

Noise removal using smoothed normals and surface fitting

Lysaker, Marius; Osher, Stanley; Tai, Xue Cheng

2004

Lysaker, M., Osher, S., & Tai, X. C., (2004). Noise removal using smoothed normals and surface fitting. IEEE Transaction on Image Processing, 13(10), 1345-1357.

<https://hdl.handle.net/10356/91598>

IEEE Transection on image processing @copyright 2004 IEEE. The journal's website is located at http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1331446.

Downloaded on 23 Aug 2022 13:33:41 SGT

Noise Removal Using Smoothed Normals and Surface Fitting

Marius Lysaker, Stanley Osher, and Xue-Cheng Tai

Abstract—In this work, we use partial differential equation techniques to remove noise from digital images. The removal is done in two steps. We first use a total-variation filter to smooth the normal vectors of the level curves of a noise image. After this, we try to find a surface to fit the smoothed normal vectors. For each of these two stages, the problem is reduced to a nonlinear partial differential equation. Finite difference schemes are used to solve these equations. A broad range of numerical examples are given in the paper.

Index Terms—Anisotropic diffusion, image denoising, nonlinear partial differential equations (PDEs), normal processing.

I. INTRODUCTION

A digital image d can contain random noise η superimposed on the pixel intensity value by the formula

$$d_0(x, y) = d(x, y) + \eta(x, y). \quad (1)$$

We would like to recover the true image $d(x, y)$ from its noisy observation $d_0(x, y)$. Noise is recognized as rapidly oscillating signals and can, therefore, be removed by the process of low-pass filtering or smoothing, unfortunately at the expense of some high-frequency information (i.e. edges). Wavelet-based methods, statistical methods, and diffusion filter methods have successfully been used to remove noise from digital images. Wavelet methods exploit the decomposition of the data into the wavelet basis and shrink the wavelet coefficients in order to denoise the images [1]–[3]. However, the space of functions of bounded variation, which allows for edges in our reconstruction, cannot be used easily with wavelet based methods. For statistical methods used in image restoration, we refer to the nonlinear median-type filters [4]–[6]. Median and mean filters replace every pixel of an image with respectively the median and the arithmetic mean of the pixels contained in a window around the pixel. Mean filtering is usually used for suppressing Gaussian noise while median filtering is a powerful tool for removing impulse-like noise. However, in this work,

we are more engrossed by diffusion filters coming from partial differential equations (PDEs). In the end of Section II, we outline some benefits inherited with this approach. Among various PDE techniques proposed in the literature, let us mention [7]–[10]. The TV-norm filter proposed in [7] gives a rigorous mathematical tool to introduce nonlinear diffusion filters and it has been used as a regularization method for many other applications where one needs to identify discontinuous functions. The TV-norm filter preserves edges but has the sometimes undesirable staircase effect, meaning that smooth functions are transformed into piecewise constants. To overcome this problem (but, then, maybe sacrifice the good property of TV norm on the edges), many other nonlinear filters have been suggested in the literature, and, during the last few years, higher order PDEs have been of special interest [11]–[16]. The method we shall use in this paper is somewhat related to the methods using higher order PDEs. We solve two second-order nonlinear PDEs sequentially. If we combine the two equations together, we would need to solve one higher order nonlinear PDE. To be more precise, our method is a two-step method. In the first step, we smooth the normal vectors of the level set curves of the noise image using a TV-norm filter. After the normal vectors are smoothed, we try to find an image that fits the normal vectors and this image is taken as the recovered image for the corrupted observation.

Our method is related to some techniques already in the literature. In Kenney and Langan [17], a method to restore images from modified flow field was proposed. In its simplest form, this process takes a single image, modifies its gradient field, and then constructs a new image from the modified field. They introduce an objective function based on a flow field to implicitly determine the edges of the image.

The idea we use in this paper is more inspired by the work [18]. In that work, they tried to process three dimensional surfaces. The essential idea was to manipulate the normal vectors for a given three dimensional surface and then find a new surface that matches the processed normal vectors in a suitable way. In this work, we are extending the idea of [18] to do image noise removal. Further, we would like to mention that normal processing has also been used in shape from shading reconstruction [19], [20] and in mesh optimization [21], [22].

Another closely related paper is the inpainting approach of [23]. In this paper, they try to minimize an energy functional with respect to two variables: a vector field \vec{n} which represents the direction of the level curves of $d(x, y)$ and the intensity value $d(x, y)$. Note that image denoising is quite different from inpainting. In [23], they clearly point out how shape recovery is achieved by first computing the vector field \vec{n} approximately, and then include this information to the reconstruction process.

Manuscript received June 24, 2003; revised June 24, 2004. This work was supported in part by the Norwegian Research Council under project number 135302/320, in part by IMAR under Contract ICA1-CT-2000-70022 with the European Commission, and in part by the NSF under Grant DMS 0312222. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Fernando M. B. Pereira.

M. Lysaker is with the Department of Scientific Computing, Simula Research Laboratory AS, Norway (e-mail: mariul@simula.no).

X.-C. Tai is with the Institute of Mathematics, Henan University, Kaifeng, 475001, China, and also with the Department of Mathematics, University of Bergen, N-5008, Bergen, Norway (e-mail: tai@mi.uib.no).

S. Osher is with the Department of Mathematics, University of California Los Angeles, CA 90024 USA (e-mail: sjo@math.ucla.edu).

Digital Object Identifier 10.1109/TIP.2004.834662

We also utilized this idea in our algorithm. In fact, the functional for the minimization problem we solve in the second step is identical to one of the terms in the inpainting functional of [23]. Another closely related work is [24]. That paper deals with minimization of constrained functionals, and in particular with p -harmonic maps. For a vector valued function $V = (u, v) : \Omega \rightarrow \mathbb{R}^2 \setminus \vec{0}$, they proposed an elegant way of solving

$$\inf_V \int_{\Omega} \left| \nabla \left(\frac{V}{|V|} \right) \right|^p dx \quad (2)$$

with Dirichlet or Neumann boundary conditions. The tools we shall use to smooth the normal vectors are closely related to problem (2). In our approach, we are essentially taking $V = \nabla d$ and $p = 1$ in (2), and we add a fidelity term to balance the smoothing and the edge preserving [see (6)].

Moreover, the approach we propose here is also strongly motivated by the fourth-order method proposed in [11] and the second-order method [7]. The original TV-norm filter is to obtain a restored image d as a solution of the constrained optimization problem

$$\inf_d \int_{\Omega} |\nabla d| dx \quad \text{subject to} \quad \int_{\Omega} |d - d_0|^2 = \sigma^2 \quad (3)$$

where σ^2 denotes the variance of the noise. In order to overcome the stair-case effects, it was suggested in [11] to replace the cost functional in (3) by

$$\int_{\Omega} (|d_{xx}| + |d_{yy}|) dx \quad \text{or} \quad \int_{\Omega} \sqrt{|d_{xx}|^2 + |d_{xy}|^2 + |d_{yx}|^2 + |d_{yy}|^2} dx. \quad (4)$$

With these minimization functionals, we try to minimize the total variational norm of ∇d instead of d . Rather good numerical performance was obtained in [11]. We could go one step further. Instead of minimizing the TV norm of d or ∇d , we could minimize the TV norm of $\nabla d / |\nabla d|$, i.e.

$$\inf_d \int_{\Omega} \left| \nabla \frac{\nabla d}{|\nabla d|} \right| dx \quad \text{subject to} \quad \int_{\Omega} |d - d_0|^2 = \sigma^2. \quad (5)$$

We know that $\nabla d / |\nabla d|$ is the unit normal vectors for the level curves of d . However, the Euler–Lagrange equation obtained by minimizing the above functional is hard to solve numerically because of the restrictive time step needed. In this paper, we propose to split this into two steps, i.e., we first smooth the unit normal vectors and then find a surface to fit the obtained normal vectors.

For images with discontinuities, the algorithm proposed here is clearly an improved version of the noise removal algorithm of [25, Section 4.2]. In [25], the functions that need to be smoothed are always continuous.

This paper is organized in the following way. In Section II, we introduce the two-step method we use for noise removal. Details are given to show how we obtain and solve the associated nonlinear PDEs coming from the two-step method. Finite

difference approximations and some implementation details are explained in Section III. Numerical results are given in Section IV. In the numerical experiments, we compare our method with some related algorithms in the literature. Finally, we have a Conclusion section followed by an Appendix, where we describe a transformation used in Section II.

II. FLOW FIELD SMOOTHING AND SURFACE FITTING

For a given image d , $\vec{n} = \nabla d / |\nabla d|$ is the unit normal vector of the level curves of d . For the noisy image d_0 , we shall try to smooth the normal vectors $\vec{n}_0 = \nabla d_0 / |\nabla d_0|$. To avoid numerical problems at low gradients, a small constant is added in the calculation of the gradient magnitude, i.e. $|\nabla d|$ is substituted by $\sqrt{|\nabla d|^2 + \epsilon}$, turning the process into a convex one. Because the normal vectors can be discontinuous vector functions we use the TV norm to do the smoothing. Similar to [7], [11], we add a fidelity term to balance the smoothing and edge capturing. To be more precise, we use a two-parameter method, one parameter controls the flow field smoothing and another parameter controls the surface fitting such that large deviation from their initial state will be penalized. The following minimization problem is solved to get a smoothed flow field (i.e., normal vectors):

$$\min_{|\vec{n}|=1} \left\{ \int_{\Omega} |\nabla \vec{n}| dx + \frac{\lambda}{2} \int_{\Omega} |\vec{n} - \vec{n}_0|^2 dx \right\}. \quad (6)$$

Presently, we are investigating several ways to select λ dynamically, but for all simulations given in this paper, the value of λ is fixed to a positive constant. At the end of this section, we give some advice on how to choose this constant.

