A FAST MODIFIED NEWTON'S METHOD FOR CURVATURE BASED DENOISING OF 1D SIGNALS

ANDY M. YIP
Department of Mathematics
National University of Singapore
Block S17, 10, Lower Kent Ridge Road, 119076, Singapore

WEI ZHU
Department of Mathematics
University of Alabama
Box 870350, Tuscaloosa, AL 35487, USA

(Dedicated to Professor Tony F. Chan’s 60th Birthday)

ABSTRACT. We propose a novel fast numerical method for denoising of 1D signals based on curvature minimization. Motivated by the primal-dual formulation for total variation minimization introduced by Chan, Golub, and Mulet, the proposed method makes use of some auxiliary variables to reformulate the stiff terms presented in the Euler-Lagrange equation which is a fourth-order differential equation. A direct application of Newton’s method to the resulting system of equations often fails to converge. We propose a modified Newton’s iteration which exhibits local superlinear convergence and global convergence in practical settings. The method is much faster than other existing methods for the model. Unlike all other existing methods, it also does not require tuning any additional parameter besides the model parameter. Numerical experiments are presented to demonstrate the effectiveness of the proposed method.

1. Introduction. Images acquired by imaging systems are unavoidably contaminated with noise. Image denoising is used to recover the clean image from the observed image. In additive noise models, the acquired image \( f \) defined on an open domain \( \Omega \subset \mathbb{R}^2 \) is modeled as the sum of the original clean image \( u \) and the noise \( n \). We consider Gaussian white noise, where the noise \( n \) is independent of the image \( u \) and \( \{n(x)\}_{x \in \Omega} \) are independent and follow a zero-mean Gaussian distribution with a common variance \( \sigma^2 \).

A variety of approaches for image denoising have been proposed. They include classical linear and non-linear filtering, wavelet shrinkage, statistical models, partial differential equation (PDE) models, and variational methods [3, 12]. We are particularly interested in variational methods. In this kind of methods, the denoised image is exhibited as the minimizer of an objective function. The main advantage of this approach is that geometric properties of the denoised image are directly controlled. A classical method of this type is the well-known total variation (TV) model...
The TV model seeks the minimizer of the energy functional

\[
E_{TV}(u) := \frac{1}{2} \int_{\Omega} (u - f)^2 \, dx + \lambda \int_{\Omega} |\nabla u| \, dx
\]

over the set of all functions on \( \Omega \) with bounded variation. Here, \( \int_{\Omega} (f - u)^2 \, dx \) is the data fidelity term, \( \int_{\Omega} |\nabla u| \, dx \) is the TV regularization term, and \( \lambda > 0 \) is a tuning parameter. The fidelity term ensures the solution \( u \) to stay close to \( f \), so that the major signals are inherited. The regularization term penalizes the presence of oscillations (noise) in the solution to achieve denoising. The main advantage of TV regularization is that edges, as discontinuities of \( u \), are preserved. In contrast, the classical \( H^1 \) regularization \( \int_{\Omega} |\nabla u|^2 \, dx \), the ancestor of TV regularization, smears out all edges.

While the TV model is a simple and elegant model that can produce denoised images with good quality and sharp edges, there are some limitations on the kind of features that can be recovered. For example, intensity contrast, corners, and textures are not preserved [14]. Moreover, the model privileges piecewise constant features, so that smooth regions can become staircases. To fix these issues, various new models have been proposed. Textures can be separated through the use of the G-norm [25]. The inf-convolution model [8], the spatially adaptive TV model [5], and numerous models based on higher-order derivatives [8, 13, 11, 18, 22, 31, 34] can be used to reduce the staircase effect. In particular, we focus on the mean curvature model proposed by Zhu and Chan [37]. Various kinds of curvature of curves and surfaces have also been used in other image and geometric processing tasks, for example, image inpainting [1, 2, 17, 23, 24], image segmentation [10, 16, 21, 26, 27, 28, 36, 35] and surface fairing [15].

In the mean curvature model, the image is viewed as a parameterized surface \( \{(x, u(x)) : x \in \Omega \} \) in \( \mathbb{R}^3 \). The mean curvature \( \kappa \) of the surface at \( (x, u(x)) \) is given by

\[
\kappa(x) = \text{div} N(x),
\]

where

\[
N(x) = \frac{\nabla u(x)}{\sqrt{|\nabla u(x)|^2 + 1}}
\]

is the surface normal projected on the plane. Here, \( \nabla \) is the gradient operator and \( \text{div} \) is the divergence operator. Then, the mean curvature is used to regularize the reconstructed image. The objective function reads

\[
E_{MC}(u) := \frac{1}{2} \int_{\Omega} (u - f)^2 \, dx + \lambda \int_{\Omega} \Psi(\kappa) \, dx,
\]

where \( \Psi : \mathbb{R} \to \mathbb{R}_+ \) is defined as

\[
\Psi(x) = \begin{cases} 
  x^2, & |x| \leq 1 \\
  |x|, & |x| > 1
\end{cases}. 
\]

The potential function \( \Psi \) is defined so that the regularization is isotropic in smooth low-gradient regions and that the high curvature points are well preserved. It has been proved in [37] that this mean curvature model can preserve edges, intensity contrast, and corners when applied to some model data. Moreover, it does not have staircase artifacts, so that smooth regions can be reconstructed faithfully. Due to its ability to preserve contrast, high-contrast textures are also preserved. As a result, the mean curvature model often yields results of better visual quality compared
Newton’s Method for Curvature Based Denoising

to the TV model. However, like the TV model and most denoising algorithms, small-scale low-contrast textures may be lost because they can be difficult to be distinguished from noise. Some other interesting properties of the model are studied in [37, 38]. The objective function $E_{MC}$ is non-convex, so that multiple local minima exist. But this does not pose any problem in practice. By using iterative methods with the observed image as initial data, good results have been obtained.

The Euler-Lagrange equation is given by

$$\lambda \text{div} \left[ \frac{1}{\sqrt{\| \nabla u \|^2 + 1}} (I - P_x) \nabla (\Psi' (\kappa)) \right] + u - f = 0$$

in $\Omega$, together with the corresponding natural boundary conditions on the boundary $\partial \Omega$. Here, $I : \mathbb{R}^2 \to \mathbb{R}^2$ is the identity operator and $P_x : \mathbb{R}^2 \to \mathbb{R}^2$ is a rank-one operator given by

$$P_x (y) = \langle N(x), y \rangle N(x).$$

In [37], an effective explicit artificial time marching scheme is proposed to solve the Euler-Lagrange equation numerically.

