A TV-Stokes denoising algorithm

Talal Rahman[1], Xue-Cheng Tai[1], and Stanley Osher[2]

[1] Department of Mathematics, University of Bergen
CIPR, Allégt. 41, 5007 Bergen, Norway
(Email: talal.rahman@mi.uib.no, tai@mi.uib.no)
[2] Department of Mathematics, UCLA, California, USA
(Email: sjo@math.ucla.edu)

Abstract. In this paper, we propose a two-step algorithm for denoising digital images with additive noise. Observing that the isophote directions of an image correspond to an incompressible velocity field, we impose the constraint of zero divergence on the tangential field. Combined with an energy minimization problem corresponding to the smoothing of tangential vectors, this constraint gives rise to a nonlinear Stokes equation where the nonlinearity is in the viscosity function. Once the isophote directions are found, an image is reconstructed that fits those directions by solving another nonlinear partial differential equation. In both steps, we use finite difference schemes to solve. We present several numerical examples to show the effectiveness of our approach.

1 Introduction

A digital image d is a function defined on a two dimensional rectangular domain $\Omega \subset \mathbf{R}^2$ where $d(\mathbf{x})$ represents the gray-level value of the image, associated with the pixel at $\mathbf{x} = (x, y) \in \Omega$. Let d_0 be the observed image (the given data) which contains some additive noise η, in other words,

$$d_0(\mathbf{x}) = d(\mathbf{x}) + \eta(\mathbf{x}), \tag{1}$$

where d is the true image. The problem is then to recover the true image from the given data d_0. This is a typical example of an inverse problem, a solution of which is normally sought through the minimization of an energy functional consisting of a fidelity term and a regularization term (a smoothing term). The classical Tikhonov regularization involving the H^1 seminorm, is quite effective for smooth functions, but behaves poorly when the function $d(\mathbf{x})$ includes discontinuities or steep gradients, like edges and textures. The famous model based on the TV-norm regularization, proposed by Rudin-Osher-Fatemi (ROF) in [14], has proven to be quite effective for such functions, removing noise without causing excessive smoothing of the edges. However, it is well known that the TV-norm regularization suffers from the so-called stair-case effect, which may produce undesirable blocky images. Several methods have been proposed since the ROF model, see for instance in [6, 8, 9, 11–13, 16].

Recently, a two step method has been proposed by Lysaker-Osher-Tai (LOT) in [11] involving a smoothing of the normal vectors $\nabla d_0/|\nabla d_0|$ in the first step, and then finding a surface to fit the smoothed normals in the second step based on ideas from [4, 2, 16]. In this paper we use the same two-step approach, but we modify the first step being motivated by the observation that tangent directions to the isophote lines (lines along which the intensity is constant) correspond to an incompressible velocity field, see [3, 15] where this observation was used to develop effective algorithms for the image inpainting. The aim of this paper is to extend this idea further into developing an effective algorithm for the image denoising. Instead of smoothing the normal field in the first step, we smooth the tangential field imposing the constraint that the field is divergence free (incompressible). This results into an algorithm that generates smooth isophote lines turning noisy images into smooth and visually pleasant denoised images, and preserves the edges quite well.

As the algorithm is still in its early stage of research, so far, we have only been interested in its qualitative nature, and not much in the convergence speed. As a result, we have only been using straight forward explicit schemes for the discrete solution. Search for a faster algorithm constitutes part of our future plans.

The paper is organized as follows. In Section 2, we present our two-step algorithm, and include a brief description of the numerical explicit scheme involved in each step. Numerical experiments showing its performance are presented in Section 3.

2 The Denoising Algorithm

Given an image d, the normal and the tangential vectors of the level curves (or the isophote lines) are given by $\mathbf{n} = \nabla d(\mathbf{x}) = (d_x, d_y)^T$ and $\tau = \nabla^\perp d = (-d_y, d_x)^T$. The vector fields then satisfy the following conditions: $\nabla \cdot \tau = 0$ and $\nabla \times \mathbf{n} = 0$, the first one being called the incompressibility condition in the fluid mechanics, a natural condition to use in our algorithm.

