Journal of Computational Mathematics
Vol.30, No.2, 2012, 177–196.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1107-m3423

HOMOTOPY CURVE TRACKING FOR TOTAL VARIATION IMAGE RESTORATION*

Fenlin Yang
School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
Email: yangfenlinlin@126.com
Ke Chen
Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, The University of Liverpool, Liverpool L69 7ZL, United Kingdom
Email: k.chen@liv.ac.uk
Bo Yu
School of Mathematical Sciences, Dalian University of Technology, Dalian 116024, China
Email: yubo@dlut.edu.cn

Abstract

The total variation (TV) minimization problem is widely studied in image restoration. Although many alternative methods have been proposed for its solution, the Newton method remains not usable for the primal formulation due to no convergence. A previous study by Chan, Zhou and Chan [15] considered a regularization parameter continuation idea to increase the domain of convergence of the Newton method with some success but no robust parameter selection schemes. In this paper, we consider a homotopy method for the same primal TV formulation and propose to use curve tracking to select the regularization parameter adaptively. It turns out that this idea helps to improve substantially the previous work in efficiently solving the TV Euler-Lagrange equation. The same idea is also considered for the two other methods as well as the deblurring problem, again with improvements obtained. Numerical experiments show that our new methods are robust and fast for image restoration, even for images with large noisy-to-signal ratio.

Mathematics subject classification: 65N06, 65B99.
Key words: Image restoration, Total variation, Newton method, Homotopy method, Correction and curve tracking.

1. Introduction

It is well known that an image can often become blurry and noisy if corrupted during formation, transmission or recording process. This degradation makes it difficult to do further image processing tasks such as edge detection, pattern recognition, and object tracking, etc. Denote by z the observed image (known) and u the desired true image (unknown), both defined on a bounded convex region Ω of $\mathbb{R}^d$ (for simplicity we will assume Ω to be a square in $\mathbb{R}^2$). Consider the common degradation model

$$z = Ku + \eta, \tag{1.1}$$

where η is an additive noise term (also unknown) and K is a known linear operator representing the blur (usually a convolution), the image is only corrupted by noise when K is the identity. We wish to reconstruct the true image u from the observed image z.

* Received June 2, 2010 / Revised version received June 20, 2011 / Accepted June 24, 2011 / Published online February 24, 2012 /

There are many different modeling methods proposed to obtain an estimate of u [28]. One effective and well-known method is the total variation-based method by Rudin, Osher and Fatemi [26], consisting of solving the following constrained optimization problem:

$$\min_u \int_\Omega |\nabla u| dxdy \quad \text{subject to} \quad \|Ku - z\|^2 = \sigma^2. \tag{1.2}$$

Here $|\cdot|$ is the Euclidean norm in $\mathbb{R}^2$, $\|\cdot\|$ is the norm in $\mathbf{L}^2(\Omega)$ and σ is the standard deviation of the noise η. This problem is naturally linked to the following unconstrained problem – the minimization of the total variation penalized least squares functional (see [8, 26, 28]):

$$\alpha \int_\Omega |\nabla u| dxdy + \frac{1}{2}\|Ku - z\|^2, \tag{1.3}$$

where α is a positive parameter controlling the trade-off between goodness of fit-to-the-data and variability in u. The main advantage of the total variation restoration models is that their solutions preserve edges very well. But other models without the TV are also effective [5,13,14].

In spite of the fact that the variational problem (1.3) is convex, the computation is not easy since the total variation semi-norm is a nonlinear nondifferentiable functional. To overcome the nondifferentiation difficulty, one approach is the dual methods (see [3, 7]) and the other is a split Bregman iteration [33]. However, the commonly used technique is to approximate the term $|\nabla u|$ by $\sqrt{|\nabla u|^2 + \beta}$, where β is a small positive parameter, and the unconstrained minimization problem (1.3) becomes

$$\min_u \left\{ f(u) = \alpha \int_\Omega \sqrt{|\nabla u|^2 + \beta} dxdy + \frac{1}{2}\|Ku - z\|^2 \right\}. \tag{1.4}$$

It is shown in [1] that the solution of (1.4) converges to the solution of (1.3) when $\beta \to 0$. The corresponding Euler-Lagrange partial differential equation (PDE) for (1.4) is

$$g(u) = -\alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + K^*(Ku - z) = 0, \qquad (x, y) \in \Omega, \tag{1.5}$$

with homogeneous Neumann boundary condition $\partial u / \partial \vec{n} = 0$, $(x, y) \in \partial\Omega$. Here $\nabla\cdot$ is the divergence operator, K^* is the adjoint operator of K with respect to the $\mathbb{L}^2$ inner product, $\partial\Omega$ is the boundary of Ω and $\vec{n}$ is the normal vector of $\partial\Omega$. It should be remarked that even for moderately small β, the Newton method does not converge with the common starting iterate $u = z$; therefore, one cannot find any use of Newton type methods for this primal equation in the literature.

Before we present a method to help the Newton method, we briefly review four categories of methods for solving (1.5).

1) **Gradient descent methods** [24, 26]. As used in Rudin *et al.* [26], instead of the elliptic PDE, a parabolic PDE with time as an evolution parameter is solved by the gradient descent method

$$u_t = \mathcal{N}(u) \equiv \alpha \nabla \cdot \Big(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \Big) - K^*(Ku - z), \quad u(x, y, 0) = z. \tag{1.6}$$

An accelerated version is $u_t = |\nabla u| \mathcal{N}(u)$ as first used by [24]. This method is preferred in many situations for its simplicity, user-independent choice of regularize parameter and fast

initial convergence, but it converges very slowly to its steady state since the parabolic term is nearly singular for small gradients. Yet an even faster version is via the additive operator splitting idea [21, 32] for (1.6) which is used on a semi-implicit time-marching scheme as widely used.

2) **Fixed-point methods** [14, 29]. The idea is to solve the Euler-Lagrange equation (1.5) directly by linearizing the highly nonlinear and ill-conditioned term $\nabla\cdot(\nabla u(\sqrt{|\nabla u|^2+\beta})^{-1})$. In details, solve

$$-\alpha\nabla\cdot\left(\frac{\nabla u^{k+1}}{\sqrt{|\nabla u^k|^2+\beta}}\right)+K^*(Ku^{k+1}-z)=0,\quad k=0,1,\cdots.$$

Here $u^0=z$, and at each iteration one must solve a linear diffusion equation, whose diffusivity depends on the previous iterate u^k. Such a linear equation is often solved by a preconditioned conjugate gradient method [28] or a multigrid method, especially the robust algebraic multigrid methods. The fixed point method is also used in defining a smoother for a nonlinear multigrid method [27].

3) **Interior-point primal-dual method** [11]. Instead of solving (1.5) directly for the primal variable u, the main idea is to introduce a new (dual) variable $w=\nabla u(\sqrt{|\nabla u|^2+\beta})^{-1}$ and replace (1.5) by the following system of nonlinear partial differential equations

$$\begin{aligned}-\alpha\nabla\cdot w+K^*(Ku-z)&=0,\\ w\sqrt{|\nabla u|^2+\beta}-\nabla u&=0.\end{aligned}$$

Practically, the system can be effectively solved by a Newton method but the smaller the β is, the higher the quality of the reconstruction of image edges and the slower the convergence of the Newton iterations (and the inner iterations) becomes [11, 16, 28].

4) **The dual methods** [7]. Instead of solving (1.5) directly for the primal variable u, we first reformulate it into an equation involving the dual variable only [7]. Then the gradient descent methods can be used for the resulting dual equation which can be shown to be globally convergent [7]. Further the dual equation can be solved by a non-smooth Newton method [23] or other iterative methods [10].

Returning to the main theme of the paper, we note that the Newton method does not work satisfactorily in the sense that its domain of convergence is very small which requires an initial guess (better than z and) close to the true solution. Chan, Zhou and Chan [15] proposed a continuation procedure for Newton method on the parameters α and β to improve the initial guess in each continuation. After selecting suitable large values of α_1 and β_1, this procedure consists of two steps: fix β_1 and gradually increase α_1 to the given value α and fix α and decrease β_1 towards β step by step. Unfortunately it is not easy to find an automatic way for the continuation to be robust. Independent of the above work, Melara *et al.* [22] used the Newton method combined with a globalization technique (i.e. line search for augmented Lagrangian method) for this TV denoising problem; it turned out the parameter homopoty for gradually decreasing β by a fixed ratio is almost identical to Chan, Zhou and Chan [15].