Once the flow field is calculated, we want to find an image d that matches this field. There are different ways to do this. In this work, we try to find a d that solves the following minimization problem

$$\min_{|d-d_0|^2 dx = \sigma^2} \int_{\Omega} (|\nabla d| - \nabla d \cdot \vec{n}) dx \quad (7)$$

where \vec{n} is the smoothed normals obtained from (6). In the above, σ is the noise level in $L^2(\Omega)$ norm, and we assume it to be approximately known or that it can be estimated [26]–[28]. For example, in [28], they proposed a statistical approach for blind estimation of noise variance. In 98% of the cases they evaluated, the relative estimation error was less than 0.2 with an average error of 0.06. In case the noise level is not known or the above estimation algorithms cannot be applied successfully, we just add a fidelity term to the minimization functional (7) and drop the noise level constraint.

Assume that α is the angle between ∇d and \vec{n} , it is clear that the functional

$$\int_{\Omega} (|\nabla d| - \nabla d \cdot \vec{n}) dx = \int_{\Omega} |\nabla d| (1 - \cos(\alpha)) dx \quad (8)$$

is always nonnegative. Heuristically, \vec{n} is orthogonal to the level lines of d , meaning that when (8) is minimized, the function $d : \Omega \rightarrow \mathbb{R}$ is constructed such that d is constant along the integral curves of \vec{n}^\perp .

We shall use a Lagrange multiplier to deal with the noise constraint $\int_{\Omega} |d - d_0|^2 dx = \sigma^2$. The Lagrange functional is defined as

$$L(d, \mu) = \int_{\Omega} (|\nabla d| - \nabla d \cdot \vec{n}) dx + \frac{\mu}{2} \left(\int_{\Omega} |d - d_0|^2 dx - \sigma^2 \right). \quad (9)$$

To find a minimizer for (7), we need to find the saddle points for L . The optimality conditions for the saddle points are

$$\begin{aligned} -\nabla \cdot \left(\frac{\nabla d}{|\nabla d|} - \vec{n} \right) + \mu(d - d_0) &= 0 \text{ in } \Omega \\ \left(\frac{\nabla d}{|\nabla d|} - \vec{n} \right) \cdot \vec{\nu} &= 0 \text{ on } \partial\Omega \end{aligned} \quad (10)$$

and

$$\int_{\Omega} |d - d_0|^2 dx = \sigma^2. \quad (11)$$

In (10), $\vec{\nu}$ is the outwards unit normal vector on the boundary $\partial\Omega$.

It is not easy to find the minimizer for (6). We shall use ideas from [24]. As \vec{n} is a unit vector, we use polar coordinate to represent it, i.e. $\vec{n} = (u, v) = (\cos \theta, \sin \theta)$. It is known that

$$|\nabla \vec{n}| = |\nabla \theta| \quad (12)$$

(see the Appendix for the details of the calculations). Thereupon, (6) becomes

$$\min_{\theta} \left\{ \int_{\Omega} |\nabla \theta| dx + \frac{\lambda}{2} \int_{\Omega} |\vec{n} - \vec{n}_0|^2 dx \right\}. \quad (13)$$

The optimality condition for θ for the above problem is

$$\nabla \cdot \frac{\nabla \theta}{|\nabla \theta|} - \lambda(\vec{n} - \vec{n}_0) \cdot \frac{\partial \vec{n}}{\partial \theta} = 0. \quad (14)$$

Thereafter, we introduce an artificial time variable t and note that θ is the steady state of the following equation:

$$\theta_t = \nabla \cdot \frac{\nabla \theta}{|\nabla \theta|} - \lambda(\vec{n} - \vec{n}_0) \cdot \frac{\partial \vec{n}}{\partial \theta}. \quad (15)$$

One could solve θ directly from the above equation. One of the troubles is that θ can be multivalued. As an alternative, we note that (c.f. [24])

$$\begin{aligned} \theta &= \tan^{-1} \left(\frac{v}{u} \right), \quad \nabla \theta = \frac{u \nabla v - v \nabla u}{u^2 + v^2} \text{ and} \\ \frac{\partial \vec{n}}{\partial \theta} &= (-\sin \theta, \cos \theta) = (-v, u) \end{aligned}$$

and with the notation $\vec{n}_0 = (u_0, v_0)$, it follows from (15) that

$$\begin{aligned} \frac{uv_t - vu_t}{u^2 + v^2} &= \operatorname{div} \left(\frac{u \nabla v - v \nabla u}{|u \nabla v - v \nabla u|} \right) \\ &\quad - \lambda [-(u - u_0)v + (v - v_0)u] \\ &= \operatorname{div} \left(\frac{u \nabla v - v \nabla u}{|u \nabla v - v \nabla u|} \right) \\ &\quad - \lambda [-uv + vu_0 + vu - uv_0]. \end{aligned} \quad (16)$$

We know that $u^2 + v^2 = 1$, which gives us $2uu_t + 2vv_t = 0$, so (16) can be separated into

$$v_t = u \operatorname{div} \left(\frac{u \nabla v - v \nabla u}{|u \nabla v - v \nabla u|} \right) - \lambda u [vu_0 - uv_0] \quad (17)$$

and

$$u_t = -v \operatorname{div} \left(\frac{u \nabla v - v \nabla u}{|u \nabla v - v \nabla u|} \right) + \lambda v [vu_0 - uv_0]. \quad (18)$$

To find the values of d and μ satisfying (10) and (11), we also introduce an artificial time variable t and solve the following equation to steady state

$$d_t = \operatorname{div} \left(\frac{\nabla d}{|\nabla d|} - \vec{n} \right) - \mu(d - d_0). \quad (19)$$

The value of μ also needs to be determined in such a way that the above equation has a steady state and the condition (11) is fulfilled at the steady state. We use the noise level constraint (11) and (10) to obtain such a formula for μ . Using the same idea as in [7], we multiply (10) by $d - d_0$ and integrate over Ω to get

$$\int_{\Omega} \nabla \cdot \left(\frac{\nabla d}{|\nabla d|} - \vec{n} \right) (d - d_0) dx = \mu \int_{\Omega} (d - d_0)^2 dx = \mu \sigma^2. \quad (20)$$

Using Green's theorem, we get the following formula for μ :

$$\mu = -\frac{1}{\sigma^2} \int_{\Omega} \left(\frac{\nabla d}{|\nabla d|} - \vec{n} \right) \cdot \nabla (d - d_0) dx. \quad (21)$$

In the numerical simulations, an explicit finite difference scheme is used to calculate d . The value of μ at each time level is updated according (21), see (30). In order to update μ using the above formula, we need to know the noise level approximately. In case that the noise level is not known, we can just fix a constant value for σ by trial and error. If we use a bigger value for the noise level σ , the restored image will be smoother. If we use a smaller value for the noise level, the algorithm is adding less diffusion to the image. Accordingly, the restored image will have better edge preservation, but the amount of noise removed from the image is also less in such a case.

In our simulations, we do not have a dynamic updating formula for λ and this parameter was chosen by trial and error. By inspecting (17) and (18) we see that $0 \ll \lambda$ is not a good choice since the dominating term then are

$$\begin{aligned} v_t &\approx -\lambda u [vu_0 - uv_0], \quad u \neq 0 \\ u_t &\approx +\lambda v [vu_0 - uv_0], \quad v \neq 0. \end{aligned} \quad (22)$$

Solve (22) to steady state will force $vu_0 - uv_0 = 0 \Rightarrow v = v_0$ and $u = u_0$. In this way, u and v will be as noisy as their observations and no progress is made. On the other hand, if $0 < \lambda \ll 1$

$$\begin{aligned} v_t &\approx +u \operatorname{div} \left(\frac{u \nabla v - v \nabla u}{|u \nabla v - v \nabla u|} \right) \\ u_t &\approx -v \operatorname{div} \left(\frac{u \nabla v - v \nabla u}{|u \nabla v - v \nabla u|} \right) \end{aligned} \quad (23)$$

would be the dominating part. Both equations in (23) reach steady state when $\nabla u \approx 0$ and $\nabla v \approx 0$, meaning that u and v are very smooth. A consequence of the extremity $u = \text{constant}$ and $v = \text{constant}$ is that (19) approaches the TV norm proposed in [7] (see the Appendix for details). Depending on the amount of noise, we found $1 < \lambda < 10$ to be a good choice.

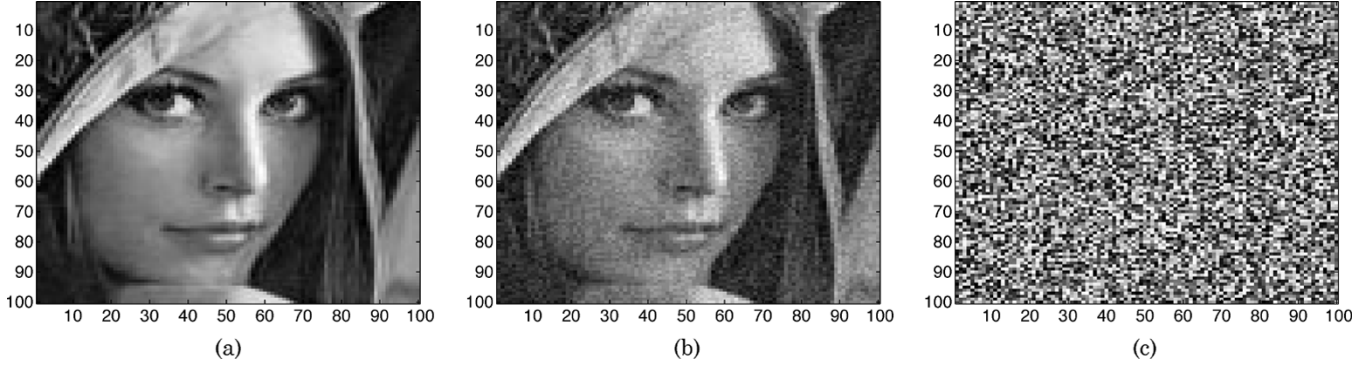


Fig. 1. Lena image. (a) Original image. (b) Noisy image SNR ≈ 20 . (c) Difference image.