Let the noisy image d_0 be given. We compute $\tau_0 = \nabla^\perp d_0$. The algorithm is then defined in two steps. In the first step, we solve the following minimization problem.

$$\min_{\tau} \int_\Omega |\nabla \tau| \, d\mathbf{x} \; + \; \frac{\delta}{2} \int_\Omega |\tau - \tau_\mathbf{0}|^2 \, d\mathbf{x} \qquad \text{subject to} \qquad \nabla \cdot \tau = 0, \tag{2}$$

where δ is a constant which is used to balance between the smoothing of the tangent field and the fidelity to the original tangent field. The gradient matrix and its norm, of the tangent vector $\tau = (v, u)$, are defined as

$$\nabla \tau = \begin{pmatrix} \nabla v \\ \nabla u \end{pmatrix}, \qquad |\nabla \tau| = \sqrt{v_x^2 + v_y^2 + u_x^2 + u_y^2}, \tag{3}$$

respectively. Once we have the smoothed tangent field, we can get the corresponding normal field $\mathbf{n} = (u, -v)$. In the second step, we reconstruct our image

by fitting it to the normal field through solving the following minimization problem.

$$\min_{d} \int_{\Omega} \left(|\nabla d| - \nabla d \cdot \frac{\mathbf{n}}{|\mathbf{n}|} \right) \, d\mathbf{x} \qquad \text{subject to} \qquad \int_{\Omega} (d - d_0)^2 \, d\mathbf{x} = \sigma^2, \quad (4)$$

where σ^2 is the estimated noise variance. This can be estimated using statistical methods. If the exact noise variance cannot be obtained, then an approximate value may be used. In which case, a larger value would result in over-smoothing and a smaller value would result in under-smoothing.

For the discretization, we use a staggered grid, see [15] for some more details. Each vertex of the rectangular grid corresponds to the position of a pixel or pixel center where the image intensity variable d is defined. Let the horizontal axis and the vertical axis represent the x-axis and the y-axis, respectively. The variables v and u, i.e. the components of the tangential vector τ, corresponding to the $-d_y$ and d_x, are then defined respectively along the vertical and the horizontal edges of the grid. Further, we approximate the derivatives by finite differences using the standard forward/backward difference operators $D_x^{\pm}$ and $D_y^{\pm}$, and the centered difference operators C_x^h and C_y^h respectively in the x and y direction where h correspond to the $h-$spacing.

2.1 Step 1: Tangent Field Smoothing

A method of augmented Lagrangian [7] is used for the solution of (2), where we use a Lagrange multiplier to deal with the constraint $\nabla \cdot \tau = 0$, and include a penalty term associated with the same constraint. The corresponding Lagrange functional takes the following form.

$$\mathcal{L}(\tau, \lambda) = \int_{\Omega} |\nabla \tau| \, d\mathbf{x} + \frac{\delta}{2} \int_{\Omega} |\tau - \tau_{\mathbf{0}}|^2 \, d\mathbf{x} + \int_{\Omega} \lambda \nabla \cdot \tau \, d\mathbf{x} + \frac{r}{2} \int_{\Omega} (\nabla \cdot \tau)^2 \, d\mathbf{x}, \quad (5)$$

where λ is the Lagrange multiplier and r is a penalty parameter. The optimality condition for the saddle point is the following set of Euler-Lagrange equations,

$$-\nabla \cdot \left(\frac{\nabla \tau}{|\nabla \tau|} \right) + \delta(\tau - \tau_0) - \nabla \lambda - r \nabla (\nabla \cdot \tau) = 0 \qquad \text{in } \Omega, \quad (6)$$

$$\nabla \cdot \tau = 0 \qquad \text{in } \Omega, \quad (7)$$

with the following boundary condition,

$$\left(\frac{\nabla \tau}{|\nabla \tau|} + \lambda I \right) \cdot \nu = \mathbf{0} \qquad \text{on } \partial\Omega, \quad (8)$$

where ν is the unit outward normal and I is the identity matrix. For the solution we use the method of gradient-descent requiring to solve the following equation to steady-state.

$$\frac{\partial \tau}{\partial t} - \nabla \cdot \left(\frac{\nabla \tau}{|\nabla \tau|} \right) + \delta(\tau - \tau_0) - \nabla \lambda - r \nabla (\nabla \cdot \tau) = 0 \qquad \text{in } \Omega, \quad (9)$$

$$\frac{\partial \lambda}{\partial t} - \nabla \cdot \tau = 0 \qquad \text{in } \Omega, \quad (10)$$

with (8) being the boundary condition, and t being the artificial time variable.

