$-$cycling and $\mu = 2$ for W$-$cycling) [Here we present the V$-$cycle with $\mu = 1$]
$\overrightarrow{\varepsilon}$	the tolerance

$$v^h \leftarrow FASNMG\left(v^h, z^h, g^h, \alpha_1, \alpha_2, \omega, MAX_S, \nu_1, \nu_2, \mu, \overrightarrow{\varepsilon}\right)$$

```
• Select and initial guess solution $\widetilde{v}^h_{initial}$ on the finest grid
• Set $K = 0$, $(v^h)^{[K]} = \widetilde{v}^h_{initial}$, $\widetilde{\varepsilon}_2 = \varepsilon_2 + 1$, $\widetilde{\varepsilon}_3 = \varepsilon_3 + 1$, and $\widetilde{\varepsilon}_4 = \varepsilon_4 + 1$
• While ($K < \varepsilon_1$ AND $\widetilde{\varepsilon}_2 \geq \varepsilon_2$ AND $\widetilde{\varepsilon}_3 \geq \varepsilon_3$ AND $\widetilde{\varepsilon}_4 \geq \varepsilon_4$)
        ▶ $(v^h)^{[K+1]} \leftarrow FASCYC((v^h)^{[K]}, z^h, g^h, \alpha_1, \alpha_2, \omega, MAX_S, \nu_1, \nu_2, \mu)$
        ▶ $\widetilde{\varepsilon}_2 = ||g^h - \mathcal{N}^h((v^h)^{[K+1]})||_2 / ||g^h - \mathcal{N}^h(\widetilde{v}^h_{initial})||_2$
        ▶ $\widetilde{\varepsilon}_3 = J^h((v^h)^{[K+1]}) / J^h(\widetilde{v}^h_{initial})$,
          [$J^h(\cdot)$ denote the discrete version of (9)]
        ▶ $\widetilde{\varepsilon}_4 = |J^h((v^h)^{[K+1]}) - J^h((v^h)^{[K]})|$
        ▶ $K = K + 1$
• end
```

where

$$\left[v^h\right] \leftarrow FASCYC(v^h, z^h, g^h, \alpha_1, \alpha_2, \omega, MAX_S, \nu_1, \nu_2, \mu)$$

```
• If $\Omega_h = coarset\ grid$ ($|\Omega_h| = 4 \times 4$), solve (14) using (19)
  and then stop. Else continue with the following steps.
• Pre-smoothing:
    For $k = 1$ to $\nu_1$, $[v^h] \leftarrow Smoother(v^h, z^h, g^h, \alpha_1, \alpha_2, \omega, MAX_S)$
• Restriction to the coarse grid:
    $v^H \leftarrow I_h^H v^h$, $z^H \leftarrow I_h^H z^h$
• Set the initial solution for the coarse-grid problem:
    $\widetilde{u}^H \leftarrow v^H$
• Compute the new right-hand side for the coarse-grid problem:
    $g^H \leftarrow I_h^H (g^h - \mathcal{N}^h(v^h)) + \mathcal{N}^H(v^H)$
• Implement the FAS-NMG method on the coarse-grid problem:
    For $k = 1$ to $\mu$, $[v^H] \leftarrow FASCYC(v^H, z^H, g^H, \alpha_1, \alpha_2, \omega, MAX_S, \nu_1, \nu_2, \mu)$
• Add the coarse-grid corrections:
    $v^h \leftarrow v^h + I_H^h (v^h - \widetilde{u}^H)$
• Post-smoothing:
    For $k = 1$ to $\nu_2$, $[v^h] \leftarrow Smoother(v^h, z^h, g^h, \alpha_1, \alpha_2, \omega, MAX_S)$
```

For practical applications our FAS-NMG approach is stopped if the maximum number of V$-$ or W$-$cycles ε_1 is reached (usually $\varepsilon_1 = 10$), the relative residual obtained from the Euler-Lagrange equations (14) is smaller than a small number $\varepsilon_2 > 0$ (typically $\varepsilon_2 = 10^{-4}$), the relative reduction of J is smaller than some $\varepsilon_3 > 0$ (typically $\varepsilon_3 = 10^{-2}$), or the change in two consecutive steps of J is smaller than a small number $\varepsilon_4 > 0$ (typically $\varepsilon_4 = 10^{-6}$).