Advantages: The proposed system of second-order PDEs are capable to restore edges and discontinuities in a better way than fourth-order PDEs and avoid the explicit computation of unstable higher order derivatives. Advantages over the classical TV minimization are that our model can recover smooth sub-surfaces and that it is more general (see the Appendix for details). An alternative approach for smoothing the flow field could be to use Gaussian filtering techniques directly on (u_0, v_0) to get (u, v) . This approach roughly halve the computing time of our two-step algorithm since (17) and (18) in step one are replaced with Gaussian filtering. However, the major disadvantage of classical Gaussian filtering techniques is the uniform smoothing in all directions of the flow field and fine details are easily destroyed with these filters. Furthermore, solving (17) and (18) guarantee that $|(u, v)| = 1$ at each iteration and this constraint is not necessarily satisfied everywhere if Gaussian filters are used to smooth the flow field.

Disadvantages: One of the drawbacks of the proposed method is the lack of a rigorous theory. We observe that our algorithm produces good results even though we still do not have any solid theoretical justification for this successful performance. Further, since we solve two second-order nonlinear PDEs sequentially, there may be a loss of efficiency compared with other second-order nonlinear PDE methods. For instance, the classical TV model is roughly twice as fast as the proposed model due to the performance of normal smoothing in the proposed model. Moreover, this is a two-parameter method and the parameter λ must be found by trial and error.

III. IMPLEMENTATION

We discretize the system (17) and (18) by finite differences and for simplicity we introduce

$$[\text{div}]^n := \left[\text{div} \left(\frac{u \nabla v - v \nabla u}{|u \nabla v - v \nabla u|} \right) \right]^n \quad \text{and} \quad [vu]^n := [vu_0 - uv_0]^n.$$

Details of how to discretize $[\text{div}]^n$ in space will follow the same scheme as for $\text{div}(\nabla d/|\nabla d|)$ described in (29). The following semi-implicit scheme of [24] is used to solve u and v :

$$u^{n+1} = u^n + \frac{\Delta t}{2} (v^{n+1} + v^n) (-[\text{div}]^n + \lambda[vu]^n) \quad (24)$$

$$v^{n+1} = v^n + \frac{\Delta t}{2} (u^{n+1} + u^n) ([\text{div}]^n - \lambda[vu]^n). \quad (25)$$

To solve the previous algebraic system v^{n+1} is used in (24) to get

$$\begin{aligned} u^{n+1} &= u^n + \frac{\Delta t}{2} \\ &\times \left(v^n + \frac{\Delta t}{2} (u^{n+1} + u^n) ([\text{div}]^n - \lambda[vu]^n) + v^n \right) \\ &\times (-[\text{div}]^n + \lambda[vu]^n) \\ &= u^n + v^n \Delta t (-[\text{div}]^n + \lambda[vu]^n) - \left(\frac{\Delta t}{2} \right)^2 \\ &\times ([\text{div}]^n - \lambda[vu]^n)^2 (u^{n+1} + u^n). \end{aligned} \quad (26)$$

The unknown u^{n+1} is collected on the right hand side and an explicit formula is obtained for (18) [shown in (27), at the bottom of the page] and (17) is approximated by the same technique [shown in (28), at the bottom of the next page]. From (27) and (28), it is easy to calculate that

$$|u^{n+1}|^2 + |v^{n+1}|^2 = |u^n|^2 + |v^n|^2, \quad \forall n.$$

As soon as steady state is reached for (27) and (28), we fix $(u^{n+1}, v^{n+1}) = (u, v)$ to be the components of the unit normal vector. Let us use the notation $\Delta_{\mp}^x d_{i,j} = \mp(d_{i\mp 1,j} - d_{i,j})$ and $\Delta_{\mp}^y d_{i,j} = \mp(d_{i,j\mp 1} - d_{i,j})$ for backward and forward difference at a given pixel (i, j) . The numerical approximation to (19) and (21) is shown in (29) at the bottom of the next page. Here, we have used the notation

$$m(a, b) = \text{minmod}(a, b) = \left(\frac{\text{sign } a + \text{sign } b}{2} \right) \min(|a|, |b|)$$

$$u_{i,j}^{n+1} = \frac{u_{i,j}^n - v_{i,j}^n \Delta t ([\text{div}]_{i,j}^n - \lambda[vu]_{i,j}^n) - u_{i,j}^n \left(\frac{\Delta t}{2} \right)^2 ([\text{div}]_{i,j}^n - \lambda[vu]_{i,j}^n)^2}{1 + \left(\frac{\Delta t}{2} \right)^2 ([\text{div}]_{i,j}^n - \lambda[vu]_{i,j}^n)^2} \quad (27)$$

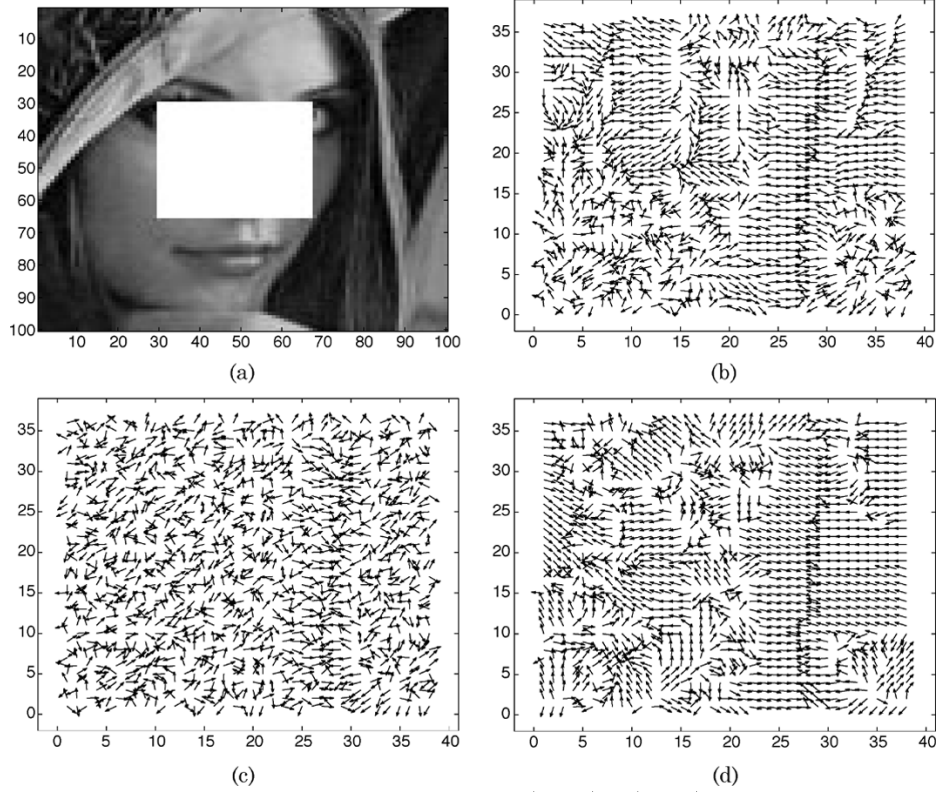


Fig. 2. For better visualization, we have only plotted the normal vectors in $\Omega_1 = (30, 65) \times (30, 67)$ indicated in (a) by the bright rectangular $\Omega_1 \subset \Omega$. (b) Original normals. (c) Noisy normals. (d) Processed normals.

and μ is defined discretely via

$$\begin{aligned} \mu^n = & -\frac{h}{\sigma^2} \\ & \times \sum_{i,j} \left[\left(\frac{\Delta_+^x d_{i,j}^n}{[(\Delta_+^x d_{i,j}^n)^2 + (m(\Delta_+^y d_{i,j}^n, \Delta_-^y d_{i,j}^n))^2]^{\frac{1}{2}}} - u_{i,j} \right) \right. \\ & \times \Delta_+^x (d_{i,j}^n - d_{i,j}^0) \\ & + \left(\frac{\Delta_+^y d_{i,j}^n}{[(\Delta_+^y d_{i,j}^n)^2 + (m(\Delta_+^x d_{i,j}^n, \Delta_-^x d_{i,j}^n))^2]^{\frac{1}{2}}} - v_{i,j} \right) \\ & \left. \times \Delta_+^y (d_{i,j}^n - d_{i,j}^0) \right]. \end{aligned} \quad (30)$$

The evaluation of $[\text{div}]^n$ is done similarly as shown in (29). The $\text{minmod}(\cdot, \cdot)$ action is applied to the terms $uv_x - vu_x$ and $uv_y - vu_y$.

If we hold $u = v = 0$ during the iterations between (29) and (30), the updating of (29) and (30) is exactly the original TV-smoothing algorithm of [7]. This means that the TV smoothing is trying to force the normal vectors to be (0,0) and the method proposed here is trying to force the normal vectors to be closed to the smoothed normals. This is one of the important reasons that the proposed is able to preserve edges and still gives good results in smooth regions.