The discrete approximation of (8)-(10) now follows. Let the Lagrange multiplier λ be defined at the centers of the rectangles of the grid. We first determine the tangential vector τ^0 as $(v^0, u^0)^T = (-D_y^- d_0, D_x^- d_0)^T$, and take h to be equal to one. The values of the variables u, v and λ at step $n+1$ are then calculated from

$$\frac{v^{n+1} - v^n}{\Delta t} = D_x^- \left(\frac{D_x^+ v^n}{T_1^n} \right) + D_y^- \left(\frac{D_y^+ v^n}{T_2^n} \right) - \delta \left(v^n - v_0 \right) + D_x^- \left(\lambda^n + Div(\tau^n) \right), \quad (11)$$

$$\frac{u^{n+1} - u^n}{\Delta t} = D_x^- \left(\frac{D_x^+ u^n}{T_2^n} \right) + D_y^- \left(\frac{D_y^+ u^n}{T_1^n} \right) - \delta \left(u^n - u_0 \right) + D_y^- \left(\lambda^n + Div(\tau^n) \right), \quad (12)$$

$$\frac{\lambda^{n+1} - \lambda^n}{\Delta t} = D_x^+ v^n + D_y^+ u^n, \quad (13)$$

where $Div(\tau^n) = D_x^+ v^n + D_y^+ u^n$ is a discrete divergence operator. For the terms T_1 and T_2, we introduce two average operators A_x and A_y by defining $A_x w = (w(x,y) + w(x+h,y))/2$ and $A_y w = (w(x,y) + w(x,y+h))/2$. Then

$$T_1 = \sqrt{\left(A_x(C_y^h v^n)\right)^2 + \left(D_x^+ v^n\right)^2 + \left(D_y^+ u^n\right)^2 + \left(A_y(C_x^h u^n)\right)^2 + \epsilon}, \quad (14)$$

$$T_2 = \sqrt{\left(A_y(C_x^h v^n)\right)^2 + \left(D_y^+ v^n\right)^2 + \left(D_x^+ u^n\right)^2 + \left(A_x(C_y^h u^n)\right)^2 + \epsilon}. \quad (15)$$

2.2 Step 2: Image reconstruction

Once we have the value of $\tau = (v, u)^T$ from Step 1 of the algorithm, we use them here to reconstruct our image d. Using a Lagrange multiplier μ for the constraint in (4) we get the following Lagrange functional.

$$\mathcal{L}(d, \mu) = \int_\Omega \left(|\nabla d| - \nabla d \cdot \frac{\mathbf{n}}{|\mathbf{n}|} \right) d\mathbf{x} + \mu \left(\int_\Omega (d - d_0)^2 \, d\mathbf{x} - \sigma^2 \right). \quad (16)$$

The corresponding set of Euler-Lagrange equations for the saddle point is

$$-\nabla \cdot \left(\frac{\nabla d}{|\nabla d|} - \frac{\mathbf{n}}{|\mathbf{n}|} \right) + \mu(d - d_0) = 0 \qquad \text{in } \Omega, \quad (17)$$

$$\int_\Omega \left(\frac{d - d_0}{\sigma} \right)^2 d\mathbf{x} = 1, \quad (18)$$

with the Neumann boundary condition

$$\left(\frac{\nabla d}{|\nabla d|} - \frac{\mathbf{n}}{|\mathbf{n}|} \right) \cdot \nu = 0 \qquad \text{on } \partial\Omega. \quad (19)$$

One way to calculate the Lagrange multiplier μ is to make use of the condition (18), see [11] for detail,

$$\mu = -\frac{1}{\sigma^2}\int_\Omega \left(\frac{\nabla d}{|\nabla d|} - \frac{\mathbf{n}}{|\mathbf{n}|}\right)\cdot\nabla(d-d_0)\,d\mathbf{x}. \tag{20}$$

Introducing an artificial time variable t we get the following time dependent problem, which needs to be solved to steady-state,

$$\frac{\partial d}{\partial t} - \nabla\cdot\left(\frac{\nabla d}{|\nabla d|} - \frac{\mathbf{n}}{|\mathbf{n}|}\right) + \mu(d-d_0) = 0 \qquad \text{in } \Omega, \tag{21}$$

with the Neumann boundary condition (19), and μ is given by the equation (20). If we replace the unit vector $\frac{\mathbf{n}}{|\mathbf{n}|}$ with the zero vector $\mathbf{0}$, then the method reduces to the classical TV denoising algorithm of Rudin, Osher and Fatemi [14].