The capabilities of the mean curvature model stem from its dependence on higher-order derivatives. This also adds some complexities into the optimization of the objective. The most prominent complication is that the Euler-Lagrange equation becomes a fourth-order nonlinear PDE whereas that of the TV model is only second-order. While standard gradient descent method with explicit time marching is very robust and easy to implement, it is restrained by very stringent stability conditions. Typically, the temporal step size $\Delta t$ has to be as small as $O(\Delta x^4)$, where $\Delta x$ is the spatial step size. This makes explicit schemes time consuming. Moreover, the objective function is non-convex so that solution to the Euler-Lagrange equation is not unique.

In this paper, we focus on the denoising of one-dimensional signals and propose a fast numerical method for solving the one-dimensional version of the Euler-Lagrange equation (3). By introducing some auxiliary variables, the terms in (3) that are stiff are effectively transformed to equations that are more amenable. The resulting system of nonlinear equations is solved by a modified Newton’s method. Based on Newton’s iteration, the method has the desirable local superlinear convergence. Unfortunately, due to the indefiniteness of the Jacobian matrix, the straightforward Newton’s method often fails to converge globally. To obtain global convergence, we propose a modification of the Jacobian matrix which results in very fast convergence and does not require the use of globalization techniques such as line search and trust-region method in practical settings.

2. Related work. In this section, we first review some existing methods for solving (2). Then, we review the primal-dual method for the TV model [9] by which our method is inspired.

2.1. Existing numerical methods. In the literature, a few approaches have been proposed for computing a solution to (2). They include a gradient descent method with explicit time marching [37], an Augmented Lagrangian method [38], a fixed point iteration scheme based on convex splitting [7], and a multigrid scheme with the aforementioned fixed point iteration as the smoother [7]. We describe these methods in more detail next.
Gradient Descent [37]: An artificial time is first introduced so that \( u = u(x, y, t) \). Then, the denoised image is given by the steady state of the gradient flow of the Euler-Lagrange equation (3):

\[
\frac{\partial u}{\partial t} = -\lambda \text{div} \left[ \frac{1}{\sqrt{|\nabla u|} + 1} (I - P_x \nabla (\Psi')) \right] - u + f.
\]

This fourth-order nonlinear PDE is then discretized by an explicit finite difference scheme. Denote the temporal and spatial step sizes by \( \Delta t \) and \( \Delta x \) respectively. The stability condition of the explicit scheme requires \( \Delta t \) to be as small as \( O(\Delta x^4) \), so that a large number of steps is needed.

Augmented Lagrangian method [38]: This optimization technique has been applied to a number of image processing problems such as total variation denoising [19, 33] and inpainting [32]. In this method, some auxiliary variables are introduced to make the objective function simple to handle. For the mean curvature model, these variables include \( p = (u_x, u_y, 1)^\top \), \( m = p/|p| \), \( n = m \) and \( q = \text{div} n \). Then, a saddle point of the Augmented Lagrangian function

\[
\mathcal{L}(u, q, n, m, p, \lambda_1, \lambda_2, \lambda_3, \lambda_4) = \frac{1}{2} \int_\Omega (u - f)^2 \, dx + \lambda \int_\Omega \Psi(q) \, dx
\]

is computed via an alternating minimization procedure in which the variables are optimized alternatively. Here, \( r_i > 0 \) are fixed constants and \( \lambda_1, \lambda_3 : \Omega \to \mathbb{R} \), \( \lambda_2, \lambda_4 : \Omega \to \mathbb{R}^3 \) are Lagrange multipliers. The term \( I(|m| \leq 1) \) is the indicator function to enforce the constraint \(|m| \leq 1 \). A nice feature of this method is that the optimization of \( \mathcal{L} \) with respect to each of its variables is a simple problem that can be done very quickly. Moreover, the outer alternating minimization procedure also converges fast. It also does not require differentiability of the function \( \Psi \).

Fixed point iteration with convex splitting [7]: The idea of convex splitting is to split the objective function into \( E_{\text{MC}} = E_1 - E_2 \) in such a way that \( E_1 \) is a convex function. Then the terms originated from \( E_1 \) are treated implicitly whereas those from \( E_2 \) are treated explicitly. In [7], the splitting is done to the Euler-Lagrange equation \( F = 0 \) such that \( F = F_1 - F_2 \) and \( F_1(u) = A(u)u \) for some positive definite operator \( A \). The splitting of (3) is given by

\[
F_1(u) = -\lambda \text{div} \left[ \frac{1}{\sqrt{|\nabla u|} + 1} P_x \nabla (\Psi') \right] - \gamma \text{div} \left( \frac{\nabla u}{\sqrt{|\nabla u|} + 1} \right) + u
\]

\[
F_2(u) = -\lambda \text{div} \left[ \frac{1}{\sqrt{|\nabla u|} + 1} \nabla (\Psi') \right] - \gamma \text{div} \left( \frac{\nabla u}{\sqrt{|\nabla u|} + 1} \right) + f.
\]
The parameter $\gamma > 0$ has to be large enough to bring in the necessary stability when $F_1$ is inverted. The fixed point iteration then reads

$$F_1(u^{n+1}) = F_2(u^n).$$

However, in [7], convergence results for (4) are not presented. A caveat with this method is that at each iteration, a complicated system of nonlinear equations has to be solved to advance one time step. In [7], the authors also presented a nonlinear multigrid method with the convex splitting scheme (4) as the smoother. In this case, (4) is only loosely solved by a few Gauss-Seidel or SOR iterations.

We remark that many numerical methods have been proposed for some related fourth-order problems. Of particular mention here is the minimization of Euler's elastica $\int_{\Omega} \Psi(\tilde{\kappa}) |\nabla u| \, dx$, or $\int_{\Omega} \Psi(\tilde{\kappa}) |\nabla u| \delta(u) \, dx$ (where $\tilde{\kappa} = \text{div}(\nabla u/|\nabla u|)$ and $\delta$ is the Dirac delta function), which has been applied to image segmentation, image inpainting and image denoising. Some numerical methods found in the literature include explicit time marching scheme (accelerated by scaling the Euler-Lagrange equation) [10], semi-implicit scheme [30], approximation with Modica-Mortola functional [16], multigrid and convex splitting [6], Augmented Lagrangian [32], graph-cut [4], and lagged curvature fixed point iteration [20]. While these methods have great potential to the model (2), we construct our method based on a primal-dual method which is reviewed next.