To evaluate $\nabla d_0 / |\nabla d_0|$, we need $|\nabla d_0| \neq 0$. In order to overcome this problem in the simulations, we use the standard trick to replace $\nabla d_0 / |\nabla d_0|$ by $\nabla d_0 / \sqrt{|\nabla d_0|^2 + \epsilon}$ where ϵ is a small number. Normally, we choose ϵ to be very small (for

$$v_{i,j}^{n+1} = \frac{v_{i,j}^n + u_{i,j}^n \Delta t ([\text{div}]_{i,j}^n - \lambda[vu]_{i,j}^n) - v_{i,j}^n \left(\frac{\Delta t}{2}\right)^2 ([\text{div}]_{i,j}^n - \lambda[vu]_{i,j}^n)^2}{1 + \left(\frac{\Delta t}{2}\right)^2 ([\text{div}]_{i,j}^n - \lambda[vu]_{i,j}^n)^2} \quad (28)$$

$$\begin{aligned} d_{i,j}^{n+1} = & d_{i,j}^n + \frac{\Delta t}{h} \left[\Delta_-^x \left(\frac{\Delta_+^x d_{i,j}^n}{[(\Delta_+^x d_{i,j}^n)^2 + (m(\Delta_+^y d_{i,j}^n, \Delta_-^y d_{i,j}^n))^2]^{\frac{1}{2}}} - u_{i,j} \right) \right. \\ & + \Delta_-^y \left(\frac{\Delta_+^y d_{i,j}^n}{[(\Delta_+^y d_{i,j}^n)^2 + (m(\Delta_+^x d_{i,j}^n, \Delta_-^x d_{i,j}^n))^2]^{\frac{1}{2}}} - v_{i,j} \right) \\ & \left. - \Delta t \mu^n (d_{i,j}^n - d_{i,j}^0) \right]. \end{aligned} \quad (29)$$

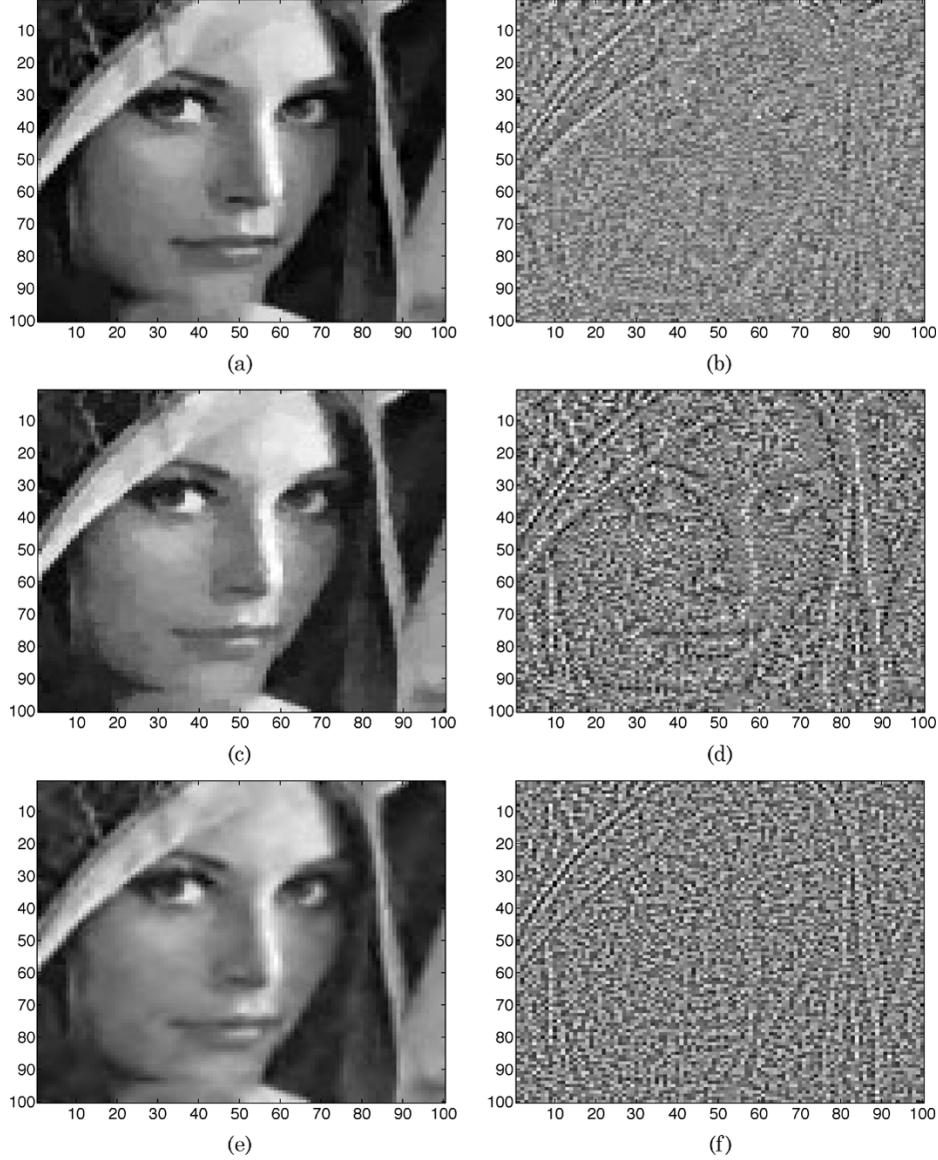


Fig. 3. Results for the Lena image. (a) Our new method. (b) Difference image. (c) TV method. (d) Difference image. (e) Fourth-order method. (f) Difference image.

example $\epsilon = 10^{-9}$) and the value of ϵ does not seem to influence the results much. The same trick is used for the calculation of $[\text{div}]^n$.

To summarize, our noise removal approach is done in the following two steps.

- 1) Choose a positive λ and take $\nabla d_0/|\nabla d_0|$ to be the initial values for (u, v) , solve (27) and (28) to steady state.
- 2) Take d_0 as the initial value for d and let (u, v) be the value of the smoothed normals. Solve (29) and (30) to steady state.

IV. NUMERICAL RESULTS

In this part, we present some of the results obtained with our system of coupled equations. First of all, we wish to compare our result with two related methods [7], [11]. Both [7], [11] have their strengths and weakness and it will be shown that our new method does an overall better job than both of them on

the set of examples tested. The classical TV model from [7] is known to give good results for almost all kinds of images. The chief criticism is that smooth regions are transformed into piecewise constant regions. On the other hand, the TV method works almost perfectly for block images. To avoid the staircase effect Lysaker–Lundervold–Tai [11] suggested a fourth-order PDE. By construction, this approach allows linear changes in the intensity value and is therefore well suited for processing images with smooth transitions like a human face. From a theoretical point of view [11] will not fulfill the good property of the TV method along edges. For the evaluations, we have used images like Fig. 1, Fig. 4, Fig. 6, and some others to show the robustness, strength, and weakness for the different algorithms concerned here.

Example 1: In this first test, the well-known Lena image is corrupted with noise. From the original and noisy image, we calculate the original and noisy normal vectors. We use the noisy observations as input to our algorithm and in this specific test

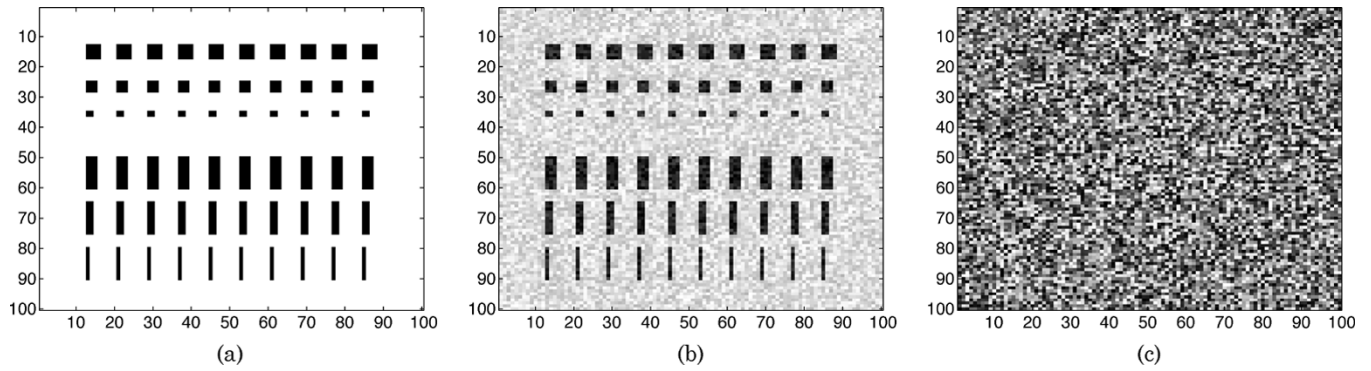


Fig. 4. Blocky image. (a) Original image. (b) Noisy image SNR ≈ 8 . (c) Difference image.