Noting that $\mathbf{n} = (u, -v)$, the discrete formulation of the image reconstruction step takes the following form.

$$\frac{d^{n+1}-d^n}{\Delta t} = D_x^-\left(\frac{D_x^+ d^n}{T_3^n} - n_1\right) + D_y^-\left(\frac{D_y^+ d^n}{T_4^n} - n_2\right) - \mu^n\left(d^n - d_0\right), \tag{22}$$

where μ^n is approximated as

$$\begin{aligned}\mu^n = &-\frac{1}{\sigma^2}\sum\left(\frac{D_x^+ d^n}{T_3^n} - n_1\right) D_x^+\left(d^n - d_0\right)\\ &-\frac{1}{\sigma^2}\sum\left(\frac{D_y^+ d^n}{T_4^n} - n_2\right) D_y^+\left(d^n - d_0,\right)\end{aligned} \tag{23}$$

with T_3 and T_4 being defined as

$$T_3 = \sqrt{\left(D_x^+ d^n\right)^2 + \left(A_x(C_y^h d^n)\right)^2 + \epsilon}, \tag{24}$$

$$T_4 = \sqrt{\left(D_y^+ d^n\right)^2 + \left(A_y(C_x^h d^n)\right)^2 + \epsilon}, \tag{25}$$

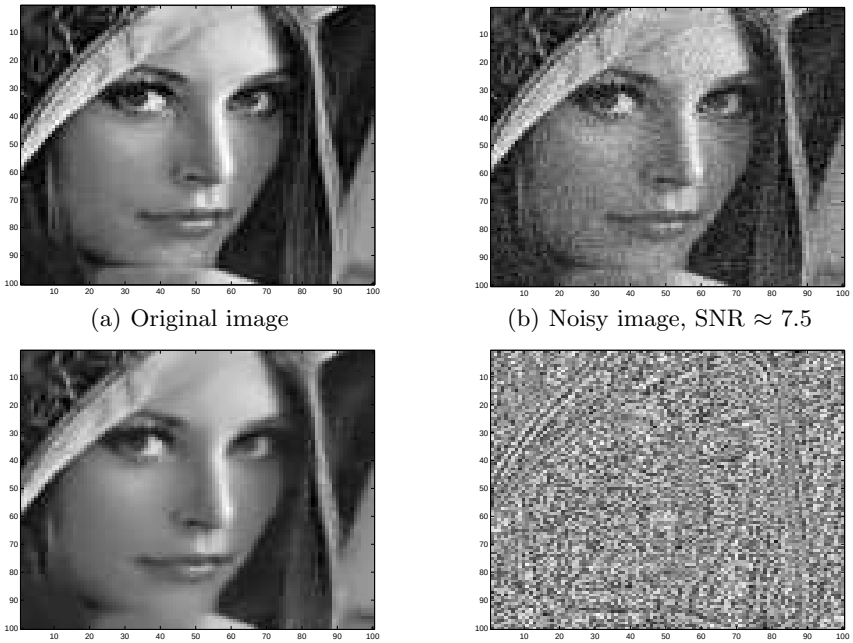
and n_1 and n_2 as

$$n_1 = \frac{u}{\sqrt{u^2 + \left(A_x(A_y v)\right)^2 + \epsilon}}, \qquad n_2 = \frac{-v}{\sqrt{v^2 + \left(A_y(A_x u)\right)^2 + \epsilon}}. \tag{26}$$

3 Numerical Experiments

Several experiments with the proposed two-step algorithm have been performed, we present a few of them in this section.