As a globally convergent method, the homotopy method has versatility and robustness, and it has become an important tool for solving nonlinear problems, see [2, 12, 17, 18, 20, 31]. In this paper, we propose a more robust and efficient homotopy method to solve (1.5) directly. The

basic homotopy algorithm to solve this problem is to construct a continuous map $H(u,t)$ which deforms a simple function $g_0(u) = H(u,0)$ to the given function $g(u) = H(u,1)$ as t varies from 0 to 1. The continuous map is defined by

$$H(u,t) = -\alpha\nabla \cdot \left(\frac{t\nabla u}{\sqrt{t^2|\nabla u|^2 + (1-t)^2}} \right) + K^*(Ku - z) = 0. \tag{1.7}$$

The equation is constructed based on gradually decreasing the smooth parameter $\beta(t) = (1-t)^2/t^2$, $t \in (0,1]$. Of course there exist many other choices of $H(u,t)$, e.g., the following is one possibility

$$H(u,t) = -\alpha\nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + (1-t)\beta_0 + t\beta}} \right) + K^*(Ku - z) = 0.$$

to advance t from 0 to the desirable 1. Here β_0 is a suitable large value of smooth parameter. The homotopy method avoids choosing β by some heuristics, and an efficient curve tracking idea is used.

Another recent method based on Bregman iterations [25] optimizes the regularization parameter α; if further combined with our homotopy algorithm, the overall method will have clear advantages but we have not explored this line of work.

The rest of this paper is organized as follows. In Section 2 we first introduce the standard finite difference scheme for our homotopy equation and then describe the details of how to track the solution curve. In Section 3 we present three new homotopy methods. Finally, we will give the numerical results of the implementation of the various algorithms on several tests in Section 4.

2. The Basic Homotopy Method

It is well-known the basic idea of homotopy algorithm is to construct a continuous map $H(u,t)$ which deforms a simple function $H(u,0)$ to the given function $H(u,1)$ as t varies from 0 to 1.

2.1. The discretization scheme

For the sake of simplicity, we assume that the image domain Ω is a square [14] such that the mesh size $\Delta x = \Delta y = 1$ when defining a regular $n \times n$ grid of pixels, indexed as (i,j), for $i = 1, \cdots, n, j = 1, \cdots, n$. Let $u_{i,j}$ represents the value of the function u at pixel (i,j). We define the discrete homotopy equation (1.7) by the standard finite difference as follows:

$$[H(u,t)]_{i,j} = -\alpha\nabla \cdot \left(\frac{t(\nabla u)_{i,j}}{\sqrt{t^2|(\nabla u)_{i,j}|^2 + (1-t)^2}} \right) + (K^*(Ku - z))_{i,j}, \quad 1 \le i,j \le n, \tag{2.1}$$

where the discrete gradient operator at each pixel (i,j) is defined by

$$(\nabla u)_{i,j} = \Big((u_x)_{i,j}, (u_y)_{i,j} \Big)$$

with

$$(u_x)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j}, & \text{if } i < n, \\ 0, & \text{if } i = n, \end{cases} \qquad (u_y)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j}, & \text{if } j < n, \\ 0, & \text{if } j = n, \end{cases} \tag{2.2}$$

and the discrete divergence operator is

$$(\nabla \cdot p)_{i,j} = \begin{cases} p^1_{i,j} - p^1_{i-1,j}, & \text{if } 1 < i < n, \\ p^1_{i,j}, & \text{if } i = 1, \\ -p^1_{i-1,j}, & \text{if } i = n, \end{cases} + \begin{cases} p^2_{i,j} - p^2_{i,j-1}, & \text{if } 1 < j < n, \\ p^2_{i,j}, & \text{if } j = 1, \\ -p^2_{i,j-1}, & \text{if } j = n. \end{cases} \tag{2.3}$$

Once we stack the grid functions u along rows of Ω into vector

$$\mathbf{u} = (u_{1,1}, \cdots, u_{n,1}, u_{1,2}, \cdots, u_{n,2}, \cdots, u_{1,n}, \cdots, u_{n,n})^T,$$

as commonly done, then $\mathbf{u} \in \mathbb{R}^{\mathrm{N}}$, where $\mathrm{N} = n^2$. Similar to $\mathbf{u}$, $\mathbf{z}$ and $H(\mathbf{u}, t)$ are vectors for z and $H(u, t)$, respectively. The homotopy equation (1.7) can be written in the form $H(\mathbf{u}, t) = 0$.

2.2. The Basic Homotopy Theory

We shall first review a general homotopy theory and then show that the theory applies to the above proposed map $H(\mathbf{u}, t)$.

Proposition 2.1. *Suppose that*

$$H : \mathbb{R}^{\mathrm{N}} \times [0, 1) \to \mathbb{R}^{\mathrm{N}}$$

satisfies these conditions:
(a) $H(\mathbf{u}, t)$ *is a* $\mathbf{C}^2$ *map,*
(b) *the* $\mathrm{N} \times (\mathrm{N} + 1)$ *Jacobian matrix* DH *has rank* N *on the set*

$$H^{-1}(0) = \{(\mathbf{u}, t) |\ \mathbf{u} \in \mathbb{R}^{\mathrm{N}},\ 0 \le t < 1,\ H(\mathbf{u}, t) = 0\},$$

(c) $H(\mathbf{u}, 0) = g_0(\mathbf{u}) = 0$ *has a unique solution* $\mathbf{u}_0$,
(d) $H(\mathbf{u}, 1) = g(\mathbf{u})$,
(e) $H^{-1}(0)$ *is bounded.*
Then there is a zero curve Γ *of* $H(\mathbf{u}, t)$, *emanating from the easily obtained solution of* $(\mathbf{u}_0, 0)$ *and reaching a zero* $\bar{\mathbf{u}}$ *of* $g(\mathbf{u})$ *at* $t = 1$.

Lemma 2.1. *The above homotopy function* $H(\mathbf{u}, t)$ *from* (1.7) *satisfies the conditions of Proposition* 2.1, *so the same results of Proposition* 2.1 *hold for* $H = H(\mathbf{u}, t)$.

Proof. First note that $H(\mathbf{u}, t)$ is smooth, so (a) is satisfied.

Assume $t \in [0, 1)$ is any given constant. If we can prove that the Jacobian matrix of $H(\mathbf{u}, t)$ (i.e., the first N columns of the DH) is a symmetric positive definite matrix, then $H(\mathbf{u}, t)$ satisfies (b), (c) and (e).

Define

$$L(u)_{i,j} = -\alpha \nabla \cdot \left(\frac{t(\nabla u)_{i,j}}{\sqrt{t^2 |(\nabla u)_{i,j}|^2 + (1 - t)^2}} \right), \quad 1 \le i, j \le n.$$

We can show that the Jacobian matrix of $L(\mathbf{u})$ is symmetric positive semidefinite. Since K^*K is symmetric positive semidefinite, the Jacobian matrix of $L(\mathbf{u}) + K^*K$ can be shown to be symmetric positive definite; see [30]. Therefore, (b), (c) and (e) hold. Then there exists a zero curve Γ of $H(\mathbf{u}, 0)$, starting from the solution of $H(\mathbf{u}, 0) = 0$ and reaching the solution of $H(\mathbf{u}, 1) = 0$. □

2.3. Curve tracking

The essence of a homotopy algorithm is to track the zero curve Γ until the solution of $H(\mathbf{u}, 1)$ is obtained, therefore, curve tracking becomes the main focus of the problem-solving effort. A typical curve tracking method consists of a succession of prediction phases and correction phases.

2.3.1. Prediction phase

Getting the predictor point is the main work for prediction phase. The approximation step is along the curve in the general direction of the tangent of the curve. The technology of numerical solution of initial value problem provides motivation for generating predictor steps. It consists of the following two steps:

1) find a predictor direction;

2) estimate the predictor steplength.