2.2. Primal-dual method by Chan, Golub, and Mulet. In the TV model (1), the Euler-Lagrange equation to solve is given by

$$-\lambda \text{div} \left( \frac{\nabla u}{|\nabla u|} \right) + u - f = 0.$$

Owing to the non-differentiability of the objective function at locations where $|\nabla u| = 0$, the Euler-Lagrange equation has singularities there. They make numerical solution difficult. This is the case especially for higher-order numerical methods such as Newton’s method because function approximation based on higher-order Taylor’s expansion becomes unreliable. Even if the standard trick of replacing $|\nabla u|$ with $|\nabla u|_\epsilon := \sqrt{|\nabla u|^2 + \epsilon}$ for some small $\epsilon > 0$ is used, the equation is still very stiff to solve. For instance, the domain of convergence of Newton’s method is extremely small, so that the method often fails to converge.

In [9], Chan, Golub, and Mulet proposed to introduce an auxiliary variable $p = \nabla u/|\nabla u|_\epsilon$ to remove the singularities. With this new variable, the Euler-Lagrange equation is reformulated to

$$\begin{cases} -\lambda \text{div} p + u - f = 0 \\ p|\nabla u|_\epsilon = \nabla u \end{cases}.$$ 

The troublesome term $|\nabla u|_\epsilon$ appeared in a denominator in (5) is now positioned in a numerator in (6). This system of equations is then solved by Newton’s method. The Newton’s system of linear equations to solve in each iteration is symmetrized. Local quadratic convergence and global convergence are observed. In the experiments, the method converges to a very high precision in just a few iterations. The linear system to solve in each Newton iteration can be solved efficiently using preconditioned conjugate gradient method.

The variable $p$ has an elegant interpretation. The Fenchel dual of (1) is given by

$$E^*_{\text{TV}}(p) = -\frac{1}{2} \int_{\Omega} |\lambda \text{div} p - f|^2 \, dx.$$
which is to be maximized over \( \{ p \in (C^1_0(\Omega)) : |p(x)| \leq 1 \} \). Due to the convexity of (1), duality theory states that the maximizer \( p^* \) of \( E^*_\text{TV} \) and the minimizer \( u^* \) of \( E_{\text{TV}} \) are related through (6) (with \( \epsilon = 0 \)). Therefore, the variable \( p = \nabla u / |\nabla u| \) is the Fenchel dual variable and the system (6) is a primal-dual system.

3. Proposed method.

3.1. Discrete 1D model. In one-dimension, we have \( u : \Omega \subset \mathbb{R} \to \mathbb{R} \) and

\[
\kappa = \left( \frac{u'}{\sqrt{(u')^2 + 1}} \right)'.
\]

Thus, \( \kappa \) is the signed curvature of the graph of \( u \), treated as a planar curve. We shall discretize the energy functional and present the Newton’s iteration in a finite dimensional setting. Let \( f_i \) for \( 1 \leq i \leq n \) be the given noisy signal. Let \( u_i \) for \( 1 \leq i \leq n \) be the restored signal. Using finite difference with a step size \( h \), the discrete curvature is

\[
\kappa_i = \begin{cases} 
\frac{\Delta_+ u_i}{h(\Delta_+ u_i)^2 + h^2} & \text{for } i = 1 \\
\frac{\Delta_+ u_i}{h(\Delta_+ u_i)^2 + h^2} - \frac{\Delta_+ u_{i-1}}{h(\Delta_+ u_{i-1})^2 + h^2} & \text{for } 2 \leq i \leq n - 1, \\
\frac{\Delta_+ u_{i-1}}{h(\Delta_+ u_{i-1})^2 + h^2} & \text{for } i = n
\end{cases}
\]

where \( \Delta_+ \) is the forward difference operator and we have imposed the boundary conditions \( \Delta_+ u_0 = \Delta_+ u_n = 0 \).

The discrete energy reads

\[
E_{\text{MC}}(u) = \frac{1}{2} \sum_{i=1}^{n} (u_i - f_i)^2 + \lambda \sum_{i=1}^{n} \Psi(\kappa_i).
\]

To make \( E_{\text{MC}} \) smooth, we adopt a smooth potential function \( \Psi \):

\[
\Psi(x) = \frac{2}{\pi} \left[ x \arctan \left( \frac{x}{\epsilon} \right) - \frac{\epsilon}{2} \ln \left( 1 + \frac{x^2}{\epsilon^2} \right) \right],
\]

where \( \epsilon > 0 \). This function is constructed such that \( \Psi'(x) = \frac{2}{\pi} \arctan(\frac{x}{\epsilon}) \) is a smooth approximation of the Heaviside function.

3.2. First-order optimality conditions. By differentiating (7), we obtain

\[
\begin{align*}
\frac{\partial \kappa_i}{\partial u_i} &= \begin{cases} 
\frac{h}{((\Delta_+ u_i)^2 + h^2)^{3/2}} & \text{for } i = 1 \\
-\frac{h}{((\Delta_+ u_i)^2 + h^2)^{3/2}} + \frac{h}{((\Delta_+ u_{i-1})^2 + h^2)^{3/2}} & \text{for } 2 \leq i \leq n - 1, \\
\frac{h}{((\Delta_+ u_{i-1})^2 + h^2)^{3/2}} & \text{for } i = n
\end{cases} \\
\frac{\partial \kappa_i}{\partial u_{i+1}} &= \frac{h}{((\Delta_+ u_i)^2 + h^2)^{3/2}} \text{ for } 1 \leq i \leq n - 1. \\
\frac{\partial \kappa_i}{\partial u_{i-1}} &= \frac{h}{((\Delta_+ u_{i-1})^2 + h^2)^{3/2}} \text{ for } 2 \leq i \leq n.
\end{align*}
\]
Hence, we obtain the first-order optimality conditions:

\[
0 = \frac{\partial E_{MC}}{\partial u_i} = u_i - f_i + \lambda \begin{cases} 
\frac{h(\Psi'_{i+1} - \Psi'_{i})}{((\Delta_+ u_i)^2 + h^2)^{3/2}} & \text{for } i = 1 \\
\frac{h(\Psi'_{i} - \Psi'_{i-1})}{((\Delta_+ u_i)^2 + h^2)^{3/2}} & \text{for } 2 \leq i \leq n - 1 \\
\frac{-h(\Psi'_{i} - \Psi'_{i-1})}{((\Delta_+ u_i - 1)^2 + h^2)^{3/2}} & \text{for } i = n
\end{cases}
\]

where \( \Psi'_i := \Psi'(\kappa_i) \).

Note that the \( \frac{\partial E_{MC}}{\partial u_i} \) computed here can also be obtained by discretizing the Euler-Lagrange equation (3) with standard finite differences under the boundary conditions \( \Psi'_0 = \Psi'_1 \) (i.e. \( \kappa_0 = \kappa_1 \)), \( \Psi'_{n+1} = \Psi'_n \) (i.e. \( \kappa_{n+1} = \kappa_n \)), \( u_0 = u_1 \) and \( u_{n+1} = u_n \). Putting it in another way, the above derivation gives us the proper numerical boundary conditions that make the solution minimize a discrete energy.