As in each step of the algorithm the minimal of an energy functional is being sought, it is reasonable to use the energy functional as an objective measure for the stopping criterion in the corresponding step. However, since the minimization problems are subject to constraints, it is not enough just to use the energy



(a) Original image (b) Noisy image, SNR ≈ 7.5

(c) Denoised image (d) Difference image

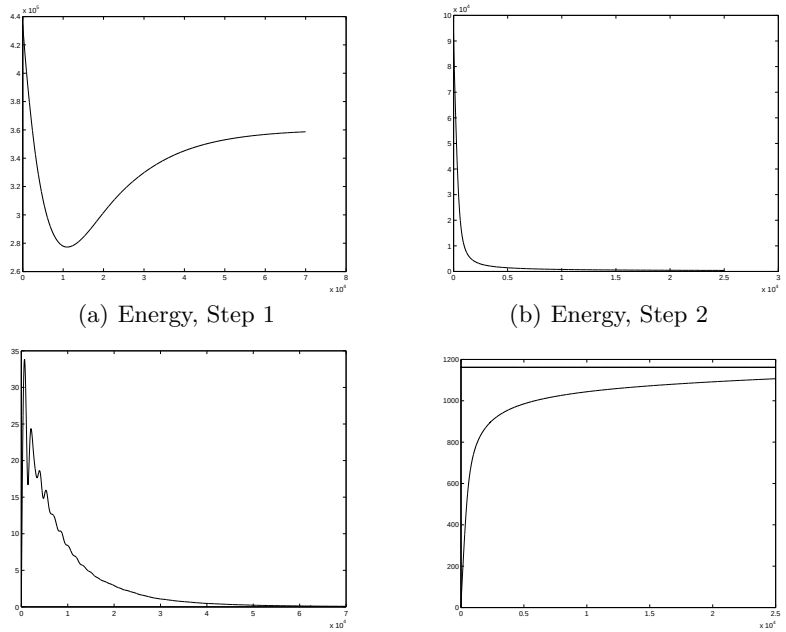
Fig. 1. The Lena image, denoised using the TV-Stokes algorithm.

functionals as stoping criterion. It has been observed through the image reconstruction step that even after the energy has nearly stabilized at some numerical minimum the image continues to improve, and the image becomes visually super when both the energy is minimum and the constraint is satisfied accurate enough. We have therefore included the constraints as additional measures in determining when to terminate the iterations. For the vector smoothing step we use the value of $\left(\int_\Omega |\nabla \cdot \tau|^2 \, d\mathbf{x}\right)^{\frac{1}{2}}$, and for the reconstruction step we compare the value of $\left(\int_\Omega (d - d_0)^2 \, d\mathbf{x}\right)^{\frac{1}{2}}$ with the given noise level σ.

Experiments have shown that the TV-Stokes can very often give smooth images that are visually very pleasant, specially in the smooth areas of the image. At places where the texture changes very rapidly, TV-Stokes seems to smear out the image and thereby loose the fine scale details.

For the presentation of our results, we have chosen three images with gray-level values in the range between 0 (black) and 255 (white). In all our experiments, we expose our image to random noise with zero mean, and apply different denoising algorithms on the noisy image. The value of ϵ is set equal to 10^{-11} in all cases.

The first image is the well known Lena image, cf. Figure 1. The TV-Stokes algorithm is applied first to the noisy image. In Figure 1 we can see the result



