

EFFICIENT HOMOTOPY SOLUTION AND A CONVEX COMBINATION OF ROF AND LLT MODELS FOR IMAGE RESTORATION

FENLIN YANG, KE CHEN, AND BO YU

Abstract. The Rudin, Osher, and Fatemi model [20] (ROF) for image restoration has been extensively studied due to its edge preserving capability, but for images without edges (jumps), the solution to this model has the undesirable staircasing effect. To improve the model, Lysaker, Lundervold and Tai [14] (LLT) proposed a better second-order functional suitable for restoring smooth images but it is difficult to preserve discontinuities for non-smooth images. It turns out that results from convex combinations of ROF model and LLT model can preserve the main advantages of both models (see [16, 9]). In this paper, we first propose an applicable homotopy algorithm based fixed point method for the LLT model. We then propose two new variants of convex combination models. Numerical experiments are shown to demonstrate the advantages of these combination models and the robustness of our homotopy algorithm.

Key words. Image restoration, total variation, fourth-order PDE, fixed point method, homotopy method, convex combination.

1. Introduction

An observed image f can often become blurry and noisy during the formation, transmission or recording process for the original image u . The common additive degradation model is

$$(1) \quad f = Ku + \eta,$$

where η is an additive noise term 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. The recovery of the original image from the observed image is an essential pre-processing phase for further image processing tasks such as edge detection, pattern recognition, and object tracking, etc.

The usual approach for image restoration solves the following constrained optimization problem:

$$(2) \quad \min_u R(u) \quad \text{subject to} \quad \|Ku - f\|^2 = \sigma^2.$$

This problem is naturally linked to the following unconstrained problem – the minimization of the total variation penalized least squares functional (see [20, 4, 24]):

$$(3) \quad \min_u \left\{ J(u) = \alpha R(u) + \frac{1}{2} \|Ku - f\|^2 \right\}.$$

Here $\|\cdot\|$ is the norm in $\mathbb{L}^2(\Omega)$ and α is a positive parameter controlling the trade-off between goodness of fit-to-the-data and variability in u . $R(u)$ is some functional which controls the regularity of u and ensures the solvability of the inverse problem

Received by the editors March 28, 2011 and, in revised form, July 13, 2011.

2000 *Mathematics Subject Classification.* 35R35, 49J40, 60G40.

The research was supported by the National Natural Science Foundation of China (10671029) and the Research Fund for the Doctoral Programme of Higher Education (20060141029).

(1). Examples of regularization functionals that can be found in the literature [24, 26, 7, 2] include $R(u) = \|u\|, \|\Delta u\|, \|\nabla u\|$.

The total variation semi-norm proposed by Rudin, Osher, and Fatemi [20] (ROF) is one of the most effective regularization functionals for $R(u)$ which does not penalize discontinuities in u , and thus allows us to recover the edges of the original image. Its formula is

$$R_1(u) = TV(u) = \int_{\Omega} |\nabla u| dx dy = \int_{\Omega} \sqrt{u_x^2 + u_y^2} dx dy.$$

The corresponding Euler-Lagrange equation for (3) is

$$(4) \quad g_1(u) = -\alpha \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + K^*(Ku - f) = 0,$$

with homogeneous Neumann boundary condition $\frac{\partial u}{\partial \vec{n}} = 0$, and \vec{n} is the normal vector. Here β is a small positive parameter to avoid the denominator equals to zero. There are many fast methods for (4) (see [20, 22, 6, 17, 5, 8, 18]) up to now.

Although the ROF model yields very satisfactory results for removing noise while preserving edges, it suffers from the undesirable staircase effect for problems without sharp edges, namely the transformation of smooth regions (ramps) into piecewise constant regions (stairs). Some effort has been made to remedy this unfavorable property [15, 17, 19, 2, 21, 7, 10].