6 Numerical Results

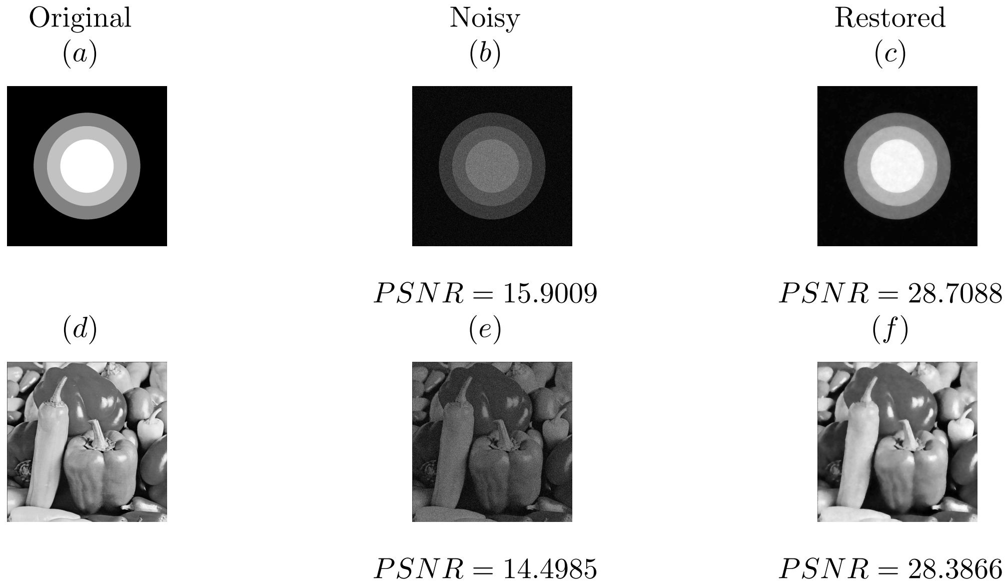

Figure 1. The set of two restoration problems with the image size 1024 × 1024. Left to right: original, noisy, and restored image. Top to bottom: Example 1 (a synthetic image) and Example 2 (a real-life image). Clearly, our proposed model provides good quality of the restored images.

In this section, we present some numerical experiments to

(i) compare the modeling results of the proposed restoration method with the recent state-of-the-art method introduced by Hirakawa and Parks [36]. We chose to focus on their work for three reasons: (a) their noise model is a special case of (8) when $\eta_1 = \eta_2 = \eta$ represents white Gaussian noise with variance one; (b) it reports quantitative results; (c) numerical tests reported in this previous work show that many existing methods compared with their work are less efficient.

(ii) illustrate the robustness of our FAS-NMG method with regard to parameter changes.

We note first that all numerical algorithms are implemented in MATLAB 2010a on 64-bit Fedora 14 and will be stopped when the relative residuals are less than $\varepsilon_2 = 10^{-8}$ for convergence tests. Second, all the tests were carried out on Dell Precision T7500 Workstation with Six-Core Intel Xeon Processor and 24Gb of RAM.

6.1 *Comparison with other image restoration methods in [36]*

Image	(k_0, k_1)	PSNR [36]	PSNR (**Proposed**)
Boats	$(10, 0)$	33.44	**35.56**
	$(25, 0)$	29.16	**30.61**
	$(25, 0.1)$	27.28	**28.65**
	$(25, 0.2)$	25.94	**26.84**
	$(50, 0)$	25.82	**27.18**
Peppers	$(10, 0)$	33.73	**34.13**
	$(25, 0)$	29.32	**30.18**
	$(25, 0.1)$	27.47	**28.39**
	$(25, 0.2)$	26.05	**26.80**
	$(50, 0)$	25.82	**26.67**

Table 2. Restoration methods evaluated using PSNR. Images with the size 512 × 512 corrupted by the combinations of zero-mean Gaussian noise generated by different values of (k_0, k_1). Recall that the third and fourth columns show the PSNR results reported in [36] and obtained from the proposed restoration method, respectively.