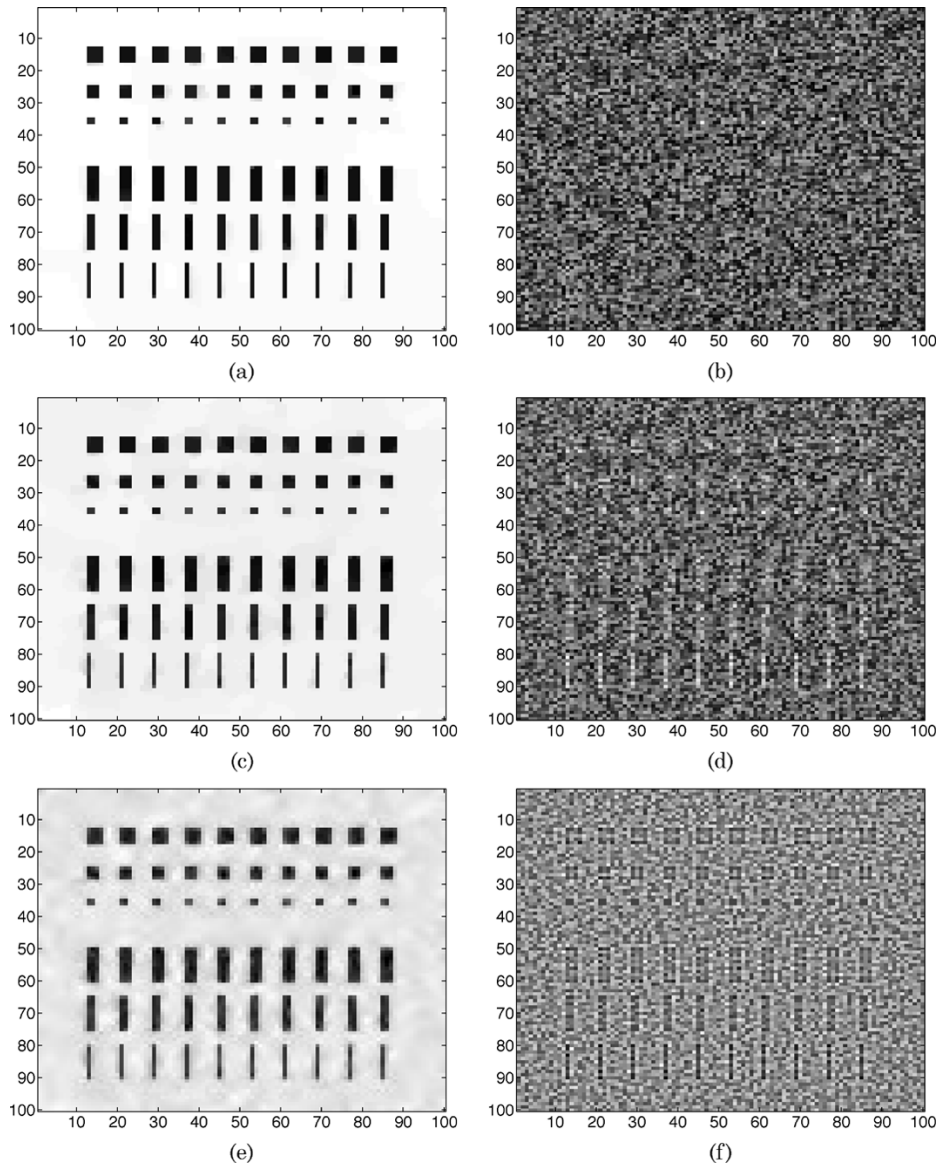


Fig. 5. Comparisons for a blocky image. (a) Our new method. (b) Difference image. (c) TV method. (d) Difference image. (e) Fourth-order method. (f) Difference image.

we have used $\lambda = 2.15$. For better visualization, we have only plotted a small portion of the flow field for the images in Fig. 2. The rectangle in Fig. 2(a) indicates the area we have plotted.

The processed unit vectors point in the wrong directions some places in Fig. 2(d), but it is certainly an improvement

compared with Fig. 2(c) where the normals more or less have a random orientation. Together with (19) and (21), we use the smoothed normals to restore a new image for the noisy one. The result is compared with the TV method (i.e. $u = 0$ and $v = 0$) and the fourth-order smoother [11]. We also

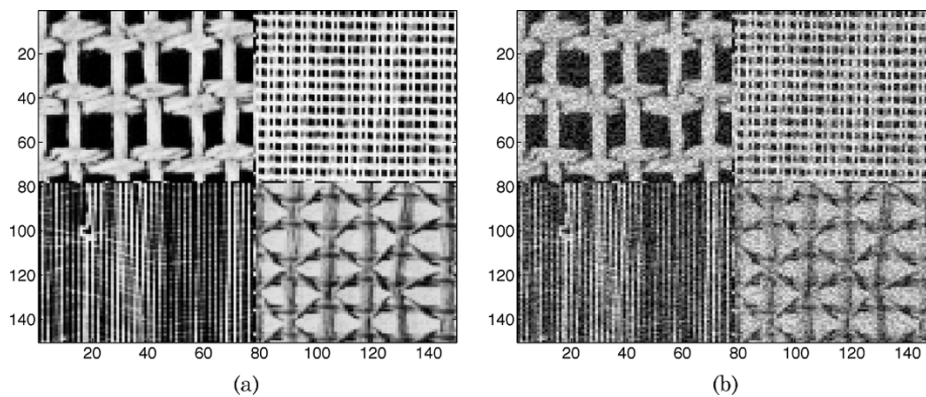


Fig. 6. Brodatz image for evaluation. (a) Original image. (b) Noisy image SNR ≈ 9 .

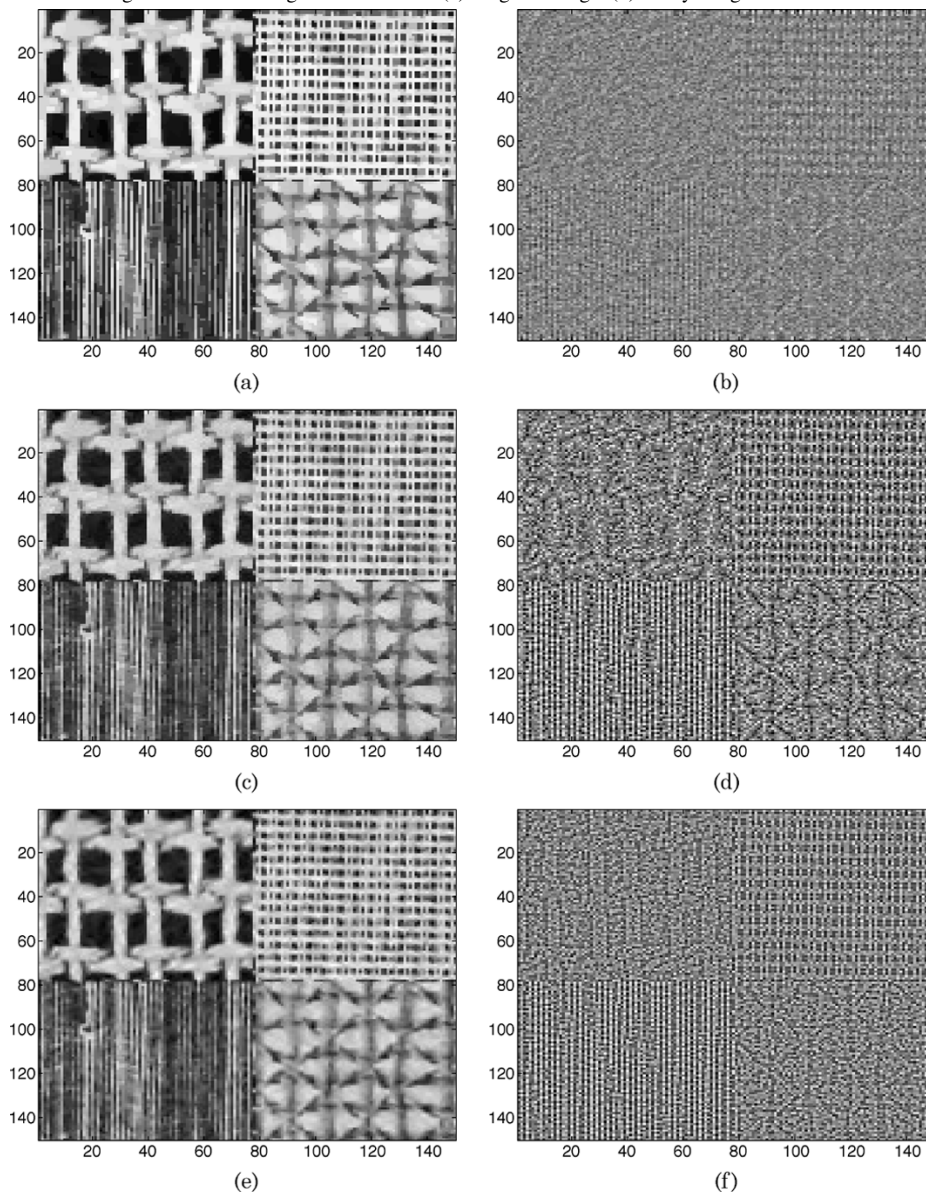


Fig. 7. Results for the Brodatz image. (a) Our result. (b) Difference image. (c) TV result. (d) Difference image. (e) Fourth-order result. (f) Difference image.

visualize the difference between the input and output image. In an ideal case, the difference image should only contain noise as indicated in Fig. 1(c).

From the restored images in Fig. 3, it is clear that much of the noise is suppressed. As expected, the TV algorithm transforms smooth regions into piecewise constant regions (see Lena's

cheek). By evaluating the image of the difference between the smoothed image and the noisy image, it is obvious that neither the fourth-order smoother or the TV method are as good as our new method.