(a) Energy, Step 1 (b) Energy, Step 2

(c) Discrete L^2-norm of $\nabla \cdot \tau$ (d) Noise level σ

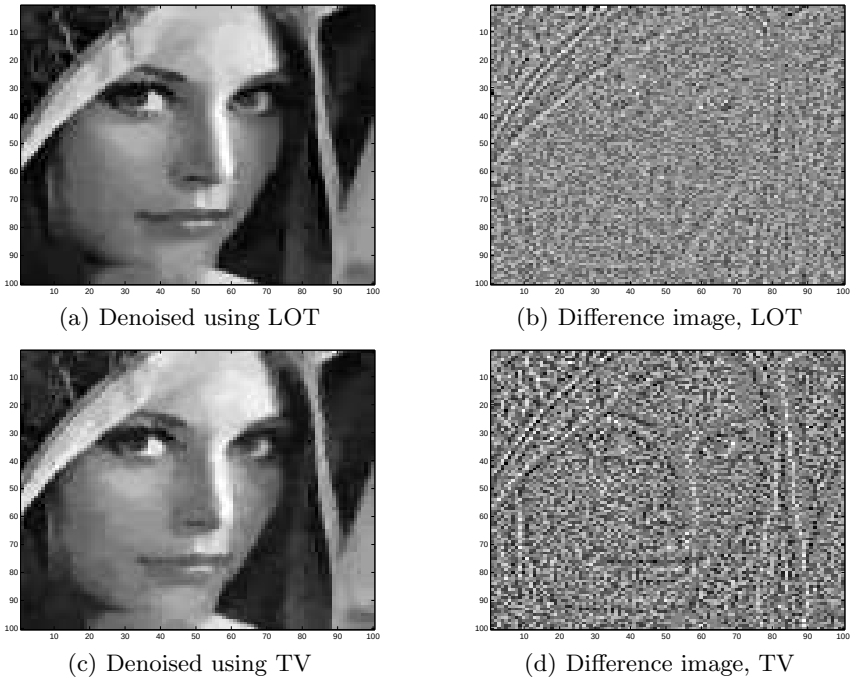
Fig. 2. Plot of energy during tangent field smoothing (Step 1) and during image reconstruction (Step 2), followed by plots of their corresponding constraint measures. The solid line in (d) indicates the true noise level σ.

of one such test where the parameter δ is equal to 0.07. As we can see from the denoised image, it is evident that the TV-Stokes algorithm does a very good job recovering the smooth area and yet keep the edges quite well.

To understand the behavior of the TV-Stokes algorithm as the iteration continues we have plotted the energy and the corresponding constraint measure, as shown in Figure 2, for both the normal smoothing and the image reconstruction step. It is clear from the plots in both cases, that the energies may stabilize long before the constraints are met with some accuracy.

The experiment of Figure 1 and 2 has been performed using fixed time steps Δt equal to 10^{-3} and 5×10^{-3} respectively for the first and second steps of the algorithm. It is well known that with large time step Δt the algorithm may become unstable and not converge to steady state. It is then necessary to choose a reasonably smaller time step. It is usual to choose Δt by trial and error or from experience. The choice of Δt depends on the parameter ϵ, for a large ϵ Δt can be large, but for smaller ϵ it is necessary to use a smaller Δt, which ofcourse slows down the algorithm in reaching the steady state. However, with large ϵ the algorithm will result in an image which may not be sharp, but the image gets sharper as the parameter is reduced. In several occasions, we have exploited

this situation in reducing the number of iterations by first running the algorithm with a large ϵ and a corresponding time step, and then gradually decreasing their values as the iteration continues.



(a) Denoised using LOT (b) Difference image, LOT

(c) Denoised using TV (d) Difference image, TV

Fig. 3. Denoised Lena image using the LOT algorithm of [11] the classical TV algorithm of [14].

For the comparison, we include the results of applying the classical TV denoising scheme [14] and the LOT algorithm of [11] on our noisy image of Figure 1. The denoised images and the corresponding difference images are shown in Figure 3. As seen from the difference images, the LOT algorithm seems to have preserved the edges best, while the proposed algorithm shows a performance which is very close to the LOT algorithm and much better than the TV algorithm in preserving edges. From the recovered images, however, we see that the image created by the TV-Stokes algorithm is the smoothest and visually the most pleasant. Moreover, the stair-case effect of the TV algorithm does not exist in the new algorithm, see Figure 4 for a comparison, where the TV method shows clear evidence of a blocky image.

The next image we consider is a commonly used blocky image on which the TV algorithm is known to perform the best. It is not easy to smooth as well as preserve the edges. For this particular experiment the parameter δ is chosen