In [14], Lysaker, Lundervold and Tai (LLT) proposed a second-order functional as the regularization functional

$$R_2(u) = \int_{\Omega} |D^2 u| dx dy = \int_{\Omega} \sqrt{u_{xx}^2 + u_{xy}^2 + u_{yx}^2 + u_{yy}^2} dx dy.$$

The corresponding Euler-Lagrange equation for (3) using this $R_2(u)$ is

$$(5) \quad g_2(u) = \alpha \left[\left(\frac{u_{xx}}{|D^2 u|_{\beta}} \right)_{xx} + \left(\frac{u_{xy}}{|D^2 u|_{\beta}} \right)_{yx} + \left(\frac{u_{yx}}{|D^2 u|_{\beta}} \right)_{xy} + \left(\frac{u_{yy}}{|D^2 u|_{\beta}} \right)_{yy} \right] + K^*(Ku - f) = 0,$$

where β is a small positive parameter and $|D^2 u|_{\beta} = \sqrt{u_{xx}^2 + u_{xy}^2 + u_{yx}^2 + u_{yy}^2 + \beta}$. It is known that the LLT model can recover smooth surfaces. However, there exist two major challenges in dealing with this model. One is to preserve jumps as done by the ROF model and the other is to get a more efficient solution method for (5) than the gradient descent.

To address the first challenge, one idea is to combine the models of ROF and LLT because we desire restoration properties of both models. Therefore, Lysaker and Tai [16] suggested a convex combination of the respective two solutions from (4) and (5). Specifically, with $w^0 = f$, a new iteration w^{k+1} is generated by the convex combination

$$(6) \quad w^{k+1} = \theta^k v^{k+1} + (1 - \theta^k) u^{k+1} \quad k = 0, 1, 2 \dots,$$

where v^{k+1} and u^{k+1} are respectively obtained by the k th time marching iteration of ROF model and LLT model with w^k as their old iteration. Here the parameter θ^k which is applied to control the combination depends on ∇w^k as follows:

$$(7) \quad \theta^k = \begin{cases} 1, & \text{if } |\nabla w^k| \geq c, \\ \frac{1}{2} \cos\left(\frac{2\pi|\nabla w^k|}{c}\right) + \frac{1}{2}, & \text{elsewhere,} \end{cases}$$

where c is some constant parameter in the interval $[0, 1]$.

The above convex combination solution (6) reduces to the ROF solution in regions where $|\nabla u|$ is large (near edges) or to the LLT solution where $|\nabla u|$ is 0 (flat regions). It would be better to use the ROF solution when $|\nabla u| \approx 0$ i.e. not exactly 0 and also one may wish to solve a single PDE (from a combined optimization) instead of solving two separate PDEs. This is the idea taken up in Chang, Tai and Xing [9] who proposed a new combination of the ROF model and the LLT model in the form

$$(8) \quad \min_u \left[\alpha \left(\int_{\Omega} \theta |\nabla u| dx dy + \int_{\Omega} (1 - \theta) |D^2 u| dx dy \right) + \frac{1}{2} \|Ku - f\|^2 \right],$$

and its Euler-Lagrange equation is

$$(9) \quad \alpha \left\{ -\theta \nabla \cdot \left(\frac{\nabla u}{\sqrt{|\nabla u|^2 + \beta}} \right) + (1 - \theta) \left[\left(\frac{u_{xx}}{|D^2 u|_{\beta}} \right)_{xx} + \left(\frac{u_{xy}}{|D^2 u|_{\beta}} \right)_{yx} + \left(\frac{u_{yx}}{|D^2 u|_{\beta}} \right)_{xy} + \left(\frac{u_{yy}}{|D^2 u|_{\beta}} \right)_{yy} \right] \right\} + K^*(Ku - f) = 0.$$

Here the variable parameter θ is chosen as:

$$(10) \quad \theta = \begin{cases} 1, & \text{if } |\nabla u| \leq C_0 \text{ and } |\nabla u| \geq C_1, \\ C_d, & \text{if } C_0 + 5 \leq |\nabla u| \leq C_1 - 5, \\ 1 - \frac{(|\nabla u| - C_0)(1 - C_d)}{5}, & \text{if } C_0 \leq |\nabla u| \leq C_0 + 5, \\ 1 + \frac{(|\nabla u| - C_1)(1 - C_d)}{5}, & \text{if } C_1 - 5 \leq |\nabla u| \leq C_1. \end{cases}$$

Note that this varying θ suggests a global iteration scheme which is exactly the implementation of [9]. In computation, the parameters $C_0 = 0$, and $C_d = 0.05$ are fixed. The parameter C_1 is taken as 50 for most images and is properly modified for some images.