In table 2 we list the details of ten experimental setups considered: the images; the noise levels determined by the different values of (k_0, k_1); the PSNR results reported in [36]; the PSNR values obtained from the proposed restoration method. As shown in the third and fourth columns of Table 2, the proposed restoration method performs a clear improvement over the existing method in [36] when images corrupted by an increased level of noise. We note that many existing methods cited and compared in [36] are less efficient than the recent state-of-the-art method proposed .

6.2 *Tests of the proposed FAS-NMG algorithm*

In Section 5 we have used the LFA to inform our theoretical choice of suitable smoother for our new FAS-NMG Algorithm 5.1. Here by experiments, we hope to first verify the reliability of this choice and then to further test the convergence issues of it with regard to parameters h, $\alpha^* = \alpha_1/\alpha_2$, β, and (k_0, k_1). In all tests, η was selected to be white Gaussian noise with variance one, while ζ was adopted to be Gamma noise of mean one.

6.2.1 ***Comparison of GFP and LFP smoothers and h−independent convergence tests.*** As is well known, one of the key properties of MG techniques is that their convergence does not depend on the number of grid points. Thus, in this test we designed our experiments not only to investigate this property with Algorithm 4.1 (GFP smoother) and Algorithm 4.2 (LFP smoother), but also to backup our theoretical results by LFA in Section 5. The number of MG steps (V-cycles), denoted by "M", used to drop the relative residual below $\varepsilon_2 = 10^{-8}$, the PSNR values, and the WUs are given in Table 3 with different sizes of grid points. Recall that ν_1, ν_2, ω, MAX_{FP}, MAX_{SOR}, PSNR, and WUs denote the number of presmoothing and postsmoothing, the relaxation parameter, the maximum iteration of FP and SOR methods, the peak signal-to-noise ratio, and the work units, respectively. Note that WU is used to get a measure of speed without using the machine-dependent CPUs. We define a WU used in measuring computational work as the work of performing a smoother or relaxation step on the finest grid. For the GFP smoother, it is defined by

$$1\text{ WU} = (37 + 21(MAX_{SOR}))N$$

where $N = n^2$ be the number of grid points, the number 37 is estimated from discretizating and computing all nonlinear coefficients and the number 21 comes from updating for each grid point by the SOR method. Similarly, a WU of the LFP smoother is given by

$$1\text{ WU} = 58MAX_{FP}N$$

where the number 58 is the cost of discretizating and computing all nonlinear coefficients and updating for each grid point by the SOR method. Therefore, the total cost of one V-cycle used L coarse grids and the GFP smoother can be estimated as follows:

$$\begin{aligned}\text{V-cycle cost} &= (\nu_1 + \nu_2)(X0 + X1(MAX_{SOR}))N\sum_{k=0}^{L}(1/4)^k \\ &< \frac{4}{3}(\nu_1 + \nu_2)\text{ WUs.}\end{aligned}$$

Here the V-cycle cost used the LFP smoother can be computed in a similar manner. Note that we have ignored the cost of interpolation and restriction procedures as well as the cost of residual correction procedure because they are relatively small compared with smoothing procedures.

As expected from the LFA results in Section 5, the numerical results shown in Table 3 clearly confirm that the LFP smoother is better than the GFP smoother in terms of convergence. However it is slightly more expensive than the GFP smoother.

6.2.2 ***α^*−dependent tests.*** Next we evaluate how our MG algorithm it affected with varying $\alpha^* = \alpha_1/\alpha_2$. To end this, the MG algorithm based on the LFP smoother was tested on Example 2 (see Figure X(b)) with results shown in Table 4. Here the following parameters are used: $\beta = 10^{-5}$, $h = 1/1024$, $\nu_1 = \nu_2 = 5$, $\omega = 1.5$, $MAX_{FP} = 2$, $(k_0, k_1) = (50, 0.25)$ for all experiments and α^* is varied from 2 to 5000. For this test, large or small α^* is not needed as moderate ones give better results, typically $\alpha^* = 50, 100, 1000$. However, the process to select the optimal value of α^* is a separate but important issue because it is in general unknown a priori and it significantly affects on the qualities of recovered images as well as the MG performance.