Let \( \Delta_+ \) be the backward difference operator. Using the conventions \( \Psi'_0 = \Psi'_1 \), \( \Psi'_{n+1} = \Psi'_n \), \( u_0 = u_1 \) and \( u_{n+1} = u_n \), we have

\[
0 = \frac{\partial E_{MC}}{\partial u_i} = u_i - f_i + \lambda \Delta_+ \left[ \frac{1}{((\Delta_+ u_i)^2 + h^2)^{3/2}} \Delta_+ \Psi'_i \right]
\]

(9)

for \( 1 \leq i \leq n \). Note that (9) is put into a form that resembles (3).

3.3. Proposed formulation. Parallel to the idea of the CGM method for the TV model, we introduce a variable

\[
w_i = \frac{\Delta_+ u_i}{\sqrt{(\Delta_+ u_i)^2 + h^2}}
\]

(10)

for \( 1 \leq i \leq n \). Then, by (7), we have

\[\kappa_i = \frac{1}{h} \Delta_+ w_i.\]

We also propose to introduce two more variables

\[
g_i = \Psi'(\frac{1}{2} \Delta_+ w_i)
\]

(11)

\[
v_i = \frac{(1 - w_i^2) \Delta_+ g_i}{\sqrt{(\Delta_+ u_i)^2 + h^2}}
\]

(12)

for \( 1 \leq i \leq n \).
By abusing notations, the optimality conditions (9) can be written as $F_j = 0$ for $j = 1, 2, 3, 4$, where

$$
\begin{align*}
F_1(u, v) &= u - f + \frac{\lambda}{h} \Delta_- v \\
F_2(u, w) &= \sqrt{(\Delta_+ u)^2 + h^2} w - \Delta_+ u \\
F_3(u, v, w, g) &= \sqrt{(\Delta_+ u)^2 + h^2} v - (1 - w^2) \Delta_+ g \\
F_4(w, g) &= g - \Psi'\left(\frac{1}{h} \Delta_+ w\right).
\end{align*}
$$

(13)

Here, $u, v, w, f, g \in \mathbb{R}^n$, $\Delta_+ = \begin{bmatrix} -1 & 1 \\ -1 & 1 \\ & \ddots & \ddots \\ & & -1 & 1 \\ & & & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}$ is the forward difference matrix under the reflective boundary condition and $\Delta_- = \begin{bmatrix} 1 \\ -1 & 1 \\ & \ddots & \ddots \\ & & -1 & 1 \\ & & & -1 & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}$ is the backward difference matrix under the zero boundary condition. The term $\sqrt{(\Delta_+ u)^2 + h^2}$ is interpreted as a diagonal matrix with diagonal entries $\sqrt{(\Delta_+ u_i)^2 + h_i^2}$. Likewise, $1 - w^2$ is a diagonal matrix with entries $1 - w_i^2$. The term $\Psi'\left(\frac{1}{h} \Delta_+ w\right)$ is a vector with entries $\Psi'\left(\frac{1}{h} \Delta_+ w_i\right)$. Note that the system (13) implies that $v_n = w_n = 0$.

### 3.4. Newton's iteration.

The Newton's update for (13) is given by the solution of the following linear system

$$
\begin{bmatrix}
\frac{\partial F_1}{\partial u} & 0 & 0 & 0 \\
0 & \frac{\partial F_1}{\partial v} & 0 & 0 \\
0 & 0 & \frac{\partial F_2}{\partial w} & 0 \\
0 & 0 & 0 & \frac{\partial F_3}{\partial g}
\end{bmatrix}
\begin{bmatrix}
\delta u \\
\delta v \\
\delta w \\
\delta g
\end{bmatrix}
= 
\begin{bmatrix}
-F_1 \\
-F_2 \\
-F_3 \\
-F_4
\end{bmatrix}.
$$

By differentiating (13), the Jacobian matrix reads

$$
\begin{pmatrix}
I & \frac{\lambda}{h} \Delta_- & 0 & 0 \\
\frac{\Delta_+ u}{\sqrt{(\Delta_+ u)^2 + h^2}} - 1 & \Delta_+ & \sqrt{(\Delta_+ u)^2 + h^2} & 0 \\
\frac{\Delta_+ u}{\sqrt{(\Delta_+ u)^2 + h^2}} & \sqrt{(\Delta_+ u)^2 + h^2} & 2w \Delta_+ g & (w^2 - 1) \Delta_+ \\
0 & 0 & -\frac{1}{h} \Psi'' \Delta_- & I
\end{pmatrix},
$$

where $\Psi'' = \Psi''\left(\frac{1}{h} \Delta_+ w\right)$. Next, we shall simplify the above linear system. Let $D = \sqrt{(\Delta_+ u)^2 + h^2}$. Note that $D$ is a diagonal matrix with positive entries so that
it is invertible. Thus, the augmented linear system can be written as

\[
\begin{bmatrix}
I & \frac{\lambda}{h} \Delta_- & 0 & 0 & -F_1 \\
\left(\frac{w\Delta u - 1}{D}\right) \Delta_+ & 0 & I & 0 & -F_2 \\
\frac{w\Delta u - 1}{D} \Delta_+ & I & \frac{2w\Delta + g}{D} & \frac{w^2 - 1}{D} \Delta_+ & -F_3 \\
0 & 0 & \frac{1}{h} \Psi'' \Delta_- & I & -F_4
\end{bmatrix}.
\]

Denote the \(i\)th block row of the above matrix by \(R_i\) for \(i = 1, 2, 3, 4\). By subtracting \(\frac{\lambda}{h} \Delta_+ \times \left[R_3 - \frac{w^2 - 1}{D} \Delta_+ \times R_4 - \left(\frac{w^2 - 1}{D} \Delta_+ \frac{1}{h} \Psi'' \Delta_- + \frac{2w\Delta + g}{D}\right) \times R_2\right]\) from \(R_1\), we obtain an equation for \(\delta u\):

\[
\left\{I - \frac{\lambda}{h} \Delta_- \left[\frac{v\Delta u}{D^2} \Delta_+ - \left(\frac{w^2 - 1}{D} \Delta_+ \frac{1}{h} \Psi'' \Delta_- + \frac{2w\Delta + g}{D}\right) \left(\frac{w\Delta u}{D^2} - \frac{1}{D}\right) \Delta_+\right]\right\} \delta u
\]

\[
= -F_1 + \frac{\lambda}{h} \Delta_- \left[\frac{F_3}{D} \frac{D}{w^2 - 1} \Delta_+ - F_4 - \left(\frac{w^2 - 1}{D} \Delta_+ \frac{1}{h} \Psi'' \Delta_- + \frac{2w\Delta + g}{D}\right) \frac{F_2}{D}\right].
\]