In this paper, we mainly address the second challenge. We first develop a fast homotopy algorithm based fixed point method for solving (5). It is also applicable to related combination models. Here to achieve fast convergence, the homotopy algorithm is equipped with an adaptively varying regularization parameter β in a predictor-corrector framework. Based on a convergent algorithm, we then propose two new variants as alternative combinations of ROF and LLT.

The rest of this paper is organized as follows. In Section 2 we first review the discretization scheme and two fixed point algorithms for ROF model and LLT model respectively. We then propose a homotopy method for LLT model. In Section 3 we present two homotopy-based fixed point schemes using convex combinations of ROF and LLT model for the image restoration problem. Finally, we give the numerical results of the implementation of the proposed algorithms on several tests in Section 4.

2. Numerical algorithms for image restoration by ROF and LLT

Before describing the detail of numerical algorithms, we proceed to outline the discretization scheme we use and to fix our notation. For the sake of simplicity, we assume that the image domain Ω is a square [8] such that the mesh size is $\Delta x = \Delta y = 1$ when defining a regular $n \times n$ grid of pixels, indexed as (i, j) , for

$i = 1, 2, \dots, n$, $j = 1, 2, \dots, n$. Let $u_{i,j}$ represent the value of the function u at pixel (i, j) . The discrete gradient operator at pixel (i, j) is defined by

$$(\nabla u)_{i,j} = \left((\nabla_x u)_{i,j}, (\nabla_y u)_{i,j} \right)$$

with

$$(\nabla_x u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j} & \text{if } i < n, \\ 0 & \text{if } i = n, \end{cases} \quad (\nabla_y u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j} & \text{if } j < n, \\ 0 & \text{if } j = n \end{cases}$$

The discrete divergence operator is the negative adjoint of the gradient operator due to the analysis of the continuous setting, namely $\nabla \cdot = -\nabla^*$. Therefore, it can be defined as follows:

$$(\nabla \cdot w)_{i,j} = \begin{cases} (w_1)_{i,j} - (w_1)_{i-1,j} \\ (w_1)_{i,j} \\ -(w_1)_{i-1,j} \end{cases} + \begin{cases} (w_2)_{i,j} - (w_2)_{i,j-1} & \text{if } 1 < i, j < n \\ (w_2)_{i,j} & \text{if } i = j = 1 \\ -(w_2)_{i,j-1} & \text{if } i = j = n. \end{cases}$$

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

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

as commonly done, then $\mathbf{u} \in \mathbb{R}^N$, where $N = n^2$. The discrete gradient $(\nabla u)_{i,j}$ can be expressed by a multiplication of the matrix $A_l^T \in \mathbb{R}^{2 \times N}$ ($l = 1, 2, \dots, N$) to the vector \mathbf{u} :

$$A_l^T \mathbf{u} = \begin{cases} (\mathbf{u}_{l+1} - \mathbf{u}_l; \mathbf{u}_{l+n} - \mathbf{u}_l) & \text{if } l \bmod n \neq 0 \text{ and } l + n \leq N \\ (0; \mathbf{u}_{l+n} - \mathbf{u}_l) & \text{if } l \bmod n = 0 \text{ and } l + n \leq N \\ (\mathbf{u}_{l+1} - \mathbf{u}_l; 0) & \text{if } l \bmod n \neq 0 \text{ and } l + n > N \\ (0; 0) & \text{if } l \bmod n = 0 \text{ and } l + n > N. \end{cases}$$