	MG with the GFP Smoother (Algorithm 4.1) $\nu_1/\nu_2/\omega/MAX_{SOR}$/M/PSNR/WUs	MG with the LFP Smoother (Algorithm 4.2) $\nu_1/\nu_2/\omega/MAX_{FP}$/M/PSNR/WUs
Example 1 (Ring)	$\alpha_1 = 4, \alpha_2 = 1/8, \beta = 10^{-5}$	
$h = 1/512$	5/5/1.5/2/16/27.92/213.3301	5/5/1.60/2/10/27.78/133.3313
$h = 1/1024$	5/5/1.5/2/15/28.66/199.9992	5/5/1.60/2/10/28.70/133.3328
$h = 1/2048$	5/5/1.5/2/14/29.21/186.6665	5/5/1.60/2/10/29.14/133.3332
$h = 1/4096$	5/5/1.5/2/14/29.46/186.6666	5/5/1.60/2/10/29.41/133.3333
Example 2 (Pepper)	$\alpha_1 = 5, \alpha_2 = 1/10, \beta = 10^{-5}$	
$h = 1/512$	5/5/1.5/2/18/25.60/239.9963	5/5/1.60/2/12/25.74/159.9976
$h = 1/1024$	5/5/1.5/2/17/28.57/226.6658	5/5/1.60/2/11/28.38/146.661
$h = 1/2048$	5/5/1.5/2/16/30.87/213.3331	5/5/1.60/2/10/30.64/133.3332
$h = 1/4096$	5/5/1.5/2/16/32.85/213.3333	5/5/1.60/2/10/32.81/133.3333

Table 3. Results of Algorithm 5.1 with the proposed smoothers for processing Examples 1 − 2 shown respectively in Figure 1 (a) and (d). The symbols 'M', 'PSNR', and 'WUs' mean the number of MG cycles, the peak signal-to-noise ratio, the work units, respectively. Recall that ν_1, ν_2, ω, MAX_{SOR}, MAX_{FP} denote, respectively, the number of pre-smoothing and post-smoothing, the relaxation parameter, the maximum iteration of the SOR and FP methods. Obviously, the LFP smoother is better than the GFP smoother in terms of convergence. However it is slightly more expensive than the GFP smoother; 1 WU of LFP smoother = 1.49(1 WU of GFP smoother).

α_1/α_2	1/(1/2)	1/(1/10)	**5/(1/10)**	**5/(5/100)**	**10/(1/100)**	50/(1/50)	50/(1/100)
α^*	2	10	**50**	**100**	**1000**	2500	5000
M	17	13	**12**	**11**	**10**	4	4
PSNR	5.10	16.94	**27.63**	**28.42**	**28.86**	17.36	13.12

Table 4. Results for α^*-dependent tests of Algorithm 5.1 with the LFP smoother for Example 2 shown in Figure 1 (d). Recall that the symbols 'M' and 'PSNR' mean the number of MG cycles and the peak signal-to-noise ratio, respectively. Clearly, the moderate ratios between α_1 and α_2, i.e. $\alpha^* = 50, 100, 1000$, are recommended.

6.2.3 *$\beta-$dependent tests.*

As is well known, the qualities of restoration results and the performance of the MG techniques in solving the nonlinear systems related to the TV regularization method are affected significantly by the values of β. Here our aim for this test is to see how the proposed variational model and FAS-NMG algorithm are affected with varying the values of β.

To see this, the MG algorithm based on the LFP smoother was tested on Example 2 using the following parameters $h = 1/1024$, $\alpha_1 = 5$, $\alpha_2 = 1/10$, $(k_0, k_1) = (50, 0.25)$, $\nu_1 = \nu_2 = 5$, $\omega = 1.6$, and $MAX_{FP} = 2$ for all experiments and β is varied from 10^{-5} to 1.

As can be seen, Table 5 shows that our MG algorithm converges within a few steps. Theoretically β should be selected to be as small as possible. However our experimental results indicate that vary small β is not necessary and not recommendable. As clearly shown in Table 5, $\beta = 10^{-4}$ or $\beta = 10^{-5}$ is enough to remove this kind of noise with the good PSNR results in a few MG steps.

β	1	10^{-1}	10^{-2}	10^{-3}	$\mathbf{10^{-4}}$	$\mathbf{10^{-5}}$
M	6	6	7	7	**7**	**11**
PSNR	18.55	22.21	25.81	27.62	**27.86**	**27.86**

Table 5. Results for $\beta-$dependent tests of Algorithm 5.1 with the LFP smoother for Example 2 shown in Figure 1 (d). Recall that the symbols 'M' and 'PSNR' mean the number of MG cycles and the peak signal-to-noise ratio, respectively. Clearly, $\beta = 10^{-4}$ or $\beta = 10^{-5}$ is enough to remove this kind of noise with the good PSNR results in a few MG steps.