Example 2: A blocky image which should favor the TV method is used in this second test. Note that some of the objects

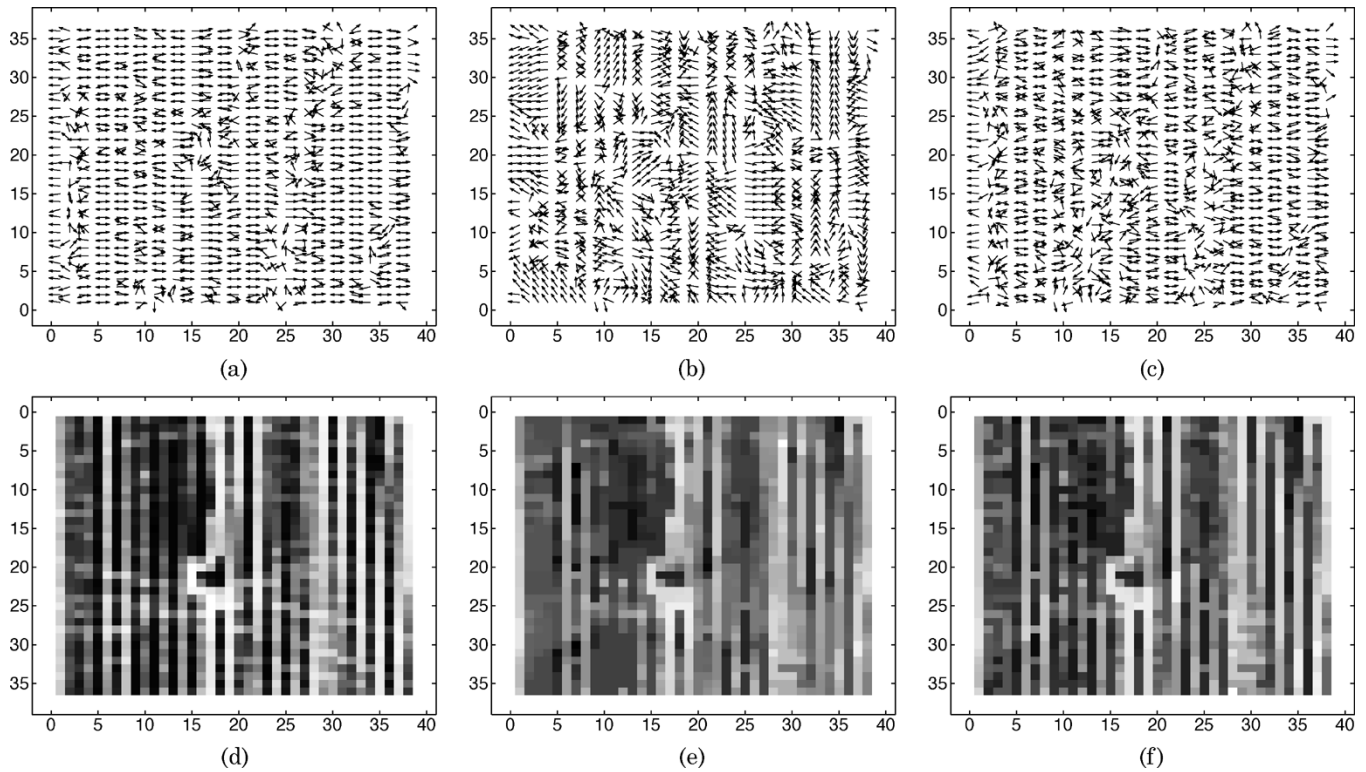


Fig. 8. Robustness of λ . (a) True normals. (b) Normals with $\lambda = 2$. (c) Normals with $\lambda = 10$. (d) True image (e) Processed using (b) and (f) processed using (c).

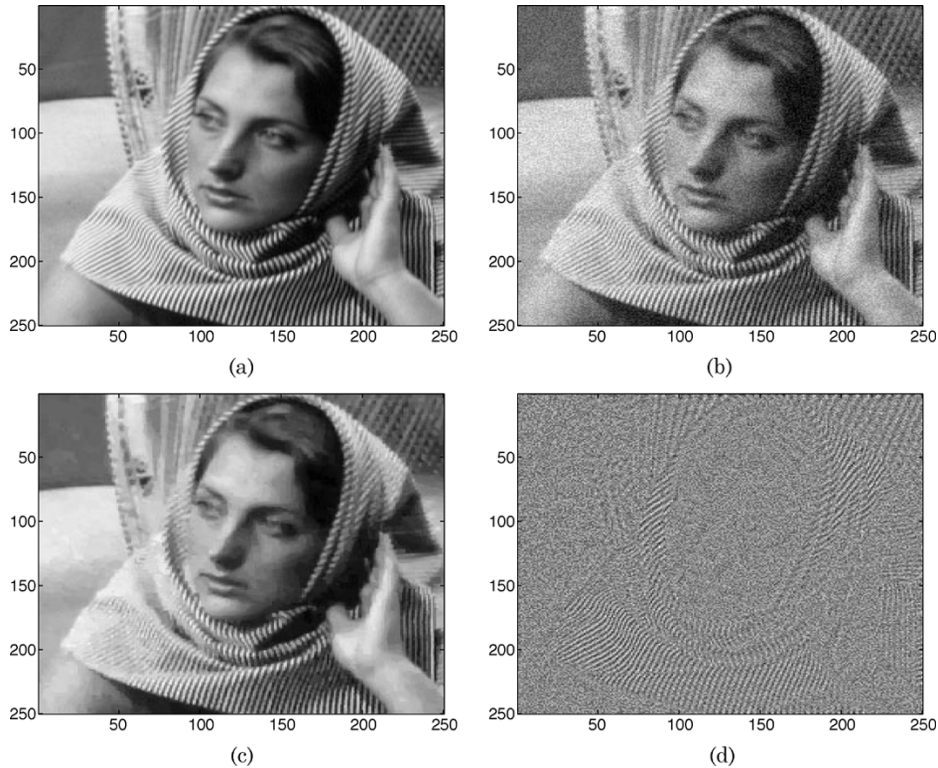


Fig. 9. Image with a human face and some textures. (a) Original image. (b) Noisy image $\text{SNR} \approx 20$. (c) Our result. (d) Difference image.

are as small as 2×2 pixels and other as narrow as 1×10 pixels. It is a nontrivial case to smooth out noise and simultaneously maintain all edges for an image like this.

The amount of noise is increased and we fix $\lambda = 5$ during the normal processing. With this kind of synthesized images $\nabla d = 0$ almost everywhere in Ω . Due to this, we do not show

the unit normal vectors but just evaluate the final result. The restored image obtained with our method is also compared with the two methods mentioned above.

We see that all methods are able to suppress much noise, but some ghosts are visible in the image restored by the fourth-order scheme (around the narrow object of 1×10 pixels). This is not

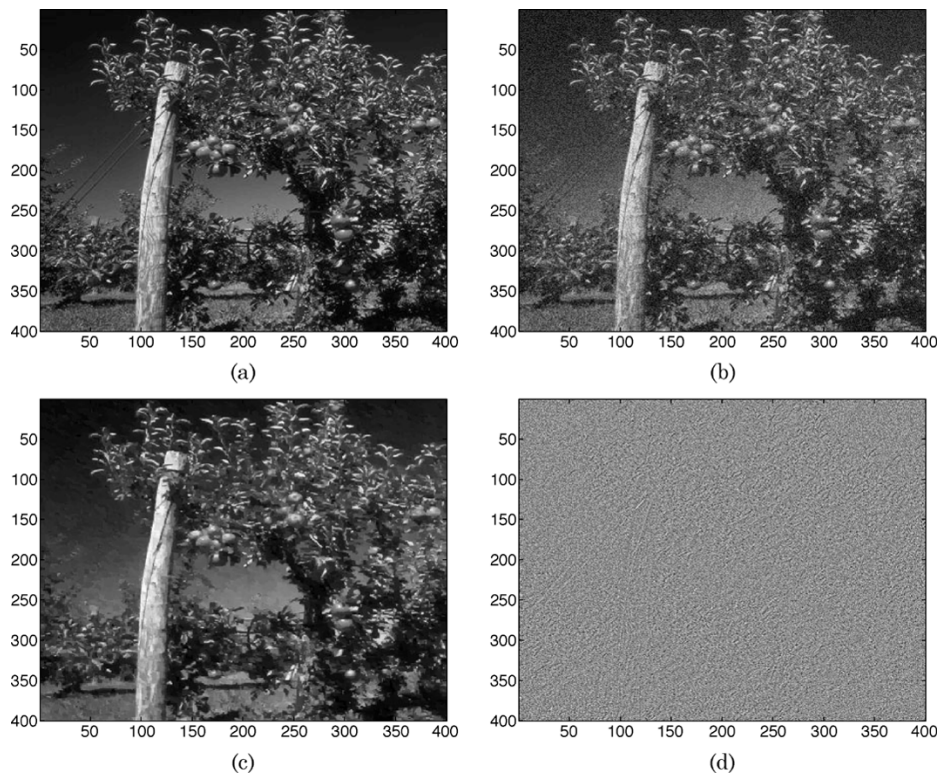


Fig. 10. Image with a tree and leaves. (a) Original image. (b) Noisy image $\text{SNR} \approx 7$. (c) Our result. (d) Difference image.