We also stack the grid functions $A_l^T \mathbf{u}$ along rows into a vector. We form the matrix A by concatenating the matrices A_l , $l = 1, 2, \dots, N$, that is,

$$A = (A_1, A_2, \dots, A_N) = (A_{1,1}, A_{1,2}, \dots, A_{N,1}, A_{N,2}) \in \mathbb{R}^{N \times 2N}.$$

Meanwhile, we form another two matrices A_x and A_y as follows:

$$A_x = (A_{1,1}, A_{2,1}, \dots, A_{N,1}) \in \mathbb{R}^{N \times N}, \quad A_y = (A_{1,2}, A_{2,2}, \dots, A_{N,2}) \in \mathbb{R}^{N \times N}.$$

In this notation, the gradient ∇u and the divergence $\nabla \cdot (\nabla u)$ are respectively $A^T \mathbf{u}$ and $-\sum_j A_j (A_j^T \mathbf{u})$. It is easy to see u_x , u_y , $u_{xxx} + u_{xyy}$, and $u_{yxx} + u_{yyy}$ are respectively $A_x^T \mathbf{u}$, $A_y^T \mathbf{u}$, $-\sum_j A_j (A_j^T (A_x^T \mathbf{u}))$ and $-\sum_j A_j (A_j^T (A_y^T \mathbf{u}))$. Denoting by $g_1(\mathbf{u})$ and $g_2(\mathbf{u})$ the discretization of $g_1(u)$ and $g_2(u)$, respectively, then we have

$$(11) \quad g_1(\mathbf{u}) = \alpha \left(\sum_j A_j \left(\frac{A_j^T \mathbf{u}}{\sqrt{|A_j^T \mathbf{u}|^2 + \beta}} \right) \right) + K^*(K\mathbf{u} - \mathbf{f}) = 0,$$

$$(12) \quad g_2(\mathbf{u}) = \alpha \left[A_x \left(\sum_j A_j \left(\frac{A_j^T (A_x^T \mathbf{u})}{\sqrt{|A_j^T (A_x^T \mathbf{u})|^2 + |A_j^T (A_y^T \mathbf{u})|^2 + \beta}} \right) \right) \right]$$

$$+A_y \left(\sum_j A_j \left(\frac{A_j^T(A_y^T \mathbf{u})}{\sqrt{|A_j^T(A_x^T \mathbf{u})|^2 + |A_j^T(A_y^T \mathbf{u})|^2 + \beta}} \right) \right) \Big] + K^*(K\mathbf{u} - \mathbf{f}) = 0.$$

We remark that there exist many competing algorithms [24] for solving (11) but no fast algorithms for (12).

Below we first discuss the fixed point type schemes before we introduce a homotopy scheme which is primarily for helping (12) but can also benefit (11).

2.1. Standard fixed point methods. A common technique for solving a nonlinear problem directly is by linearization. As far as the nonlinear equations (11) and (12) are concerned, a fixed point method must linearize some nonlinear terms by using the previous iterate $\mathbf{u}^{(k)}$ with $\mathbf{u}^{(0)} = \mathbf{f}$. The computational formula for (11) is

$$(13) \quad \alpha \left(\sum_j A_j \left(\frac{A_j^T \mathbf{u}^{(k+1)}}{\sqrt{|A_j^T \mathbf{u}^{(k)}|^2 + \beta}} \right) \right) + K^*(K\mathbf{u}^{(k+1)} - \mathbf{f}) = 0,$$

which is known to be convergent [24] for any β .