6.2.4 *Signal−dependent tests.* Table 6 shows the robustness of our restoration method and the proposed MG algorithm for different noise levels. Here we tested with the LFP smoother using the parameters $h = 1/1024$, $\alpha_1 = 8$, $\alpha_2 = 0.02$, $\beta = 10^{-5}$, $\nu_1 = \nu_2 = 5$, $\omega = 1.6$, and $MAX_{FP} = 2$ for all tests. As can be clearly seen, although convergence is slower for noisier images, the PSNR values show that the restored images would come with good quality. Moreover, the number of MG steps does not increase very much.

(k_0, k_1)	$(25, 0.1)$	$(25, 0.2)$	$(50, 0.1)$	$(50, 0.2)$
M	8	9	10	11
PSNR	33.02	32.43	29.47	29.39

Table 6. Results for signal−dependent tests of Algorithm 5.1 with the LFP smoother for Example 2 shown in Figure 1 (d). Recall that the symbols 'M' and 'PSNR' mean the number of MG cycles and the peak signal-to-noise ratio, respectively.

7 Conclusion

In this work we have presented a novel variational model and one-step algorithm for the restoration of images corrupted at one time by independent AN and MN. We have also discussed the existence and uniqueness of the solution for the new model and proposed a very fast and reliable NMG algorithm. Numerical tests confirmed that the new model is able to restore better degraded images than former models and very importantly its numerical realization is very fast making the combination of our algorithm and model suitable to process high-resolution images.

References

[1] R. Acar and C. Vogel. Analysis of bounded variation penalty method. *Inverse Problems*, 10:1217–1229, 1994.

[2] A. Achim, P. Tsakalides, and A. Bezerianos. Image denoising via bayesian wavelet shrinkage based on heavy-tailed modeling. *IEEE Trans. Geosci. Remote. Sensing*, 41(8):1773–1784, 2003.

[3] B. Aiazzi, L. Alparone, and S. Baronti. Multiresolution local statistics speckle filtering based on ratio laplacian pylamid. *IEEE Trans. Geosci. Remote. Sensing*, 36(5):1466–1746, 1998.

[4] G. Aubert and J.-F. Aujol. A variational approach to removing multiplicative noise. *SIAM J. Appl. Math.*, 68(4):925–946, 2008.

[5] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (2nd Edition)*. Springer, 2006.

[6] N. Badshah and K. Chen. Multigrid method for the Chan-Vese model in variational segmentation. *Communications in Computational Physics*, 4(2):294–316, 2008.

[7] N. Badshah and K. Chen. On two multigrid algorithms for modelling variational multiphase image segmentation. *IEEE Transactions on Image Processing*, 18(5):1097–1106, 2009.

[8] P. Bao and L. Zhang. Noise reduction for magnetic resonance images via adaptive multiscale products thresholding. *IEEE Trans. Med. Imaging.*, 22(9):1089–1099, 2003.

[9] W.L. Briggs, V.E. Henson, and S.F. McCormick. *A Multigrid Tutorial (2nd Edition)*. SIAM Publications, Philadelphia, USA, 2000.

[10] C. Brito-Loeza and K. Chen. Multigrid method for a modified curvature driven diffusion model for image inpainting. *JCM*, 26(6):856–875, 2008.

[11] C. Brito-Loeza and K. Chen. Fast numerical algorithms for Euler's Elastica digital inpainting model. *International Journal of Modern Mathematics*, 5(2):157–182, 2010.

[12] C. Brito-Loeza and K. Chen. Multigrid algorithm for high order denoising. *SIAM J.Imaging Sci.*, 3(3):363–389, 2010.

[13] C. Brito-Loeza and K. Chen. On high-order denoising models and fast algorithms for vector-valued images. *IEEE Transactions on Image Processing*, 19(6):1518–1527, 2010.

[14] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vision*, 20:89–97, 2004.