As the initial guess of the correction, the predictor point will directly affect the correction. How to choose the predictor point is one of the key factors for the speed of homotopy method.

Here we consider two strategies to get the predictor direction. Assume h is an estimate of the optimal step to take along Γ.

Strategy I is the naive choice of

$$(\mathbf{u}(t), t) = (\mathbf{u}(t_1), t_1 + h), \tag{2.4}$$

where $(\mathbf{u}(t_1), t_1)$ is the previous corrector point.

Strategy II is by the Lagrange interpolation formula.
Take the Lagrange quadratic interpolation as an example. A predictor point is obtained by three known points on the solution curve Γ. Assume the three points are $P^{(1)} = (\mathbf{u}(t_1), t_1)$, $P^{(2)} = (\mathbf{u}(t_2), t_2)$, and $P^{(3)} = (\mathbf{u}(t_3), t_3)$. The predictor point is

$$(\mathbf{u}(t), t) = p(t_3 + h),$$

where $p(t)$ is the Lagrange quadratic interpolating $(\mathbf{u}(t), t)$ at t_1, t_2, and t_3. Precisely,

$$p(t_1) = (\mathbf{u}(t_1), t_1), \quad p(t_2) = (\mathbf{u}(t_2), t_2), \quad p(t_3) = (\mathbf{u}(t_3), t_3),$$

and each component of $p(t)$ is a polynomial in t as follows

$$p(t) = \left(\frac{(t-t_2)(t-t_3)}{(t_1-t_2)(t_1-t_3)}\mathbf{u}(t_1) + \frac{(t-t_1)(t-t_3)}{(t_2-t_1)(t_2-t_3)}\mathbf{u}(t_2) + \frac{(t-t_1)(t-t_2)}{(t_3-t_1)(t_3-t_2)}\mathbf{u}(t_3),\ t \right). \tag{2.5}$$

By a similar procedure we can obtain the Lagrange linear interpolation which uses two points on the curve and the Lagrange cubic interpolation which use four points on the curve.

For the predictor steplength, we first estimate h by $h = \theta(1-t)$ ($\theta \in (0,1)$), and we further adjust h from tuning θ according to the performance of the corrector procedure as done below in Algorithms 2–3: When a corrector step terminates within prescribed steps of $it1$, θ is considered too small for the next predictor and is increased, when the iterations terminate over some $it2 > it1$ and converge, θ is considered too large and will be decreased, while if the iterations diverge, the predictor-corrector step is abandoned and then is restarted starting with a smaller θ.

2.3.2. Correction phase

As we know, the prediction phases and the correction phases mutually affect each other. After a predictor point $(\mathbf{u}^0, t_0)$ has been determined, a correction phase is performed to bring the predicted point back to the curve by applying one or more iterative steps of an iterative procedure (typically of Newton or gradient type) for solving $H(\mathbf{u}, t) = 0$ at $t = t_0$.

Specifically, the Euler-Lagrange equation $H(\mathbf{u}, t_0)$ is solved by

$$\mathbf{u}^{k+1} = \mathbf{u}^k - A(\mathbf{u}^k)^{-1} H(\mathbf{u}^k, t_0), \quad k = 0, 1, \cdots, \tag{2.6}$$

where $A(\mathbf{u})$ is a linear operator for $H(\mathbf{u}, t_0)$. As discussed in Section 1, apart from the Newton method, there are four kinds of methods for solving the TV model $H(\mathbf{u}, t) = 0$.

If the predictor point lies in its quadratic convergent domain, the Newton method has a local quadratic convergence, implying that the correction will be finished usually within 3 steps, so indeed we set $it1 = 3$ in the homotopy methods below. However, such a domain of convergence is small, in order to ensure that $u(t)$ lies in the quadratic convergent domain of $H(\mathbf{u}, t_0) = 0$, the predictor point should approximate to the curve as accurately as it possibly can.

Comparing with the Newton method, the fixed-point method and the primal-dual method have a larger domain of convergence, so demanding less in approximation accuracy of the predictor point. Below we shall also use such methods as correctors in a comparison.

3. Homotopy Methods

In this section we first introduce the Chan, Zhou and Chan's continuation method [15], and then describe the homotopy method of Melara et al. [22] before we present three new homotopy methods. These homotopy methods differ from each other only in the choice of algorithms for the prediction phase and the corrector phase. Our aim to construct the homotopy equation is mainly to get a fast and highly accurate solution of the Euler-Lagrange equation (1.5) with the desirable (small) smooth parameter $(1-t)^2/t^2$ by tracking the zero curve Γ efficiently. During the curve tracking process, a reliable predictor point $(\mathbf{u}(t), t)$ for the correction is obtained from previous corrections.

Based on the curve tracking framework, different choice of prediction schemes and correction methods, we get several variants of a homotopy method. We now introduce these methods in details, defining $H(\mathbf{u}, t_0)$ in each case and the numerical schemes.

3.1. CZC method

For ease of comparisons, we shall denote by CZC the Chan, Zhou and Chan's continuation method (see [15]) for solving equation (1.5). The Newton method for (1.5) is the following

$$\left[-\alpha\nabla \cdot \left(\frac{1}{\sqrt{|\nabla u^k|^2 + \beta}} \Big(I_2 - \frac{\nabla u^k (\nabla u^k)^T}{|\nabla u^k|^2 + \beta} \Big) \nabla \right) + 1 \right] \delta u^k = -g(u^k), \tag{3.1}$$

where u^k is the kth iterate and $\delta u^k = u^{k+1} - u^k$. The algorithm proceeds as follows:

Algorithm 3.1 (CZC method – Chan, Zhou and Chan [15])

Step 1. *Set $k = 1$ and choose a suitable large α_1 and β_1. Use the initial guess $u^{1,1} = z$.*

Step 2. *While $\alpha_k \leq \alpha$, do*

(a) *Use the Newton method, with $u^{k,1}$ as the initial guess (i.e., solve (3.1) with β_1). Denote the solution as $u^{k+1,1}$.*

(b) *Choose $\alpha_{k+1} > \alpha_k$ by $\alpha_{k+1} = \gamma\alpha_k$.*

(c) *Set $k = k + 1$.*

Step 3. *Let the final solution of Step 2 be denoted by $u^{*,1}$. Set $l = 1$.*

Step 4. *While $\beta_l \geq \beta$, do*

(a) *Use the Newton method, with $u^{*,l}$ as the initial guess (i.e., solve (3.1) with β_l). Denote the solution as $u^{*,l+1}$.*

(b) *Choose $\beta_{l+1} < \beta_l$ by $\beta_{l+1} = \tau\beta_l$.*

(c) *Set $l = l + 1$.*

Step 5. *The final solution of Step 4, denoted by $u^{*,*}$ will be the solution to the Euler-Lagrange equation (1.5) with the prescribed parameters α and β.*

As no curve tracking is used in [15], their Newton corrector steps for parameters α and β using the simplistic $\alpha_k = \gamma\alpha_{k-1}$ and $\beta_k = \tau\beta_{k-1}$ till α_k and β_k approximate to the prescribed parameters at which they want to solve (1.5), where the initial α_1 and β_1 are chosen first, γ is an incremental ratio – we have tested several choices of $\gamma = 1.2, 2, 4, 5, 10$ and found that $\gamma = 2$ is usually a good choice, and τ is a decremental ratio whose choice influenced by the noisy level and the size of β, in [15] the authors choose $\tau = 0.5$ as the first step and then $\tau = 0.2$ for their tests when $\beta = 0.1$, but this is not always a good choice for other cases. Theoretically their method only works if the domain of convergence depends linearly on the parameters (which is hardly the case for a nonlinear problem).

3.2. Melara's method

Melara *et al.* [22] proposed a similar method to CZC which just reduced β by a fixed ratio. Since they used a Newton method combined with a line search step in an augmented Lagrangian method, our implementation below with a linear search step for the standard TV model is only approximate to their work.