Thus,

\[
\left\{I - \frac{\lambda}{h} \Delta_- \left[\frac{v\Delta u}{D^2} + \frac{2w\Delta + g}{D} - \frac{2w^2 \Delta_+ u \Delta_+ g}{D^3} + \frac{w^2 - 1}{D} \Delta_+ \frac{1}{h} \Psi'' \Delta_- \left(\frac{1}{D} - \frac{w\Delta u}{D^2}\right) \Delta_+\right]\right\} \delta u
\]

\[
= -F_1 + \frac{\lambda}{h} \Delta_- \left[\frac{F_3}{D} \frac{D}{w^2 - 1} \Delta_+ - F_4 - \left(\frac{w^2 - 1}{D} \Delta_+ \frac{1}{h} \Psi'' \Delta_- + \frac{2w\Delta + g}{D}\right) \frac{F_2}{D}\right].
\]

Let us introduce some notations to simplify the notation. We define

\[
\begin{align*}
A_1 &= \frac{v\Delta u}{D^2} + \frac{2w\Delta + g}{D} - \frac{2w^2 \Delta_+ u \Delta_+ g}{D^3} \\
A_2 &= 1 - \frac{w^2}{D} \\
A_3 &= \frac{1}{h} \Psi'' \\
A_4 &= \frac{1}{D} - \frac{w\Delta u}{D^2} \\
A &= I - \frac{\lambda}{h} \Delta_- (A_1 - A_2 \Delta_+ A_3 \Delta_- A_4) \Delta_+ \\
b &= \frac{F_3}{D} + A_2 \Delta_+ F_4 - \left(-A_2 \Delta_+ A_3 \Delta_- + \frac{2w\Delta + g}{D}\right) \frac{F_2}{D}
\end{align*}
\]

Then, the equation for \(\delta u\) is given by

\[
(15) \quad A \delta u = -F_1 + \frac{\lambda}{h} \Delta_- b.
\]
Once the linear system is solved to obtain $\delta u$, we can compute other unknowns using (14). More precisely, we have

\begin{align}
\delta w &= A_4 \Delta_+ \delta u - \frac{F_2}{D} \\
\delta g &= A_3 \Delta_- \delta w - F_4 \\
\delta v &= -\frac{v \Delta_+ u}{D^2} \Delta_+ \delta u - \frac{2w \Delta_+ g}{D} \delta w + A_2 \Delta_+ \delta g - \frac{F_3}{D}.
\end{align}

### 3.5. Modified Newton system

Due to the non-convexity of the objective function $E_{MC}$, the linear system (15) can be indefinite. In practice, the Newton’s iteration often diverges (see Figure 5). To remedy this defect, we propose to modify the system. Note that $A_1$ is a diagonal matrix. Let

$$\hat{A}_1 = \max\{A_1, 0\}$$

to be the positive part of $A_1$ and let

$$A_5 = \frac{1}{2}(A_2 + A_4).$$

Let

$$\tilde{A} = I - \frac{\lambda}{h} \Delta_- (\hat{A}_1 - A_5 \Delta_+ A_3 \Delta_- A_5) \Delta_+.$$  

Then, we replace (15) by the following system:

\begin{align}
\tilde{A} \delta u &= -F_1 + \frac{\lambda}{h} \Delta_- b.
\end{align}

Note that $\Delta_- = -\Delta_+^\top$. By the strict convexity of $\Psi$, we have that $A_3$ is a positive diagonal matrix. Then, it is straightforward to verify that the matrix $\tilde{A}$ is symmetric positive definite.

To shed some light on how this method works, consider the case where the auxiliary variables $v, w, g$ are near convergence so that $\|F_2\|, \|F_3\|, \|F_4\| \ll \|F_1\|$ so that $\|\frac{\lambda}{h} \Delta_- b\| \ll \|F_1\|$. Moreover, by (9) and (13), we have

$$F_1 = u - f + \frac{\lambda}{h} \Delta_- v \approx \frac{\partial E_{MC}}{\partial u}.$$  

Thus,

$$\delta u = -\tilde{A}^{-1} F_1 + \frac{\lambda}{h} \tilde{A}^{-1} \Delta_- b \approx -\tilde{A}^{-1} \frac{\partial E_{MC}}{\partial u}.$$  

Since $\tilde{A}$ is symmetric positive definite, we have that $\delta u$ is a descending direction. Hence, as long as $\|F_2\|, \|F_3\|, \|F_4\| \ll \|F_1\|$, global convergence is favored.

### 4. Numerical results

In this section, we evaluate the effectiveness of the proposed method by comparing it with the gradient descent method in [37] and the augmented Lagrangian method in [38] numerically and by testing it in various parameter settings. The focus of the tests is the speed and robustness of the algorithm. For assessment of the quality of the denoised signals, we refer the readers to [37].

All computations are done using Matlab R2011b on an Intel Pentium D machine.
4.1. Testing method and parameter settings.

Data and model. Three original clean signals are used. They are defined by

- **Signal A:**
  
  \[
  u_{\text{clean}}(x) = \begin{cases} 
  0.2 & \text{for } 0 \leq x \leq 0.2 \\
  x & \text{for } 0.2 < x \leq 0.4 \\
  4(x - 0.5)^2 + 0.5 & \text{for } 0.4 < x \leq 0.8 \\
  0.5 & \text{for } 0.8 < x \leq 1 
  \end{cases}
  \]

- **Signal B:**
  
  \[u_{\text{clean}}(x) = \sin(4\pi x)\]

- **Signal C:**
  
  \[
  u_{\text{clean}}(x) = \begin{cases} 
  0 & \text{for } x \in [0, 0.1] \cup (0.9, 1] \\
  0.5 & \text{for } x \in (0.1, 0.2] \cup (0.3, 0.4] \cup (0.5, 0.6] \cup (0.7, 0.8] \\
  -0.5 & \text{for } x \in (0.2, 0.3] \cup (0.4, 0.5] \cup (0.6, 0.7] \cup (0.8, 0.9] 
  \end{cases}
  \]

These data are chosen to contain various characteristics. Signal A contains piecewise constant, piecewise linear, piecewise quadratic, and jump features. Signal B is a smooth signal. Signal C contains the largest number of jumps. These data are depicted in Figure 1, where the signal size is \( n = 2^{12} \). In the figure, we also show some examples of noisy signals with signal-to-noise (SNR) ratio 30dB and some denoised results using an optimal choice of \( \lambda \) (see below for the definition of optimal \( \lambda \)). Notice that under the curvature based model, jumps are preserved and staircase artifacts are absent.