[15] T.F. Chan and K. Chen. On a nonlinear multigrid algorithm with primal relaxation for the image total variation minimization. *Journal of Numerical Algorithms*, 41:387–411, 2006.

[16] T.F. Chan and K. Chen. An optimization-based multilevel algorithm for total variation image denoising. *Multiscale Mod. Simu.*, 5(2):615–645, 2006.

[17] T.F. Chan, K. Chen, and J.L. Carter. Iterative methods for solving the dual formulation arising from image restoration. *Electronic Transactions on Numerical Analysis*, 26:299–311, 2007.

[18] T.F. Chan, K. Chen, and X.-C. Tai. *Nonlinear multilevel scheme for solving the total variation image minimization problem.* in Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems Series: Mathematics and Visualization, edited by X.-C. Tai, K.-A. Lie, T.F. Chan and S. Osher, Springer Verlag, pp. 1-27, 2006.

[19] T.F. Chan, G. H. Golub, and P. Mulet. A nonlinear primal-dual method for total variationbased image restoration. *SIAM J. Sci. Comput.*, 20:1964–1997, 1999.

[20] T.F. Chan and P. Mulet. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM J. Numer. Analysis*, 36(2):354–367, 2007.

[21] T.F. Chan and J.H. Shen. *Image Processing and Analysis - Variational, PDE, Wavelet, and Stochastic Methods.* SIAM Publications, Philadelphia, USA, 2005.

[22] K. Chen and X.-C. Tai. A nonlinear multigrid method for total variation minimization from image restoration. *Journal of Scientific Computing*, 32(2):115–138, 2007.

[23] N. Chumchob and K. Chen. A variational approach for discontinuity-preserving image registration. *East-West Journal of Mathematics*, Special volume 2010:266–282, 2010.

[24] N. Chumchob and K. Chen. An improved variational image registration model and a fast algorithm for its numerical approximation. *Numerical Methods for Partial Differential Equations*, page DOI: 10.1002/num.20710, 2011.

[25] N. Chumchob and K. Chen. A robust multigrid approach for variational image registration models. *Journal of Computational and Applied Mathematics*, 236:653–674, 2011.

[26] N. Chumchob, K. Chen, and C. Brito. A fourth order variational image registration model and its fast multigrid algorithm. *SIAM J. Multiscale Modeling and Simulation*, 9:89–128, 2011.

[27] J. Darbon and M. Sigelle. Exact optimization of discrete constrained total variation minimization problems. *LNCS*, 3322:548–557, 2004.

[28] J. Darbon and M. Sigelle. A fast and exact algorithm for total variation minimization. *LNCS*, 3522:351–359, 2005.

[29] D. L. Donoho and M. Johnstone. Adapting to unknown smoothness via wavelet shrink-age. *Journal of the American Statistical Association*, 90(432):1200–1224, 1995.

[30] C. F.-Schauf, S.Henn, and K. Witsch. Nonlinear multigrid methods for total variation image denoising. *Comput. Visual. Sci.*, 7:199–206, 2004.

[31] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach Intel.*, 6(6):721–741, 1984.

[32] E. Haber, R. Horesh, and J. Modersitzki. Numerical optimization for constrained image registration. *To appear in Numerical Linear Algebra with Applications*, DOI: 10.1002/nla.715, 2010.

[33] E. Haber and J. Modersitzki. A multilevel method for image registration. *SIAM J. Sci. Comput.*, 27(5):1594–1607, 2006.

[34] W. Hackbusch. *Multi-grid Methods and Applications.* Springe-Verlag, Berlin, Heidelberg, New York, 1985.

[35] S. Hamilton, M. Benzi, and E. Haber. New multigrid smoothers for the oseen problem. *To appear in Numerical Linear Algebra with Applications*, 2009.

[36] K. Hirakawa and T. W. Parks. Image denoising using total least squares. *IEEE Trans. Image Process.*, 15(9):2730–2742, 2006.

[37] L.-L. Huang, L. Xiao, and Z.-H. Wei. Multiplicative noise removal via a novel variational model. *EURASIP Journal on Image and Video Processing*, 250768:DOI:10.1155/2010/25076, 2010.

[38] Y. Huang, M. Ng, and Y. Wen. A new total variation method for multiplicative noise removal. *SIAM J. Imaging Sci.*, 2(1):20–40, 2009.