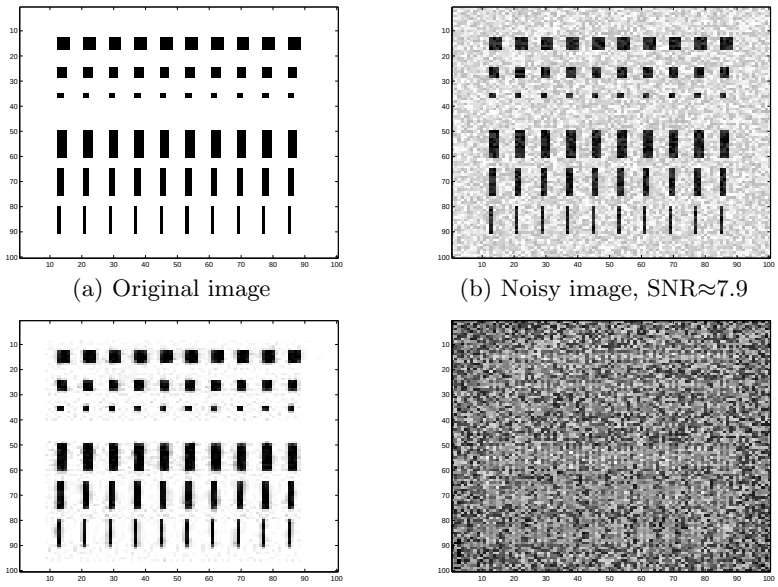
(a) Denoised using TV-Stokes (b) Denoised using TV

Fig. 4. Comparison of the results of the two methods: the TV-Stokes on the left, and the TV on the right.

equal to 0.2, and the time steps for the first and second steps of the algorithm are chosen equal to 5×10^{-3} and 10^{-2}, respectively. The denoised image and the difference image obtained by using the TV-Stokes algorithm are shown in Figure 5, illustrating that the TV-Stokes algorithm has managed to suppress the noise sufficiently well, and at the same time it has maintained the edges.

As the final image we consider the Cameraman image, cf. Figure 6, consisting of a smooth background (the sky), a relatively weak skyline, and very random grass texture. This image is considered to be difficult for most algorithms including the TV algorithm, the LOT algorithm of [11] and the TV-Stokes algorithm. However, as seen from the recovered images in this case, the TV-Stokes algorithm performs better than the LOT algorithm preserving the shapes of objects far away. Moreover, the TV-Stokes algorithm results in a much smoother image compared to the other one, cf. Figure 6. Here, the parameter δ is equal to 0.575, and the time steps are equal to 5×10^{-3} and 10^{-2} respectively for the first and the second steps of the TV-Stokes algorithm.

We close this section with a comment on the choice of the parameter δ. The choice of δ is crucial for the TV-Stokes algorithm to succeed. For δ sufficiently small the algorithm performs normally quite well. The recovered image may however become too smooth causing it to loose edges. The image can in most cases be improved by tuning up the parameter. However, as we gradually increase the parameter δ, the quality of the restored image may decrease. For instance, in case of the Lena image it has been observed that the algorithm restores the image perfectly well for δ around 0.06, but for δ around 0.1 the restored image still contains some noise.



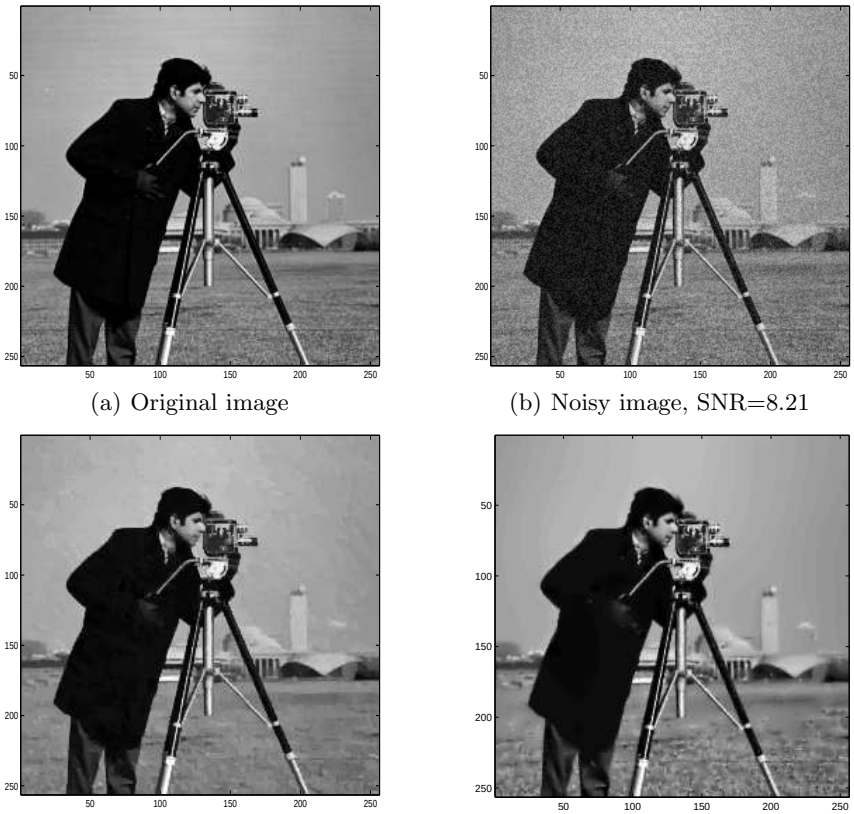
(a) Original image (b) Noisy image, SNR≈7.9