**Figure 1.** Three sets of original signals (\( n = 2^{12} \)), noisy signals (SNR= 30dB) and denoised signals (optimal \( \lambda \)).
Table 1. Default parameters of data and model.

| $n$    | $2^{12}$ |
|--------|----------|
| SNR    | 30dB     |
| $\lambda$ | optimal |
| $h$    | 1        |
| $\epsilon$ | 1       |

Table 2. Optimal $\lambda$ (two significant digits) for various SNRs. In each entry, parameters not shown are set to their defaults indicated in Table 1.

| Optimal $\lambda$ | SNR (dB) |
|-------------------|----------|
|                   | 40       |
| Signal A          | $5.0 \times 10^{4}$  |
|                   | 30       |
| Signal B          | $4.0 \times 10^{5}$  |
|                   | 20       |
| Signal C          | $2.5 \times 10^{12}$ |

Table 3. Optimal $\lambda$ (two significant digits) for various signal sizes $n$. In each entry, parameters not shown are set to their defaults indicated in Table 1.

| Optimal $\lambda$ | $2^{8}$  |
|-------------------|----------|
|                   | $2^{10}$ |
| Signal A          | $6.0 \times 10^{2}$  |
|                   | $2^{12}$ |
| Signal B          | $4.0 \times 10^{3}$  |
|                   | $2^{14}$ |
| Signal C          | $1.7 \times 10^{4}$  |
|                   | $2^{16}$ |

The default values of the parameters of the data and model are listed in Table 1. But in some experiments, we test the method by varying some of the parameters. The optimal $\lambda$ is chosen (with two significant digits) to minimize the $l_1$ error:

$$||u_{\text{clean}} - u||_1 = \frac{1}{n} \sum_{i=1}^{n} |u_{\text{clean},i} - u_i|.$$  

Such an optimal $\lambda$ is recomputed for each observed signal, which uses a different clean signal, $n$, SNR and noise. These values are reported in Tables 2 and 3. It is well known that such a choice does not necessarily optimize visual quality. For example, in Figure 1, the denoised Signal C still contains some noise. Using a larger $\lambda$ can further reduce these oscillations, at the expense of an increased $l_1$ error. But our aim is to test the proposed method under a large number of parameters. Thus, we resort to a way of determining $\lambda$ that can be automated. We have also tried to use $l_2$ and $l_\infty$ norms to measure errors, but the visual quality of the denoised signals worsens.

Parameters of Newton’s method. The stopping criterion of the Newton’s iteration that we adopt is a very stringent one:

$$||\delta u||_\infty = \max_{1 \leq i \leq n} |\delta u_i| \leq 10^{-15}.$$
Table 4. Optimal time steps for the gradient descend method (one significant digit).

| Signal | SNR (dB) | Optimal $\Delta t$ |
|--------|----------|---------------------|
| A      | $10^{-4}$ | $10^{-5}$ $10^{-6}$ |
| B      | $10^{-6}$ | $10^{-7}$ $10^{-8}$ |
| C      | $10^{-2}$ | $10^{-3}$ $10^{-4}$ |

Except in some extreme parameter settings, the proposed method usually converges in a few iterations. The maximum number of iterations is set to 30.

The initial guess of $u$ is the noisy signal $f$. The initial guess of the other variables $v, w, g$ are chosen according to (12), (10), and (11), respectively.

The linear system (20) to solve in each Newton’s iteration is pentadiagonal and can be solved efficiently by the Matlab backslash function.

To assess the performance of the proposed method, we report the energy (8) vs CPU time (in seconds) and the residual norm $\|F\|_{\infty}$ vs CPU time (in seconds). Here, $F = 0$ represents the system that the method is solving, i.e. $F : (\mathbb{R}^n)^4 \to (\mathbb{R}^n)^4$
where

$$F(u, v, w, g) = \begin{bmatrix} F_1(u, v) \\ F_2(u, w) \\ F_3(u, v, w, g) \\ F_4(w, g) \end{bmatrix},$$

where $F_j$ is defined by (13).

**Parameters of gradient descent method.** The maximum number of iterations is set to 10000. The time step is tuned on individual basis to obtain the best timings and ensure convergence. The time steps are shown in Table 4. The initial guess is $u = f$.

The gradient descend method directly solves the optimality condition (9). Thus, the residual $F_{GD} : \mathbb{R}^n \to \mathbb{R}^n$ is defined by

$$F_{GD}(u) = u - f + \lambda h \Delta - \left[ \frac{1}{((\Delta^t u)^2 + h^2)^{3/2}} \Delta + \Psi' \right].$$

We report the normalized $l_1$ residual norm

$$\|F_{GD}\|_1 = \frac{1}{n} \sum_{i=1}^{n} |F_{GD,i}|$$

since the convergence w.r.t. $l_\infty$ norm is too slow. But we insist to use the $l_\infty$ norm to the proposed method to highlight its fast and uniform convergence.

We remark that, theoretically, if $F(u^*, v^*, w^*, g^*) = 0$, then we also have $F_{GD}(u^*) = 0$. Thus one may think of using $\|F_{GD}(u)\|_{\infty}$ as well to measure the merit of the proposed method so that a common measure is used across different methods. But in practice we found that although our method can achieve $\|F(u, v, w, g)\|_{\infty} \leq 10^{-15}$ after some iterations, we only have $F_{GD}(u) = O(10^{-15}(\lambda + 1))$. Thus the residual norm $\|F_{GD}(u)\|_{\infty}$ cannot reach the level of $10^{-15}$ when a large $\lambda$ is used. Hence, we define the residual to be the system that each method directly deals with.
Parameters of augmented Lagrangian method. The maximum number of iterations is set to 1000. The parameters $r_1, r_2, r_3, r_4$ are selected manually as follows. For each $(r_1, r_2, r_3, r_4) \in \{1, 10^2, 10^4, 10^6\}^4$, we run the method for 500 iterations and record the final energy and residual norm. Then, the parameters are selected so that both energy and residual norm are as small as possible. The selected parameters are shown in Table 5.

| $\{r_1, r_2, r_3, r_4\}$ | 40         | 30         | 20         |
|--------------------------|------------|------------|------------|
| Signal A                 | $(10^2, 10^2, 10^4, 10^2)$ | $(10^2, 10^4, 10^4)$ | $(10^3, 10^5, 10^6)$ |
| Signal B                 | $(10^4, 10^4, 10^6, 10^4)$ | $(10^4, 10^6, 10^4)$ | $(10^5, 10^6, 10^4)$ |
| Signal C                 | $(1, 1, 1, 10^2)$ | $(10^2, 10^2, 10^4, 10^2)$ | $(10^4, 10^4, 10^6, 10^4)$ |

Table 5. Manually selected parameters for the augmented Lagrangian method.