[39] Z. Jin and X. Yang. Analysis of a new variational model for multiplicative noise removal. *J. Math. Anal. Appl.*, 362:415–426, 2010.

[40] Z. Jin and X. Yang. A variational model to remove the multiplicative noise in ultrasound images. *J. Math Imaging Vis*, 362:DOI: 10.1007/s10851–010–0225–3., 2010.

[41] M. Karamam, M. Kutay, and G.Bozdagi. An adaptive speckle suppression filter for medical ultrasould images. *IEEE. Trans. Med. Imaging*, 14(2):283–292, 1995.

[42] H. Köstler, K. Ruhnau, and R. Wienands. Multigrid solution of the optical flow system using a combined diffusion- and curvature-based regularizer. *Numer. Linear Algebra Appl.*, 15:201–218, 2008.

[43] W.C. Rheinboldt. *Methods for Solving Systems of Nonlinear Equations (2nd Edition).* SIAM Publications, Philadelphia, USA, 1998.

[44] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[45] L. I. Rudin, P. L. Lions, and S. Osher. *Multiplicative denoising and deblurring: theory and algorithms, in Geometric Level Set Methods in Imaging, Vision, and Graphics, S. Osher and N.Paragios, Eds., pp. 103-120.* Springer, Berlin, Germany, 2003.

[46] Y. Saad. *Iterative Methods for Sparse Linear Systems (2nd Edition).* SIAM Publications, Philadelphia, USA, 2003.

[47] J. Savage and K. Chen. An improved and accelerated nonlinear multigrid method for total-variation denoising. *International Journal of Computer Mathematics*, 82(8):1001–1015, 2005.

[48] J. Savage and K. Chen. *On multigrids for solving a class of improved total variation based staircasing reduction models.* in Proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems Series: Mathematics and Visualization, edited by X.-C. Tai, K.-A. Lie, T.F. Chan and S. Osher, Springer Verlag, pp. 69-94, 2006.

[49] B. Seynaeve, E. Rosseel, B. Nicolaï, and S. Vandewalle. Fourier mode analysis of multigrid methods for partial differential equations with random coefficients. *Journal of Computational Physics*, 224:132–149, 2007.

[50] J. Shi and S. Osher. A nonlinear inverse scale space method for a convex multiplicative noise model. *SIAM J. Imaging Sci.*, 1(3):294–321, 2008.

[51] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid.* Academic Press, 2001.

[52] N. N. Ponomarenko S. K. Abramov O. Pogrebnyak K. O. Egiazarian V. V. Lukin, D. V. Fevralev and J. T. Astola. Discrete cosine transform-based local adaptive filtering of images corrupted by nonstationary noise. *J. Electron. Imaging*, 19:DOI: 10.1117/1.3421973, 2010.

[53] C.R. Vogel. *A mutigrid method for total variation-based image denoising.* Computation and control IV, 20, Progress in systems and control theory, Birkhauser, 1995.

[54] C.R. Vogel. *Negative results for multilevel preconditioners in image deblurring.* Scale-space theories in computer vision, Springer-Verlag, 1999.

[55] C.R. Vogel and M.E. Oman. Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, 17(1):227–238, 1996.

[56] C.R. Vogel and M.E. Oman. Fast, robust total variation-based reconstruction of noizy, blurred images. *IEEE Trans. Image Proc*, 7:813–824, 1998.

[57] J. Weickert, B.M. ter Haar Romeny, and M. Viergever. Effcient and reliable schemes for nonlinear diffusion filtering. *IEEE Trans. Image Processing*, 7:398–410, 1998.

[58] P. Wesseling. *Multigrid Methods.* Edwards: Philadelphis, PA, U.S.A., 2004.

[59] R. Wienands and W. Joppich. *Practical Fourier Analysis for Multigrid Method.* Chapman & Hall/CRC, U.S.A., 2005.

[60] B. Zalesky. Network flow optimization for restoration of images. *Journal of Applied Mathematics*, 2(4):199–218, 2002.

[61] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, 2010.

[62] L. Zhang, R. Lukac, X. Wu, and D. Zhang. PCA-based spatially adaptive denoising of cfa images for single-sensor digital cameras. *IEEE Trans. Med. Imaging.*, 18(4):797–812, 2009.