Motivated by the above idea, a natural fixed point method is the following

$$(14) \quad \alpha \left[A_x \left(\sum_j A_j \left(\frac{A_j^T(A_x^T \mathbf{u}^{(k+1)})}{\sqrt{|A_j^T(A_x^T \mathbf{u}^{(k)})|^2 + |A_j^T(A_y^T \mathbf{u}^{(k)})|^2 + \beta}} \right) \right) + A_y \left(\sum_j A_j \left(\frac{A_j^T(A_y^T \mathbf{u}^{(k+1)})}{\sqrt{|A_j^T(A_x^T \mathbf{u}^{(k)})|^2 + |A_j^T(A_y^T \mathbf{u}^{(k)})|^2 + \beta}} \right) \right) \right] + K^*(K\mathbf{u}^{(k+1)} - \mathbf{f}) = 0.$$

The above equation is equal to (see [22]):

$$(15) \quad M(\mathbf{u}^{(k)})\delta\mathbf{u}^{(k)} = -g_2(\mathbf{u}^{(k)}),$$

where $\delta\mathbf{u}^{(k)} = \mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}$ and

$$M(\mathbf{u}^{(k)}) = \alpha \left(A_x \sum_j \frac{A_j A_j^T}{\sqrt{|A_j^T(A_x^T \mathbf{u}^{(k)})|^2 + |A_j^T(A_y^T \mathbf{u}^{(k)})|^2 + \beta}} A_x^T + A_y \sum_j \frac{A_j A_j^T}{\sqrt{|A_j^T(A_x^T \mathbf{u}^{(k)})|^2 + |A_j^T(A_y^T \mathbf{u}^{(k)})|^2 + \beta}} A_y^T \right) + K^* K.$$

Here $M(\mathbf{u}^{(k)})$ is a symmetric positive definite matrix (see [23]), which guarantees the existence of $\delta\mathbf{u}^{(k)} = -M^{-1}(\mathbf{u}^{(k)})g_2(\mathbf{u}^{(k)})$ and moreover the linear system $M(\mathbf{u}^{(k)})\delta\mathbf{u}^{(k)} = -g_2(\mathbf{u}^{(k)})$ can be solved by suitable iterative solvers; here we use a preconditioned conjugate gradient method. Unfortunately the above linear systems are high singular for small β .

For now, we summarise how the above two fixed point schemes are implemented as algorithms in preparation for introducing our homotopy algorithm.

Algorithm 1. $[\mathbf{u}, \text{iter}] \leftarrow \text{FP1_method}(\mathbf{f}, \mathbf{u}^{(0)}, \alpha, \beta, \text{maxit}, \text{tol})$

step 1. Set $k := 0$.
step 2. Compute (13) to obtain $\mathbf{u}^{(k+1)}$, and then compute $\|g_1(\mathbf{u}^{(k+1)})\|_2$.
step 3. If $\|g_1(\mathbf{u}^{(k+1)})\|_2 \leq \text{tol}$ or $k = \text{maxit} - 1$, then
 record the iteration $\text{iter} := k + 1$, and return with $\mathbf{u} := \mathbf{u}^{(k+1)}$.
 Else set $k := k + 1$, then
 return to step 2.

Algorithm 2. $[\mathbf{u}, \text{iter}] \leftarrow \text{FP2_method}(\mathbf{f}, \mathbf{u}^{(0)}, \alpha, \beta, \text{maxit}, \text{tol})$

step 1. Set $k := 0$.
step 2. Compute (14) to obtain $\mathbf{u}^{(k+1)}$, and then compute $\|g_2(\mathbf{u}^{(k+1)})\|_2$.
step 3. If $\|g_2(\mathbf{u}^{(k+1)})\|_2 \leq \text{tol}$ or $k = \text{maxit} - 1$, then
 record the iteration $\text{iter} := k + 1$, and return with $\mathbf{u} := \mathbf{u}^{(k+1)}$.
 Else set $k := k + 1$, then
 return to step 2.

To give an indication of how sensitive Algorithm 2 for LLT model is about β , Table 1 shows with different β the number of fixed point (FP) steps and CPU time needed to achieve a reduction of the residual by a factor of 10^{-4} on a 128×128 “triangular” image (with the random noise having a signal to noise ratio (SNR) of 7.3008). It is clear to draw the conclusion that the smaller the β is, the higher the quality of the reconstruction of image edges and the more the iterations are. To speed up the convergence of the fixed point for the Euler-lagrange equations (12) with small β , the following section will introduce a homotopy method constructed by gradually decreasing parameter β to improve its convergence.