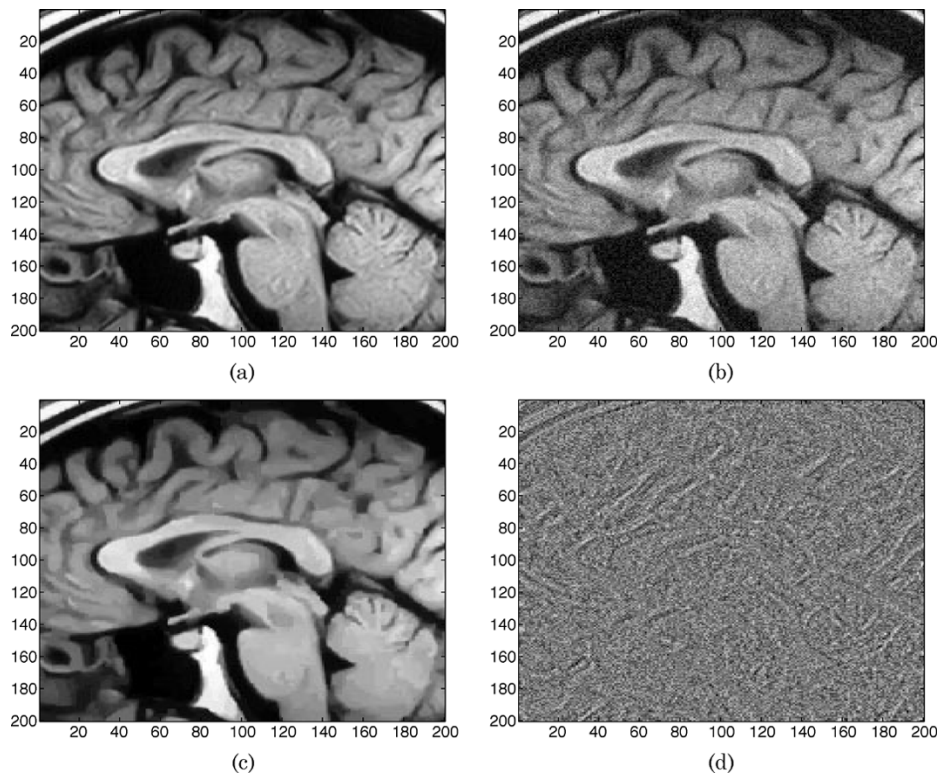


Fig. 11. MR brain image. (a) Original image. (b) Noisy image $\text{SNR} \approx 20$. (c) Our result. (d) Difference image.

surprising since the fourth-order scheme does not allow discontinuous jumps in the same way as the other two methods do. From Fig. 5(f) it is clear that the fourth-order scheme filters out some edges, and in some sense the same is also observed in Fig. 5(d).

Example 3: Here, we try our new method on an image composed of four different Brodatz textures as given in Fig. 6(a).

Processing texture images is generally a hard task since fine texture details often are filtered out. The texture in the lower left and upper right part of Fig. 6(a) mostly contains high frequency

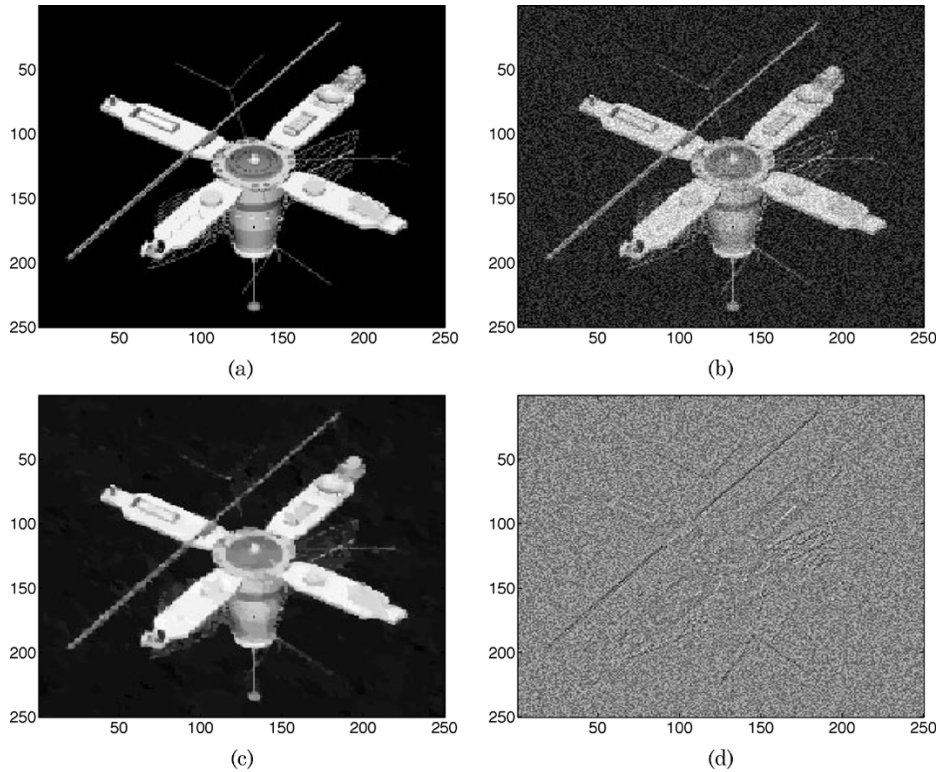


Fig. 12. Results for a satellite image. (a) Original image. (b) Noisy image SNR ≈ 5 . (c) Our result. (d) Difference image.

information (i.e. very oscillating intensity values) and can easily be mixed up with noise. From Fig. 7(b), we see that this is the part of the image with the poorest result, but the main quality is maintained. Both Fig. 7(d) and Fig. 7(f) reveal oversmoothing for some of the fine structures. Methods like [12] may be better suited to deal with texture images, but this example shows the robustness of our new algorithm.

In Fig. 8, we visualize two results using different values for λ . The purpose of this test is to show that our algorithm is robust with respect to this parameter. We use the image depicted in Fig. 6(b) as our observation data, but for better visualization, we only plot a small part of the lower left corner. By choosing $\lambda = 2$, we smooth the flow field too much and the final image depicted in Fig. 8(e) misses some details. Using $\lambda = 10$, the flow field resembles its initial state and the full potential of the algorithm is not exploited. However, these deviations are hardly noticeable when evaluating the images in full scale.

We will end this section by showing some more results from a texture image, a MR image, a landscape image, and a satellite image. The rest of this paper will only deal with results obtained with our new method.

Our next example uses an image containing both a human face and some textures. The challenge with this image is to maintain both texture details and smooth transitions in the human face during processing.

The background and human feature like a hand, shoulder, and face is restored in a proper way, but the difference image tells us that textures on the scarf is smoothed to much (see Fig. 9).

An image of leaves on a tree is evaluated next (see Fig. 10). Restoration algorithms may have troubles with this kind of images due to the lack of connection in the image. Observe that the

leaves are not smeared together and this is an important quality. The only negative remark is that the wire (going left from the top of the pole) disappeared in some places. The same effect was observed for the TV method and the fourth-order smoother. The result is not reported here, but both of them reconstruct an image which is almost as good as the one given in Fig. 10(c).

In our next example, an MR image is corrupted with noise, see Fig. 11. We use an image of a human brain, zoomed so it is possible to see all the fine features in the tissues. We see that the restoration algorithm is able to maintain all important information in the image, and in the same time filter out noise. This means that the intensity value is more accurately computed inside each tissue region after processing.

The final example concerns a satellite image. The recovered image coincides with the true one almost everywhere (c.f. Fig. 12).

We have also carried out a qualitative evaluation of the different schemes. In real applications, the exact noise level is seldom known. To simulate such a case, the value we used for σ is not the true noise level, but an approximation for the true noise level. Let $\bar{\sigma}$ denote the true noise level and σ denote the noise level we used for the algorithm. We investigate the performance of the schemes using two different images. For the texture image Fig. 6(b), we challenge the schemes by using a poor estimate for the amount of noise when we deal with the noise constraint $\sigma^2 = (1/2)\bar{\sigma}^2$. For Fig. 4(b), we use the estimate $\sigma^2 = (85/100)\bar{\sigma}^2$.

In Table I, the reduction ratio of the error measured in L_2 norm is reported in the second and fourth columns, and the improvement ratio of SNR is reported in the third and fifth column. From Table I, we see that our method gives noticeable better

TABLE I
EVALUATION OF SNR AND L_2 -NORM FOR EACH OF THE SCHEMES

Method	$\sigma^2 = \frac{1}{2}\bar{\sigma}^2$		$\sigma^2 = \frac{85}{100}\bar{\sigma}^2$	
Our	1.9	1.6	20	16
TV	1.6	1.3	8	6
Fourth order	1.5	1.2	4	3

results than the two others. We obtain higher SNR improvements and the reduction ratio of the error (compared with the true image) measured in L_2 -norm are significant higher for the set of examples tested. We want to emphasize that several statistical approach [26]–[28] can be used to get much closer estimation of noise variance than that which we evaluated in Table I.

V. CONCLUSION

A variational method for filtering gray-scale images corrupted by additive noise is proposed in this paper. Our method considers two second-order nonlinear PDEs which are solved sequentially. The interpretation is simple; first we denoise the normals for the level sets of the image intensity function, and, thereafter, we try to find an image which fits the smoothed normals. With this approach, geometric information of the level contours are incorporated in our image processing model. If we reject the smooth normals obtained in the first step, the image processing step reduces to the original TV method.

Numerical experiments substantiate that our composed method holds three important qualities; it is superior in recovering sharp edges of an image, and, second, it enhances the recovery of smooth subsurfaces contained in the image. Third, we have shown that it is easy to control the amount of smoothing with our method.