The residual in this case is defined as

$$F_{AL}(u, q, n, m, p) = \begin{bmatrix} |p| - \langle p, m \rangle \\ p_1 - \frac{1}{\kappa} \Delta_+ u \\ p_2 - 1 \\ q - \frac{1}{\kappa} \Delta_- n_1 \\ n_1 - m_1 \\ n_2 - m_2 \end{bmatrix}. $$

We report the normalized $l_1$ residual norm

$$\|F_{AL}\|_1 = \frac{1}{6n} \sum_{i=1}^{6n} |F_{AL,i}|.$$
Newton’s Method for Curvature Based Denoising

Third, for Signal A & C, both Newton, GD and AL reach a similar energy level but Newton uses much less time in most cases (recall that the times are plotted in log scale). For Signal B, GD stops at a higher energy level (about 10 times larger when SNR=30dB and about 100 times larger when SNR=20dB).

Fourth, besides the obvious case of GD, Newton also features a monotonically decreasing energy.

Fifth, while the profiles of the residual norm of Newton and AL are very stable across different noise levels, it is less so for GD. For example, for Signal A, GD’s residual norm at the 10000th iteration increases from about $10^{-4}$ to $10^{-2}$ and $10^{0}$ as the SNR changes from 40dB to 30dB and 20dB, respectively. This can be attributed to the time steps shown in Table 4. As noise level increases (SNR decreases), a smaller time step is needed to facilitate convergence. This in turn makes convergence slower.

4.3. Effects of the signal size \( n \). In this test, we investigate how the method depends on the signal size \( n \). For each \( n \in \{2^{8}, 2^{10}, 2^{12}, 2^{14}, 2^{16}\} \), we run the proposed method and the residual norm \( \|F\|_{\infty} \) during the iteration is plotted against the CPU time (in seconds), see Figure 5. To demonstrate the need of modifying the Newton’s system, we report the results using the original Newton’s iteration as well.

For the modified Newton’s iteration (20) and (16)–(18), fast convergence is observed for all values of \( n \). For the original Newton’s iteration (15)–(18), divergence is observed for larger values of \( n \).

We also observe that even when \( n \) is as large as 2^{16}, the residual norm resulted from the proposed method becomes less than 10^{-15} in less than 4 seconds. The number of Newton’s iterations is quite stable for all \( n \).

For illustration purpose, we show the condition number of the modified Newton’s system in Figure 6. The condition number does not vary at all during the iteration. As expected, it increases with the signal size.

4.4. Effects of the regularization parameter \( \lambda \). In this test, we study how the method performs under a wide range of regularization parameter \( \lambda \). For each \( \lambda \in \{10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}, 10^{6}, 10^{8}, 10^{10}\} \), we run the proposed method and the residual norm \( \|F\|_{\infty} \) during the iteration is plotted against the CPU time (in seconds), see Figure 7.

In Figure 7, fast convergence is observed for all but one case. The time to convergence also does not vary too much. The only case where divergence is observed is Signal C with \( \lambda = 10^{6} \). The optimal \( \lambda \) in this case is 4.0. Thus the divergence occurred at a value much larger than the practical value. Indeed, if \( \lambda \) is very large, then the energy is dominated by the regularization term and the problem becomes more ill-posed due to the non-uniqueness of minimizer to the regularization term. This is also reflected in the Newton’s system (14) whose limiting system is singular. Thus we expect the performance can deteriorate as \( \lambda \) increases.

4.5. Line search. In Figure 7, divergence is observed for Signal C with \( \lambda = 10^{6} \). However, thanks to the modification of the Newton’s system, the computed direction \( \delta u \) is a descending direction under very mild conditions. Thus, global convergence is encouraged with the help of line search. The aim of this experiment is to test the performance of line search.

The line search procedure we adopt is as follows.
1. Compute $\delta u$ as usual
2. $\alpha_u \leftarrow 1$
3. While $E_{MC}(u + \alpha_u \delta u) > E_{MC}(u)$, $\alpha_u \leftarrow 0.5 \alpha_u$
4. $\delta u \leftarrow \alpha_u \delta u$
5. Compute $\delta v$, $\delta w$, $\delta g$ as usual

In Figure 8, we show the result of Newton with line search. The data is Signal C with $\lambda = 10^6$, which is the only case where divergence is observed in the previous experiments. Convergence is reached after eleven iterations. The timing is also comparable to those in Figure 7, where line search is not utilized. The step sizes $\alpha_u$ obtained are shown in Figure 8. The step size is reduced to 0.5 only in three of the iterations.
We remark that this line search procedure will not reduce the step size in all other cases conducted in the previous experiments because there the energy is monotonically decreasing.

5. Conclusion. Based on the Chan-Golub-Mulet idea, we reformulated the curvature based denoising model into a more amenable nonlinear system of equations. We also introduced a modified Newton’s method to solve the system. We demonstrated that the modification is essential and that the method exhibits very fast convergence in a wide range of parameters. Only when some extreme values of $\lambda$ are used, a line search is needed to obtain global convergence. The proposed method does not require any parameter tuning whereas gradient descent and augmented Lagrangian

Figure 3. Comparison with gradient descend (GD) and augmented Lagrangian (AL) methods. SNR=30dB.
methods do. Another advantage of the proposed method is that the accuracy is very high. This makes convergence very easy to detect. When using a slow numerical method such as gradient descent, sometimes the stopping criterion is met merely because the progress is too slow that $\delta u$ becomes tiny; in this case a prolonged run of the iteration can eventually result in a significant change of the restored signal.

The curvature model has some nice features such as contrast and corner preservation which are not shared by the seemingly alike Euler’s elastica model. Moreover, the presence of the constant 1 in the curvature term $\left(\frac{u'}{\sqrt{(u')^2+1}}\right)'$ makes the Euler-Lagrange equation somehow easier to solve. The method could break down if the constant 1 is replaced by a smaller value. This makes the curvature model valuable.
Figure 5. Effects of the signal size $n$.

In a future work, we shall investigate the generalization to the two-dimensional case, where more sophisticated ways to modify the Newton’s system and its preconditioning are called.

Acknowledgments. We would like to thank the anonymous reviewers for their comments and suggestions.