(c) Denoised using TV-Stokes (d) Difference image

Fig. 5. TV-Stokes algorithm on a blocky image.

References

1. Gilles Aubert and Pierre Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, Applied Mathematical Sciences 147, 2002, Springer Verlag, New York.
2. C. Ballaster, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, *Filling in by Joint Interpolation of Vector Fields and Gray Levels*, IEEE Trans. Image Processing, No. 10, 2000, pp. 1200–1211.
3. M. Bertalmio, A. L. Bertozzi, and G. Sapiro, *Navier-Stokes, Fluid Dynamics and Image and Video Inpainting*, In Proc. Conf. Comp. Vision Pattern Rec., 2001, pp. 355–362.
4. P. Burchard, T. Tasdizen, R. Whitaker, and S. Osher, *Geometric Surface Processing via Normal Maps*, Tech. Rep. 02-3, Applied Mathematics, 2002, UCLA.
5. T.F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, 2005, SIAM, Philadelphia.
6. C. Frohn-Schauf, S. Henn, and K. Witsch, *Nonlinear Multigrid Methods for Total Variation Image Denoising*, Comput. Visual. Sci., Vol. 7, 2004, pp. 199–206.
7. R. Glowinski and P. LeTallec, *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM Studies in Applied Mathematics, Vol. 9, 1989, SIAM, Philadelphia.
8. D. Goldfarb and W. Yin, *Second Order Cone Programming Methods for Total Variation-Based image Restoration*, SIAM J. Sci. Comput., Vol. 27, No. 2, 2005, pp. 622–645.



(a) Original image (b) Noisy image, SNR=8.21

(c) Denoised using LOT (d) Denoised using TV-Stokes

Fig. 6. Comparing the TV-Stokes algorithm with the LOT algorithm of [11].

9. S. Kindermann, S. Osher, and J. Xu, *Denoising by BV-duality*, J. Sci. Comput., Vol. 28, Sept. 2006, pp. 414–444.
10. D. Krishnan, P. Lin, and X.C. Tai, *An Efficient Operator-Splitting Method for Noise Removal in Images*, Commun. Comput. Phys., Vol. 1, 2006, pp. 847–858.
11. M. Lysaker, S. Osher, and X.C. Tai, *Noise Removal Using Smoothed Normals and Surface Fitting*, IEEE Trans. Image Processing, Vol. 13, No. 10, October 2004, pp. 1345–1357.
12. S. Osher, A. Sole, and L. Vese, *Image Decomposition and Restoration Using Total Variation Minimization and the H^{-1} norm*, Multiscale Modelling and Simulation, A SIAM Interdisciplinary J., Vol. 1, No. 3, 2003, pp. 1579–1590.
13. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, *An Iterative Regularization Method for Total Variation Based Image Restoration*, Multiscale Modelling and Simulation, Vol. 4, No. 2, 2005, pp. 460–489.
14. L.I. Rudin, S. Osher, and E. Fatemi, *Nonlinear Total Variation Based Noise Removal Algorithms*, Physica D., Vol. 60, 1992, pp. 259–268.

15. X.C. Tai, S. Osher, and R. Holm, *Image Inpainting using TV-Stokes equation*, in: Image Processing based on partial differential equations, 2006, Springer, Heidelberg.
16. L. Vese and S. Osher, *Numerical Methods for P-Harmonic Flows and Applications to Image Processing*, SIAM J. Numer. Anal., Vol. 40, No. 6, December 2002, pp. 2085–2104.