Recall from (1.4) the TV energy functional is $f(u) = \alpha \int_\Omega \sqrt{|\nabla u|^2 + \beta} dxdy + \frac{1}{2}\|Ku - z\|^2$, and each Newton step is as given by (3.1).

To ensure that $u^{k+1} = u^k + \delta u^k$ is meaningful, the TV energy functional is used. The initial linesearch parameter $s \in \mathbb{R}$ is set to $s := 1$. The value of s is determined based on a sufficient decrease criterion

$$f(u^k + \delta u^k) < f(u^k) + 10^{-4} s g(u^k) \cdot \delta u^k. \tag{3.2}$$

If the steplength s does not meet (3.2), then it is reduced by 1/2, i.e., $s := s/2$, until (3.2) is satisfied. Then, reset $s := 1$ to compute the $(k + 1)$th iterate by (3.1). Thus the procedure for Melara's method is similar to the continuation of β for Algorithm 3.1, namely Step 3 to Step 5, and (1.5) is solved by the Newton method combined with a line search.

3.3. Method 1

Although the CZC method is quadratically convergent if converging, it has not addressed the problem of how to find a robust and fully satisfactory selection procedure; Melara's method is a variant of the CZC. Methods 1–3 can be considered as new CZC methods for the continuation of smooth parameter only, with curve tracking to select the smooth parameter adaptively.

As a local quadratic convergence method, the Newton method converges to the solution with few steps when the initial guess approximates to the solution. Method 1 makes full use of this property by adjusting the steplength not only according to whether the Newton method's success or failure, but also in accordance with the steps of correction. Here the Newton correction step for (1.7) is

$$\left[K^*K - \alpha\nabla\cdot\left(\frac{t}{\sqrt{t^2|\nabla u^k|^2+(1-t)^2}}\Big(I_2 - \frac{t^2\nabla u^k(\nabla u^k)^T}{t^2|\nabla u^k|^2+(1-t)^2}\Big)\nabla\right)\right]\delta u^k = -H(u^k,t), \quad (3.3)$$

where $I_2 \in \mathbb{R}^{2\times 2}$ is an identity operator. We discretize the linear operator

$$K^*K - \alpha\nabla\cdot\left(\frac{t}{\sqrt{t^2|\nabla u^k|^2+(1-t)^2}}\Big(I_2 - \frac{t^2\nabla u^k(\nabla u^k)^T}{t^2|\nabla u^k|^2+(1-t)^2}\Big)\nabla\right)$$

as in §2.1 to get $A(\mathbf{u}^k)$ in (2.6) by standard finite differences.

The algorithm with Strategy I for the predictor phase is shown below. In the numerical implementation we terminate the curve tracking when the homotopy parameter t reaches $1/(1+\sqrt{\beta})$ for a desired β.

Algorithm 3.2 (Method 1) *Assume the required β is prescribed.*

Step 1. *Set $\theta \in (0,1)$, $h := \theta(1-t)$, $t_0 := 0$, $\mathbf{u}(t_0) := z$, arcar := absolute residual tolerance for tracking Γ, and ansar := absolute residual tolerance for the answer.*

Step 2. *Set $t := t_0 + \theta(1-t_0)$,*
$\mathbf{u}(t) := \mathbf{u}(t_0)$.
If $t \geq 1/(1+\sqrt{\beta})$, then
$t = 1/(1+\sqrt{\beta})$, *and solve* (2.6) *to obtain* $\mathbf{u}^*$.
If $\|g(\mathbf{u}^*)\| \leq ansar$,
then return (solution has been found).
Else go to Step 3.
Else solve (2.6) *to obtain* $\mathbf{u}^*(t)$ *until either*
$\|H(\mathbf{u}^*(t),t)\| \leq arcar$, *and record the iteration count,*
or the maxit iterations have been performed.

Step 3. *If the iteration* (2.6) *did not converge yet when maxit steps are reached, then reduce θ by $\theta := \theta/2$.*
If θ is unreasonably small, then
return with an error flag.
Else return to Step 2.

Else set $(\mathbf{u}(t_0), t_0) := (\mathbf{u}^*(t), t)$.
If the iteration count is less than $it1$*, then*
increase θ *by* $\theta := \min(1.2\theta, 0.9)$.
If the iteration count is more than $it2$*, then*
reduce θ *slightly by* $\theta := \theta/1.2$.
Return to Step 2.

In the above algorithm *maxit* denotes the number of maximum correction iterations prescribed to guarantee the predictor point is a reasonable approximation, and we estimate the steplength h by $h = \theta(1-t)$ with $\theta \in (0,1)$. It is well known that the Newton method can reduce a nonlinear residual by the factor of 10^{-6} within 3 steps when the initial guess lies in its quadratic convergence domain, so we can set $it1 = 3$ and $it2 = 5$ in this algorithm. With these choices, we hope to accelerate the curve tracking using the least number of correction iterations.

3.4. Method 2

Method 2 below is our new and recommended restoration method which combines the local quadratic convergence of the Newton method with the more efficient predictor method of Strategy II, i.e., the Lagrange interpolation. The following table summarizes three Lagrange interpolation methods that we shall consider in our homotopy method:

Method	Interpolation type	Correction by
Method 2a	Lagrange linear interpolation	(3.3)
Method 2b	Lagrange quadratic interpolation	(3.3)
Method 2c	Lagrange cubic interpolation	(3.3)

As we know, the Newton method can arrive at quadratical convergence when the initial guess is close to the solution. The predictor step and corrector steps mutually influence each other during the curve tracking. Generally speaking, the correction methods converge quickly when the predictor point is close to the corrector point. In order for the correction to succeed in few steps, we should ensure the predictor point approximates well to a corrector point. Our previous use of one correction point to get the next predictor point in Algorithm 2 does not make full use of the geometric property of the curve, which may not be good for prediction. While a Lagrange interpolation using more than one point on the curve Γ, we hope to follow the geometric property of Γ more accurately.

In summary, taking the Lagrange quadratic interpolation as an example, the new algorithm is:

Algorithm 3.3 (Method 2) *Assume the required* β *is prescribed.*

Step 1. *Set* $\theta \in (0,1)$, $h := \theta(1-t)$, $t_0 := 0$, $\mathbf{u}(t_0) := \mathbf{z}$, $t_1 := \theta$, *and* $t_2 := 2\theta - \theta^2$. $arcar :=$ *absolute residual tolerance for tracking* Γ, $ansar :=$ *absolute residual tolerance for the answer. Compute* $\mathbf{u}(t_1)$, *using* (2.6) *with* $\mathbf{u}(t_0)$, *and then compute* $\mathbf{u}(t_2)$ *taking* $\mathbf{u}(t_1)$ *as the initial guess.*

Step 2. Set $t := t_2 + \theta(1 - t_2)$,
Step 2 is similar to Algorithm 3.2, but the predictor point is obtained by Lagrange interpolation.

Step 3. *This step is the same as Step* 3 *of* Algorithm 3.2*, and* $(\mathbf{u}(t_0), t_0) := (\mathbf{u}(t_1), t_1)$, $(\mathbf{u}(t_1), t_1) := (\mathbf{u}(t_2), t_2)$, $(\mathbf{u}(t_2), t_2) := (\mathbf{u}^*(t), t)$.

3.5. Method 3

Method 3 below, similar to Method 2, uses the same prediction scheme, but they differ in the correction steps, i.e., they can share the same algorithm with different choices of $A(\mathbf{u})$ in (2.6). Method 3 uses a fixed-point method as its correction, and the corrector equation is thus

$$\left[K^*K - \alpha\nabla \cdot \left(\frac{t_0}{\sqrt{t_0^2|\nabla u^k|^2 + (1-t_0)^2}}\nabla\right)\right]\delta u^k = -H(u^k, t_0). \tag{3.4}$$

This application, though as a minor departure from our main theme of improving the Newton methods, aims to illustrate the usefulness of a homotopy method. As usual [28], we use the standard finite difference discrete the linear operator

$$K^*K - \alpha\nabla \cdot \left(\frac{t_0}{\sqrt{t_0^2|\nabla u^k|^2 + (1-t_0)^2}}\nabla\right) \tag{3.5}$$

to get $A(\mathbf{u}^k)$ in (2.6). It is well known that the method is robust and have a large domain of convergence, but only linearly convergent. Moreover, the iterative methods slow down for small β so applying a homotopy idea is also a natural consideration. As our aim is not to trace the solution curve accurately but to improve the initial guess for the final equation, we can set $it1 = 8$, $it2 = 10$ and $ansar = 10^{-4}$ when we use the fixed-point method as the correction method.