REFERENCES

[1] L. Ambrosio and S. Masnou, A direct variational approach to a problem arising in image reconstruction, Interfaces and Free Boundaries, 5 (2003), 63–81.
[2] L. Ambrosio and S. Masnou, On a variational problem arising in image reconstruction, In “Proc. Free Boundary Problems,” volume 147 of Internat. Series of Num. Math., Basel, 2004. Birkhauser, 17–26.
Figure 6. Condition number of the modified Newton’s system during the iteration.

[3] G. Aubert and P. Kornprobst, “Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations,” Second edition. With a foreword by Olivier Faugeras. Applied Mathematical Sciences, 147. Springer, New York, 2006.

[4] E. Bae, J. Shi, and X. C. Tai, Graph cuts for curvature based image denoising, IEEE Trans. Image Process, 20(2011), 1199–1210.

[5] P. Blomgren, P. Mulet, T. Chan and C. Wong, Total variation image restoration: Numerical methods and extensions, In “ICIP,” pages 384–387, Santa Barbara, 1997.

[6] C. Brito and K. Chen, Fast numerical algorithms for Euler’s elastica inpainting model, Int. J. Modern Math., 5 (2010), 157–182.

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

[8] A. Chambolle and P. Lions, Image recovery via total variation minimization and related problems, Numer. Math., 76 (1997), 167–188.

[9] T. Chan, G. Golub and P. Mulet, A nonlinear primal-dual method for total variation-based image restoration, SIAM J. Sci. Comp., 20 (1999), 1964–1977.
Figure 7. Effects of the regularization parameter $\lambda$.

[10] T. Chan, S. Kang and J. Shen, *Euler’s elastica and curvature-based image inpainting*, SIAM J. Appl. Math., 63 (2002), 564–592.

[11] T. Chan, M. Marquina and P. Mulet, *High-order total variation-based image restoration*, SIAM J. Sci. Comput., 22 (2000), 503–516.
Figure 8. Using line search to guarantee global convergence (Signal C, $\lambda = 10^6$).

[12] T. F. Chan and J. Shen, “Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods,” Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.

[13] T. F. Chan, S. Esedoḡlu and F. Park, A fourth order dual method for staircase reduction in texture extraction and image restoration problems, In “Proceedings of 2010 IEEE 17th Int. Conf. on Image Process,” pages 4137–4140, Hong Kong, 2010.

[14] T. F. Chan, S. Esedoḡlu, F. E. Park and A. M. Yip, Recent developments in total variation image restoration, In Nikos Paragios, Yunmei Chen, and Olivier Faugeras, Editors, Handbook of Mathematical Models in Computer Vision, pages 17–31. Springer, Berlin, 2005.

[15] M. Elsey and S. Esedoḡlu, Analogue of the total variation denoising model in the context of geometry processing, SIAM J. Imaging Sci., 7 (2009), 1549–1573.

[16] S. Esedoḡlu and R. March, Segmentation with depth but without detecting junctions, Special issue on imaging science (Boston, MA, 2002). J. Math. Imaging and Vision, 18 (2003), 7–15.

[17] S. Esedoḡlu and J. Shen, Digital inpainting based on the Mumford-Shah-Euler image model, European J. Appl. Math., 13 (2002), 353–370.

[18] J. B. Greer and A. L. Bertozzi, Traveling wave solutions of fourth order PDEs for image processing, SIAM J. Math. Anal., 36 (2004), 38–68.

[19] K. Ito and K. Kunisch, An active set strategy based on the augmented Lagrangian formulation for image restoration, RAIRO Math. Mod. and Num. Analysis, 33 (1999), 1–21.
Newton's Method for Curvature Based Denoising

[20] T. S. Lau and A. M. Yip, *A fast method to segment images with additive intensity value*, SIAM J. Imaging Sci., 5 (2012), 963–1021.

[21] Y. N. Law, H. K. Lee, C. Liu and A. M. Yip, *A variational model for segmentation of overlapping objects with additive intensity value*, IEEE Trans. Image Process., 20 (2011), 1495–1503.

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

[23] S. Masnou, *Disocclusion: A variational approach using level lines*, IEEE Trans. Image Process., 11 (2002), 68–76.

[24] S. Masnou and J. M. Morel, *Level lines based disocclusion*, In “Proc. IEEE Int. Conf. on Image Process.,” pages 259–263, Chicago, IL, 1998.

[25] Y. Meyer, “Oscillating Patterns in Image Processing and Nonlinear Evolution Equations,” The fifteenth Dean Jacqueline B. Lewis memorial lectures. University Lecture Series, 22. AMS, Providence, Rhode Island, 2001.

[26] D. Mumford, *Elastica and computer vision*, Algebraic Geometry and its Applications, 491–506, Springer, New York, (1994).

[27] M. Nitzberg and D. Mumford, *The 2.1D sketch*, In “Proc. of the 3rd Intl. Conf. on Computer Vision,” pages 138–144, 1990.

[28] M. Nitzberg, D. Mumford and T. Shiota, “Filtering, Segmentation, and Depth,” Lecture Notes in Computer Science, 662. Springer-Verlag, New York, 1993.

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

[30] P. Smereka, *Semi-implicit level set methods for curvature and surface diffusion motion*, J. Sci. Compt., 19 (2003), 439–456.

[31] D. M. Strong, P. Blomgren and T. F. Chan, *Spatially adaptive local feature-driven total variation minimizing image restoration*, In “Proceedings of The International Society of Photographic Instrumentation Engineers: Statistical and Stochastic Methods in Image Processing II,” 3167 (1997), 222–233.

[32] X. C. Tai, J. Hahn, and G. J. Chung, *A fast algorithm for Euler's elastica model using augmented Lagrangian method*, SIAM J. Imaging Sci., 4 (2011), 313–344.

[33] C. Wu and X. C. Tai, *Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and higher order models*, SIAM J. Imaging Sci., 3 (2010), 300–339.

[34] Y. L. You and M. Kaveh, *Fourth-order partial differential equations for noise removal*, IEEE Trans. Image Process., 9 (2000), 1723–1730.

[35] W. Zhu and T. Chan, *A variational model for capturing illusory contours using curvature*, J. Math. Imaging. Vis., 27 (2007), 29–40.

[36] W. Zhu, T. Chan and S. Esedoḡlu, *Segmentation with depth: A level set approach*, SIAM J. Sci. Comput., 28 (2006), 1957–1973.

[37] W. Zhu and T. F. Chan, *Image denoising using mean curvature of image surface*, SIAM J. Imaging Sci., 5 (2012), 1–32.

[38] W. Zhu, X. C. Tai and T. F. Chan, *Augmented Lagrangian method for a mean curvature based image denoising model*, UCLA CAM Report, 12-02, 2012.

Received June 2012; revised December 2012.

E-mail address: andyyip@nus.edu.sg
E-mail address: wzhu7@bama.ua.edu