APPENDIX

- a) We give the details for the calculations for (12). Given $\theta = \theta(x_1, x_2)$

$$\nabla\theta = \left(\frac{\partial\theta}{\partial x_1}, \frac{\partial\theta}{\partial x_2} \right) \quad \text{and} \quad |\nabla\theta| = \sqrt{\left(\frac{\partial\theta}{\partial x_1} \right)^2 + \left(\frac{\partial\theta}{\partial x_2} \right)^2}. \quad (31)$$

The gradient matrix and its norm of the vector-valued function \mathbf{T} are respectively defined by

$$\nabla\mathbf{T} = \begin{pmatrix} \nabla T_1 \\ \nabla T_2 \end{pmatrix} = \begin{pmatrix} \frac{\partial T_1}{\partial x_1} & \frac{\partial T_1}{\partial x_2} \\ \frac{\partial T_2}{\partial x_1} & \frac{\partial T_2}{\partial x_2} \end{pmatrix} \quad \text{and} \quad |\nabla\mathbf{T}| = \sqrt{\sum_{i=1}^2 \sum_{j=1}^2 \left(\frac{\partial T_i}{\partial x_j} \right)^2}. \quad (32)$$

Let $\vec{n} = (\cos\theta, \sin\theta)$, it is easy to see that [shown in (33), at the bottom of the page]

- b) We illuminate some details concerning the original TV-norm filter [7] and our approach. The restoration model we propose here can be interpreted as the steady state solution of the nonlinear diffusion process

$$d_t = \nabla \cdot \left(\frac{\nabla d}{|\nabla d|} - \vec{n} \right) + \lambda(d - d_0), \quad \text{where } d(x, y, 0) = d_0(x, y). \quad (34)$$

An interesting observation is that this PDE reduces to the PDE in [7] if we choose $\vec{n} = 0$ in (34), i.e.

$$d_t = \nabla \cdot \left(\frac{\nabla d}{|\nabla d|} \right) + \lambda(d - d_0), \quad \text{where } d(x, y, 0) = d_0(x, y). \quad (35)$$

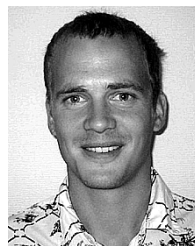
Note also that setting $\tilde{d}_0 = d_0 + (1/\lambda)\nabla \cdot \vec{n}$ in (34), we get (35), but with a new initial value $\tilde{d}_0(x, y)$ replacing $d_0(x, y)$.

REFERENCES

- [1] J. B. Weaver, X. Yansun Jr., D. M. Healy, and L. D. Cromwell, "Filtering noise from images with wavelet transforms," *Magn. Res. Med.*, vol. 24, pp. 288–295, 1991.
- [2] D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biomet.*, vol. 81, pp. 425–455, 1994.
- [3] —, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Stat. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.
- [4] A. B. Hamza and H. Krim, "Image denoising: a nonlinear robust statistical approach," *IEEE Trans. Signal Processing*, vol. 49, pp. 3045–3054, Dec. 2001.
- [5] V. P. Melnik, I. Shmulevich, K. Egiazarian, and J. Astola, "Block-median pyramidal transform: analysis and denoising applications," *IEEE Trans. Signal Processing*, vol. 49, pp. 364–372, Feb. 2001.
- [6] R. C. Gonzales and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1993.
- [7] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
- [8] A. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 629–639, July 1990.

$$\begin{aligned} |\nabla\vec{n}| &= \sqrt{\left(-\sin\theta \frac{\partial\theta}{\partial x_1} \right)^2 + \left(-\sin\theta \frac{\partial\theta}{\partial x_2} \right)^2 + \left(\cos\theta \frac{\partial\theta}{\partial x_1} \right)^2 + \left(\cos\theta \frac{\partial\theta}{\partial x_2} \right)^2} \\ &= \sqrt{\left(\frac{\partial\theta}{\partial x_1} \right)^2 (\sin^2\theta + \cos^2\theta) + \left(\frac{\partial\theta}{\partial x_2} \right)^2 (\sin^2\theta + \cos^2\theta)} \\ &= \sqrt{\left(\frac{\partial\theta}{\partial x_1} \right)^2 + \left(\frac{\partial\theta}{\partial x_2} \right)^2} = |\nabla\theta|. \end{aligned} \quad (33)$$

- [9] R. A. Carmona and S. Zhong, "Adaptive smoothing respecting feature directions," *IEEE Trans. Image Processing*, vol. 7, pp. 353–358, Mar. 1998.
- [10] J. Weickert, *Anisotropic Diffusion in Image Processing*. Stuttgart, Germany: B. G. Teubner, 1998.
- [11] M. Lysaker, A. Lundervold, and X. C. Tai, "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images is space and time," *IEEE Trans. Image Processing*, vol. 12, pp. 1579–1590, Dec. 2003.
- [12] S. Osher, A. Sole, and L. Vese, "Image decomposition and restoration using total variation minimization and the H^{-1} norm," *Multiscale Modeling and Simulation: A SIAM Interdisciplinary J.*, vol. 1, no. 3, pp. 349–370, 2003.
- [13] T. Chan, A. Marquina, and P. Mulet, "High-order total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 22, no. 2, pp. 503–516, 2000.
- [14] Y.-L. You and M. Kaveh, "Fourth-order partial differential equation for noise removal," *IEEE Trans. Image Processing*, vol. 9, pp. 1723–1730, Oct. 2000.
- [15] J. B. Greer and A. L. Bertozzi, "Traveling wave solutions of fourth order PDE's for image processing," Univ. California, Los Angeles, Tech. rep. (03-25), 2003.
- [16] A. L. Bertozzi and J. B. Greer, "Low Curvature image simplifiers: Global regularity of Ssmooth solutions and Laplacian limiting schemes," Univ. California, Los Angeles, Tech. rep. (03-26), 2003.
- [17] C. Kenney and J. Langan, *A New Image Processing Primitive: Reconstructing Images From Modified Flow Fields*. Santa Barbara, CA: Univ. California, 1999, pp. 1–10.
- [18] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric surface processing via normal maps," *ACM Trans. Graph. (TOG)*, vol. 22, no. 4, pp. 1012–1033, 2003.
- [19] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.
- [20] B. K. P. Horn and M. J. Brooks, "The variational approach to shape from shading," *Comput. Vis. Graph. Image Process.*, vol. 33, no. 2, pp. 174–208, 1986.
- [21] Y. Ohtake, A. Belyaev, and I. Bogaevski, "Mesh regularization and adaptive smoothing," *Comput.-Aided Design*, vol. 33, pp. 789–800, 2001.
- [22] Y. Ohtake and A. G. Belyaev, "Dual/primal mesh optimization for polygonized implicit surfaces," in *Proc. ACM Solid Modeling*, Saarbrücken, Germany, June 17–21, 2002, pp. 171–178.
- [23] 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*, vol. 9, pp. 1200–1211, Oct. 2000.
- [24] L. A. Vese and S. Osher, "Numerical methods for p -harmonic flows and applications to image processing," *SIAM J. Numer. Anal.*, vol. 40, no. 6, pp. 2085–2104, 2002.
- [25] T. F. Chan and X.-C. Tai, "Identification of discontinuous coefficient from elliptic problems using total variation regularization," *SIAM J. Sci. Comput.*, vol. 25, no. 3, pp. 881–904, 2003.
- [26] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 8, pp. 679–698, June 1986.
- [27] P. Besl and R. Jain, "Segmentation through variable-order surface fitting," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 167–192, Feb. 1988.
- [28] P. Meer, J.-M. Jolion, and A. Rosenfeld, "A fast parallel algorithm for blind estimation of noise variance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 216–223, Feb. 1990.



Marius Lysaker received the M.S. degree in applied mathematics in 2001 from the University of Bergen, Bergen, Norway. The topic of his M.S. thesis was to develop algorithms to denoise MRI images via partial differential equations. He is currently pursuing the Ph.D. degree in applied mathematics at the University of Bergen.

His research interests include constructions, denoising, and segmentation of medical images (MRI/PET). During the last few years, he has mainly focused on level set methods. Since 2004, he has held a Postdoctorate position at the Department of Scientific Computing, Simula Research Laboratory AS, Oslo, Norway.



Stanley Osher received the M.S. and Ph.D. degrees from the Courant Institute, New York University.

He was with Brookhaven National Laboratory, University of California, Berkeley, and the State University of New York, Stony Brook, and he has been with the University of California, Los Angeles (UCLA), since 1976. He is the Director of Special Projects at the Institute for Pure and Applied Mathematics, UCLA. He is the coinventor of level set methods for computing moving fronts (8600 references on GOOGLE, ENO, and WENO) and

other numerical methods for computing solutions to hyperbolic conservation laws and Hamilton–Jacobi equations, and total variation and other PDE-based image processing techniques. He has cofounded three companies, based, in part, on his own research. His work has been written up numerous times in the scientific and international media, e.g., *Science News*, *Die Zeit* (1999). He is a highly cited researcher according to web-of-science and an Associate Editor of five major journals.

Dr. Osher has been a Fulbright and Alfred P. Sloan Fellow, he received the NASA Public Service Group Achievement Award, the Japan Society of Mechanical Engineers Computational Mechanics Award, and was an invited speaker at the International Congress of Mathematicians, and recently received the SIAM Pioneer Prize at the ICIAM conference last summer.



Xue-Cheng Tai received the Licenciante degree in 1989 and the Ph.D. degree in 1991 in applied mathematics from Jyväskylä University, Finland. The subject of his Ph.D. dissertation was on inverse problems and parallel computing.

After holding several research positions in Europe, he was employed as an Associate Professor in 1994 at the University of Bergen, Bergen, Norway and as a Professor since 1997. He has also worked as a part-time Senior Scientist at the private company Rogaland Research. He is now a member of the Center for

Mathematics for Applications in Oslo, Norway, and a member of the Center of Integrated Petroleum Research, Bergen, Norway. His research interests include Numerical PDE for image processing, multigrid and domain decomposition methods, iterative methods for linear and nonlinear PDE problems, and parallel computing. He has educated numerous M.S. and Ph.D. students and has published more than 60 scientific papers. He has been a Reviewer and Editor for several international journals.