4. Numerical Experiments and Discussions

In this section we test our restoration algorithms on several images of resolutions 128×128, and 256×256 pixels. The quality of the restored images will be measured by signal to noise ratio (SNR) and peak signal to noise ratio (PSNR) which are defined as follows:

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}u_{i,j}^2}{\sum_{i=1}^{n}\sum_{j=1}^{n}(u_{i,j}-\tilde{u}_{i,j})^2}, \quad \mathrm{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}(u_{i,j}-\tilde{u}_{i,j})^2},$$

where u, and $\tilde{u}$ are the original image, and the restored image. In all tests we use the ratio notation

$$\mathrm{AS} = \text{total Newton steps / total continuation steps}$$

to denote the accumulated number of Newton steps (indicating the computational complexity) over the total number of continuation steps (indicating the level of nonlinearity); if the denominator is 1, it means there is no continuation used.

Comparisons of Methods 1–2 with Newton, CZC and Melara's method. In our numerical experiment we first compare the convergence of the Newton method, the previous

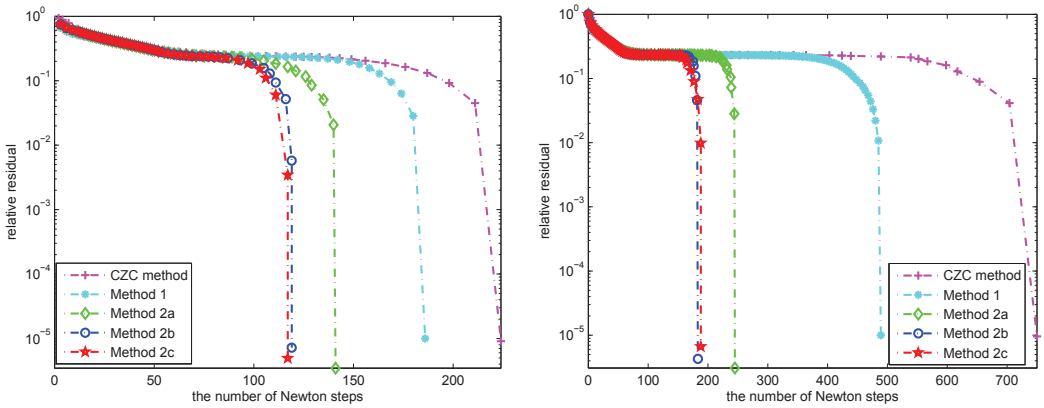

Fig. 4.1. A relative residual history by CZC method, Method 1, and Methods 2a-2c with $\beta = 10^{-10}$ (left) and $\beta = 10^{-16}$ (right).

CZC method (Algorithm 3.1) and the Melara's method with the above described new homotopy Method 1 and Method 2. The test image is a 128×128 "triangle" image contaminated with random noise, as shown in Fig. 4.3. We take the noisy image as the initial guess and $\alpha = 20$. The stopping criterion is to make the reduction of relative residual by a factor of 10^{-5}. Here we take $\theta = 0.01$ for Method 1 and Methods 2a-2c, and $\gamma = 2$, $\tau = 0.5$ for the CZC method. The summary of computational results for these methods is listed in Table 4.1. Clearly all homotopy type methods converge while the Newton method does not converge at all as expected. In particular, our new Method 2 based algorithm is about 3–4 times more efficient than the CZC method and for this test, our Method 1 is still competitive (to CZC and Melara) but less efficient than our Method 2.

For a graphical illustration, in Fig. 4.1, we plot the relative residuals' history of this experiment, and we present the changes of homotopy parameter t in Fig. 4.2 for these methods. Comparing with the CZC method and Method 1, Methods 2a–2c take less cpu time and are in general robust; among them, Method 2b will be use and recommended in later tests.

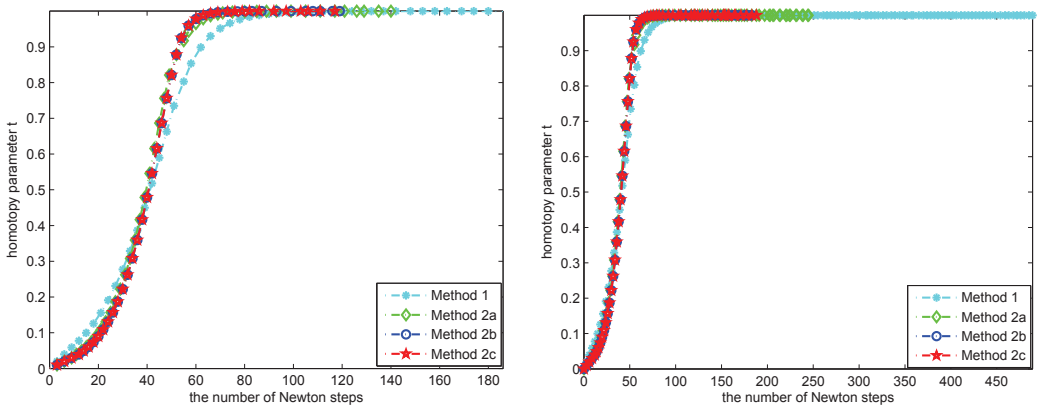

Fig. 4.2. A description for the change of homotopy parameter t by Method 1, and Methods 2a-2c with $\beta = 10^{-10}$ (left) and $\beta = 10^{-16}$ (right).

Table 4.1: Comparison of the Newton method, CZC method, Melara's method, Method 1, and Methods 2a-2c with different β.

method	$\beta = 10^{-6}$			$\beta = 10^{-8}$			$\beta = 10^{-10}$		
	AS	PCG	time	AS	PCG	time	AS	PCG	time
Newton	*	*	*	*	*	*	*	*	*
CZC	119/33	19005	57	153/40	36000	86	224/46	71500	203
Melara's	216/8	48466	120	284/10	82466	201	563/12	222466	567
Method 1	115/33	17784	54	142/39	31254	81	186/47	54257	137
Method 2a	83/34	11344	35	101/38	20286	59	141/46	41289	113
Method 2b	77/33	9644	29	90/36	16144	51	119/43	31144	82
Method 2c	76/33	9128	27	89/36	15628	50	117/42	30628	80
method	$\beta = 10^{-12}$			$\beta = 10^{-14}$			$\beta = 10^{-16}$		
	AS	PCG	time	AS	PCG	time	AS	PCG	time
Newton	*	*	*	*	*	*	*	*	*
CZC	336/53	127500	361	557/60	238486	676	750/66	336000	952
Melara's	375966	869/14	958	1469/16	677466	1767	2069/18	978966	2597
Method 1	268/66	95757	246	372/91	148257	381	489/118	207257	509
Method 2a	176/57	58789	154	220/69	82289	206	245/77	95789	248
Method 2b	144/49	43644	132	169/56	57144	153	183/61	64644	167
Method 2c	136/48	40128	127	158/55	52128	147	188/61	69128	184

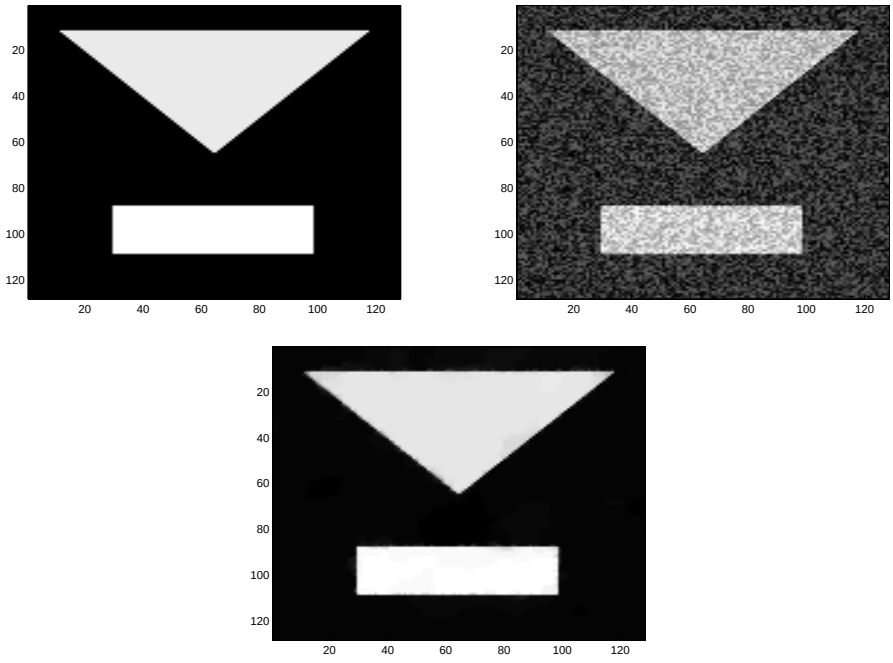

Fig. 4.3. Original "triangle" image (above left), noisy image (above right) and image recovered by Method 2b with $\beta = 10^{-10}$ (bottom).

Table 4.2: Comparison between the primal-dual method (PD) and Method 2b with different β.

method	$\beta = 10^{-8}$			$\beta = 10^{-9}$			$\beta = 10^{-10}$		
	AS	PCG	time	AS	PCG	time	AS	PCG	time
PD	28/1	13266	102	56/1	27266	205	148/1	73266	558
Method 2b	90/36	16144	117	106/39	24644	171	119/43	31144	224

Table 4.3: Comparison between the fixed-point method (FP) and method 3 with different β.

method	$\beta = 10^{-8}$			$\beta = 10^{-9}$			$\beta = 10^{-10}$		
	AS	PCG	time	AS	PCG	time	AS	PCG	time
FP	1324/1	603438	3661	1578/1	788088	4677	1937/1	967608	5852
Method 3	812/86	165335	1042	895/95	197335	1256	1042/104	265066	1721

Table 4.4: The restoration results of FP, PD, and Method 2b with different α.

method	$\alpha = 1$			$\alpha = 0.1$			$\alpha = 0.01$		
	AS	PCG	PSNR	AS	PCG	PSNR	AS	PCG	PSNR
FP	500/1	49500	23.80	500/1	49500	33.70	247/1	24453	44.02
PD	500/1	49500	23.79	500/1	49500	33.71	271/1	26829	44.02
Method 2b	212/53	20774	24.16	138/37	14454	33.75	85/35	8811	44.02

Comparisons of Method 2b and Method 3 with old non-homotopy methods. Next we compare the primal-dual method (PD) with Method 2b, and the fixed-point method (FP) with Method 3 with $\beta = 10^{-8},\ 10^{-9},\ 10^{-10}$, respectively. We use the same test image, the same α and the same stopping criterion as the above experiment. Table 4.2 presents the comparative results between the primal-dual method (PD) and Method 2b. Although the primal-dual method is slightly faster than Method 2b when $\beta = 10^{-8}$, the results are the other way round when β becomes even smaller.

Table 4.3 shows the comparison between the fixed-point iteration (FP) and Method 3. It is easy to draw the conclusion that Method 3 accelerates the convergence of the fixed-point method as much as three times, although the latter is known as a converging method.

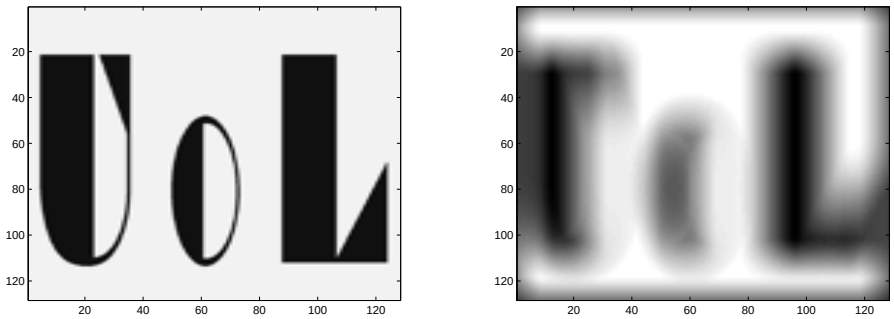

Fig. 4.4. The original "UoL" image (left) and image degraded by Gaussian blur with PSNR≈13.70 (right).

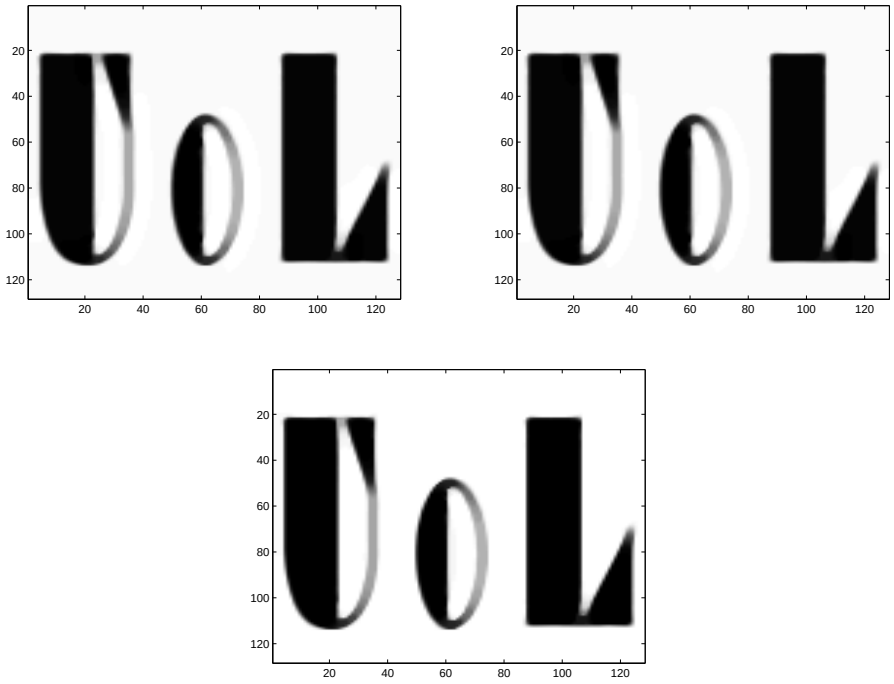

Fig. 4.5. Images restored by the fixed-point method (above left), primal-dual method (above right), and Method 2b (bottom) with $\alpha = 1,\ \beta = 10^{-8}$.

Table 4.5: The restoration results of FP, PD, and Method 2b on 128×128 "UoL" image contaminated with noise and blur.

method	AS	PCG	relative residual	time	PSNR
FP	100/1	9900	2.64e-2	478	16.1634
PD	100/1	9900	3.6e-2	495	16.1738
FP	300/1	29700	2.07e-2	1426	16.3961
PD	300/1	29700	2.8e-2	1484	16.3945
FP	500/1	49500	2.26e-3	2363	16.5360
PD	500/1	49500	3.1e-3	2461	16.5307
Method 2b	207/57	21087	4.8e-6	996	16.7007

Table 4.6: The comparison between Fast-TV method and Method 2b. The maximum number of iterations is set to be 200 for the Fast-TV method.

method	"Uol" (128)		"cameraman" (256)	
	PSNR	time	PSNR	time
Fast-TV	16.4332	290	27.1082	1868
Method 2b ($\beta = 10^{-1}$)	16.686	230	27.6596	956
Method 2b ($\beta = 10^{-6}$)	16.7007	795	27.6507	1987

Comparisons of Method 2b with FP, PD and the splitting method for image deblurring. The test image is a 128×128 "UoL" image which is applied with Gaussian blur with PSNR$\approx$13.70 as shown in Fig. 4.4. We consider three choices of $\alpha = 1, 0.1, 0.01$ as the

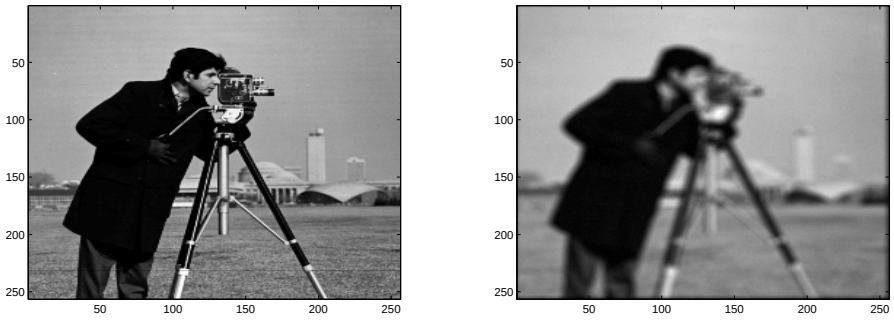

Fig. 4.6. Original "cameraman" image (left) and image degraded by Gaussian blur and Gaussian noise with PSNR=20.54 (right).

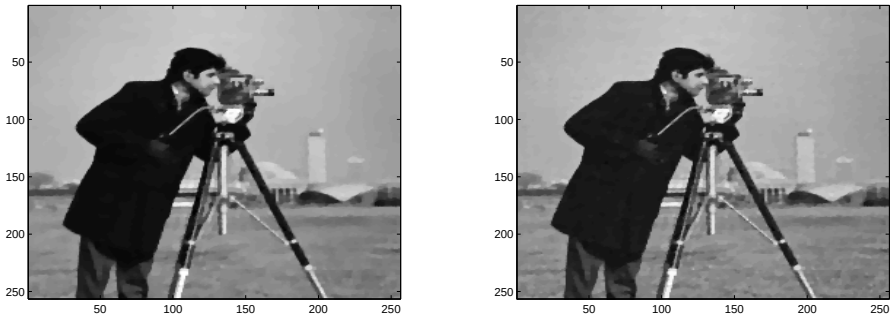

Fig. 4.7. Images restored by the Fast-TV method (left) and Method 2b with $\beta = 10^{-6}$(right).

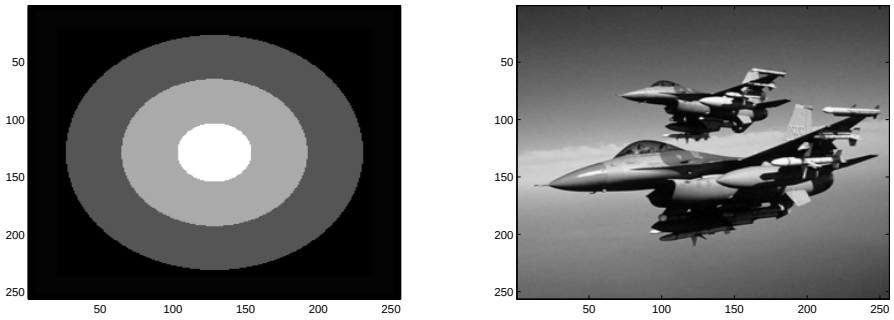

Fig. 4.8. Original images (left: "Circbox" and right: "Aircraft")

regularization parameter and $\beta = 10^{-8}$; Table 4.4 shows the computational results of these four methods. The conclusion one can draw from this table is that Method 2b not only gets a better restored image, but also requires less outer iterations and inner iterations. Fig. 4.5 displays images restored using these three methods with $\alpha = 1$.

Next we test an image which is contaminated with both Gaussian noise and Gaussian blur. First, we compare the fixed-point method and the primal-dual method (PD) with Method 2b on the above "UoL" image (with the initial PSNR = 10.26 due to noise added to the right plot of Fig. 4.4). We take $\alpha = 1$ and $\beta = 10^{-8}$. Table 4.5 reports the numerical results of these three algorithms. It is not difficult to see Method 2b is the most robust and the fastest algorithm of three for this denoising and deblurring problem. Second, we compare Method 2b with the

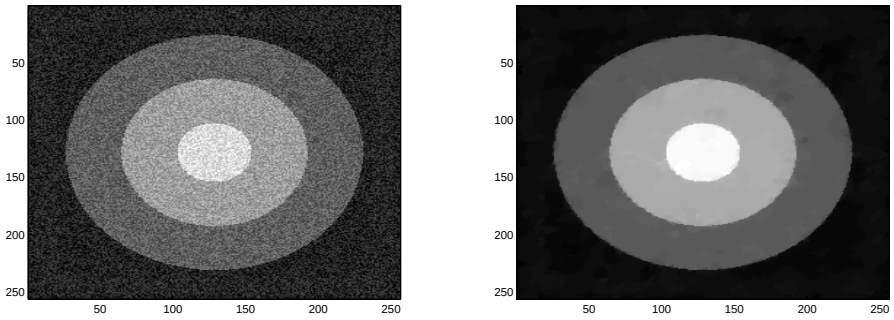

Fig. 4.9. Noisy "Circbox" image with SNR≈5 (left) and image recovered by Method 2b with AS = 92/36 (right).

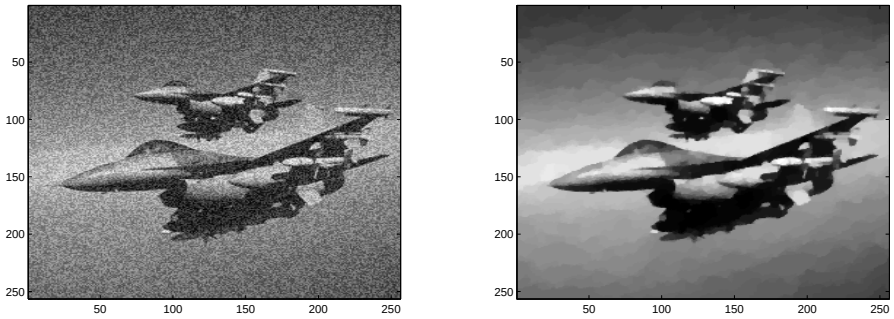

Fig. 4.10. Noisy "Aircraft" image with SNR≈5 (left) and image recovered by Method 2b with AS = 105/40 (right).

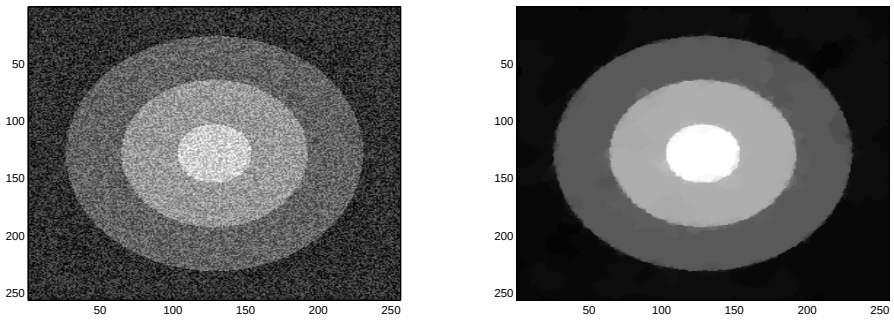

Fig. 4.11. Noisy "Circbox" image with SNR≈1 (left) and image recovered by Method 2b with AS = 121/44 (right).

fast total variation minimization (Fast-TV) method from [19]; we take the parameter $\alpha_1 = 0.05$ for "UoL" and $\alpha_1 = 0.02$ for "cameraman" as the suitable choices, and take the parameter $\alpha_2 = \alpha = 1$ for "UoL" and $\alpha_2 = 0.1$ for "cameraman". The computational results are shown in Table 4.6. According to Table 4.6 and Fig. 4.7, we find that Method 2b attains slightly better quality of image restorations than the Fast-TV method in comparable complexity. Of course, the Newton type methods are known to have no convergence for this type of problems so the fact that our Method 2b based on homotopy and the Newton method can reach comparable efficiency is already remarkable. It remains to explore the option of accelerating the Fast-TV method by a homotopy method.

Further tests of Method 2b for images with heavy noise. Finally we test Method 2b on two images as shown in Fig. 4.8. We add heavy noise to these images, as shown in Figs. 4.9–4.12. In our tests we take the noisy image as the initial guess as before, $\beta = 10^{-8}$ in these cases, $\alpha = 20$ for "Circbox" image and 40 for "Aircraft" image for the noise level SNR ≈ 5, and 40 for "Circbox" image and 50 for "Aircraft" image for the level SNR ≈ 1. It is easy to see that Method 2b can remove the noise efficiently even for images with large noisy-to-signal ratio.

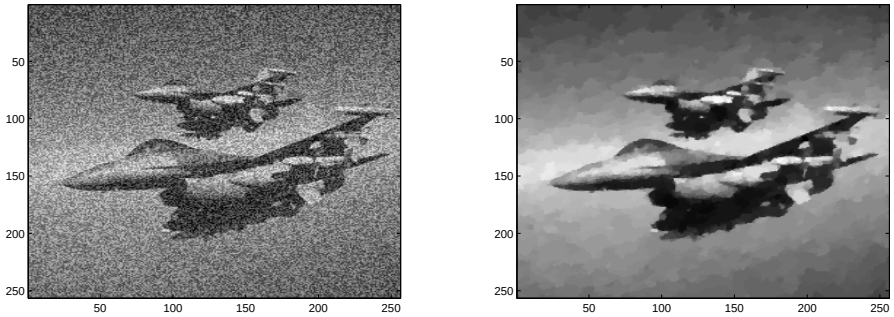

Fig. 4.12. Noisy "Aircraft" image with SNR≈1 (left) and image recovered by Method 2b with AS = 110/40 (right).

5. Conclusions

The primal TV image restoration model can be solved by many effective methods. But none of them is of a Newton type, if not using a dual formulation, because of no convergence. This paper proposed to use a homotopy method with curve tracking to choose the regularizing parameter β adaptively. The resulting method turns out to be quite robust to drive the Newton method (as a corrector) to convergence for a range of test images. When used for the converging fixed-point method and the primal-dual method, the proposed homotopy algorithm also accelerates on these methods. Numerical tests for deblurring problems and comparisons to the splitting method are also presented. For high order (non-TV) image restoration models [5, 6, 13, 14, 34] it will be of interest to see how to generalize our homotopy method which will be considered in our future work.

References

[1] R. Acar, C.R. Vogel and C. Fatemi, Analysis of bounded variation penalty methods for ill-posed problems, *Inverse Probl.,* **10** (1994), 1217-1229.

[2] E.L. Allgower and K. Georg, Numerical Path Following, in Handbook of Numerical Analysis, North-Holland, Amsterdam, 1997.

[3] J.L. Carter, *Dual methods for total variationbased image restoration,* Ph.D. thesis, UCLA, 2001.

[4] J. F. Cai, S. Osher and Z. Shen, Split Bregman methods and frame based image restoration, *Multiscale Model. Simul.,* **8** (2009), 337-369.

[5] Carlos Brito-Loeza and Ke Chen, Multigrid algorithm for high order denoising, *SIAM J. Image Sci.,* **3**:3 (2010), 363-389.

[6] Carlos Brito-Loeza and Ke Chen, On high-order denoising models and fast algorithms for vector-valued images, *IEEE T. Image Process.,* **19**:6 (2010), 1518-1527.

[7] A. Chambolle, An algorithm for total variation minimization and applications, *Journal Math. Imaging Vis.*, **20** (2004), 89-97.
[8] A. Chambolle and P.-L. Lions, Image recovery via total variation minimization and related problems, *Numer. Math.,* **76**:2 (1997), 167-188.
[9] T.F. Chan and Ke Chen, An optimisation-based multilevel algorithm for total variation image denoising, *SIAM Journal on Multiscale Modeling and Simulation,* **5**:2 (2006), 615-645.
[10] T.F. Chan, Ke Chen and J.L. Carter, Iterative methods for solving the dual formulation arising from image restoration, *Electron. T. Numer. Ana.,* **26** (2007), 299-311.
[11] T.F. Chan, G.H. Golub and P. Mulet, A nonlinear primal-dual method for total variation-based image restoration, *SIAM J. Sci. Comput.,* **20** (1999), 1964-1977.
[12] S.N. Chow, J. Mallet-Paret and J.A. Yorke, Finding zeros of maps: homotopy methods that are constructive with probability one, *Math. Comput.,* **32** (1978), 887-899.
[13] T.F. Chan, A. Marquina and P. Mulet, High-order total variation-based image restoration, *SIAM J. Sci. Comput.,* **22**:2 (2000), 503-516.
[14] T.F. Chan and J.H. Shen, Image Processing And Analysis-Variational, PDE, Wavelet, And Stochastic Methods, SIAM publications, Philadelphia, 2005.
[15] T.F. Chan, H. Zhou and R. Chan, Advanced signal processing algorithms, *in Proceedings of the International Society of Photo-Optical Instrumentation Engineers, F. T. Luk, ed., SPIE,* (1995) 314-325.
[16] D.C. Dobson and C.R. Vogel, Convergence of an iterative methods for total variation denoising, *SIAM J. Numer. Anal.,* **34** (1997), 1779-1791.
[17] Z. Lin, B. Yu and D. Zhu, A continuation method for solving fixed points of self-mappings in general nonconvex sets, *Nonlinear Analysis,* **52** (2003), 905-915.
[18] C.B. Garcia and W.I. Zangwill, Pathways to Solutions, Fixed Points and Equilibria, Prentice-Hall, Englewood Cliffs, N.J., 1981.
[19] Y. Huang, M.K. Ng and Y. Wen, A fast total variation minimization method for image restoration, *SIAM Journal on Multiscale Modeling and Simulation,* **7** (2008), 774-795.
[20] R.B. Kellogg, T.Y. Li and J.A. Yorke, A constructive proof of the Brouwer fixed-point theorem and computational results, *SIAM J. Numer. Anal.,* **18** (1976), 473-483.
[21] T. Lu, P. Neittaanmäki and X.-C. Tai, A parallel splitting up method and its application to Navier-Stokes equations, *Appll. Math. Lett.,* **4**:2 (1991), 25-29.
[22] L.A. Melara, A.J. Kearsley and R.A. Tapia, Augmented Lagrangian homotopy method for the regularization of total variation denoising problems, *J. Optimiz. Theory App.,* **134**:1 (2007), 15-25.
[23] M.K. Ng , L.Q. Qi , Y.F. Yang and Y.M. Huang, On semismooth Newton's methods for total variation minimization, *J. Math. Imaging Vis.,* **27**:3 (2007), 265-276.
[24] A. Marquina and S. Osher, Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal, *SIAM J. Sci. Comput.,* **22** (2000), 387-405.
[25] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, An iterative regularization method for total variation-based image restoration, *SIAM Journal on Multiscale Modeling and Simulation.,* **4** (2005), 460-489.
[26] L. Rudin, S. Osher and C. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D,* **60** (1992), 259-268.
[27] J. Savage and Ke Chen, An improved and accelerated non-linear multigrid method for total-variation denoising, *Int. J. Comput. Math.,* **82**:8 (2005), 1001-1015.
[28] C.R. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, 2002.
[29] C.R. Vogel and M.E. Oman, Iterative methods for total variation denoising, *SIAM J. Sci. Comput.,* **17** (1996), 227-238.
[30] C.R. Vogel and M.E. Oman, Fast, robust total variation-based reconstruction of noisy, blurred images, *IEEE T Image Process.*, **7**:6 (1998), 813-824.
[31] L.T. Watson, C. Billups and P. Morgan, Algorithm 652 Hompack: A suite of codes for globally

convergent homotopy algorithms, *ACM T Math. software,* **13**:3 (1987), 281-310.

[32] J.Weickert, B. M. ter Haar Romeny and M. A. Viergever, Efficient and reliable schemes for nonlinear diffusion filtering, *IEEE T. Image Process.,* **7** (1998), 398-410.

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

[34] Wei Zhu and T.F. Chan, Image denoising using mean curvature, *SIAM J. Image Sci.,* (2012) in press.