TABLE 1. Number of fixed point steps, CPU time and the signal to noise ratio (SNR) for LLT model on 128×128 “triangular” image with different β .

β	1.0	0.1	10^{-2}	10^{-4}	10^{-8}	10^{-12}
FP steps	19	25	32	49	116	238
CPU	8	11	14	21	55	546
SNR	9.1531	9.1862	9.1983	9.2021	9.2022	9.2022

Fixed point scheme 1. Recall that in Lundervold and Tai’s iterative method combining ROF model and LLT model [16] for image restoration, the time marching method is used. Since a time marching method should fulfill the Courant-Friedrichs-Lewy (or CFL) stability criterion, it is generally slow. Equipped with the above Algorithm 1 and Algorithm 2, we may replace their time marching method and obtain an iterative method based on the fixed point algorithm as follows.

Algorithm 3 (Fixed point scheme 1).

step 1. Given $k:=0$, and put $\mathbf{w}^k := \mathbf{v}^k := \mathbf{u}^k := \mathbf{f}$.
 step 2. Compute (13) to get \mathbf{v}^{k+1} ,
 Compute (14) to get \mathbf{u}^{k+1} ,
 Using \mathbf{w}^k to get θ by (7), set $\mathbf{w}^{k+1} = \theta\mathbf{v}^{k+1} + (1 - \theta)\mathbf{u}^{k+1}$.
 step 3. If $k = \text{maxit} - 1$, then
 return with $\mathbf{w} := \mathbf{w}^{k+1}$.
 Else set $k := k + 1$, $\mathbf{v}^k := \mathbf{w}^k$, $\mathbf{u}^k := \mathbf{w}^k$, then
 return to step 2.

For ease of comparison, we shall denote by LT1 for Lundervold and Tai’s original time marching based iterative method and by LT2 the above Algorithm 3.

2.2. Homotopy method for LLT model. In this section, a homotopy method is presented to solve the fourth order Euler-Lagrange equation (12) with the above fixed point method (Algorithm 2) as its correction. As a globally convergent method, the homotopy method has versatility and robustness, and it has become an important tool for solving nonlinear problems, see [12, 3, 11, 25, 1, 13]. As is known, the basic idea of a homotopy algorithm is to construct a continuous map $H(\mathbf{u}, t)$ which deforms a simple function $H(\mathbf{u}, 0)$ to the given function $H(\mathbf{u}, 1)$ as t varies from 0 to 1. Our continuous map is constructed by gradually reducing parameter β as follows:

$$\begin{aligned}
 & H(\mathbf{u}, t) \\
 (16) \quad & = \alpha \left[A_x \left(\sum_j A_j \left(\frac{tA_j^T(A_x^T \mathbf{u})}{\sqrt{t^2(|A_j^T(A_x^T \mathbf{u})|^2 + |A_j^T(A_y^T \mathbf{u})|^2) + (1-t)}} \right) \right) \right. \\
 & \quad \left. + A_y \left(\sum_j A_j \left(\frac{tA_j^T(A_y^T \mathbf{u})}{\sqrt{t^2(|A_j^T(A_x^T \mathbf{u})|^2 + |A_j^T(A_y^T \mathbf{u})|^2) + (1-t)}} \right) \right) \right] \\
 & + K^*(K\mathbf{u} - \mathbf{f}) = 0.
 \end{aligned}$$

The essence of a homotopy algorithm is to find the solution of the final function by tracking the zero curve Γ emanating from the solution of $H(\mathbf{u}, 0)$. We usually use a predictor-corrector path following method to get this solution. The whole process consists of a succession of the following two phases.

Predictor step. For $t_0 = 0$, the solution of $H(\mathbf{u}, t_0) = 0$ is known, namely $\mathbf{u}^0 = \mathbf{f}$. After we have obtained an approximate solution \mathbf{u}^{k-1} of $H(\mathbf{u}, t_{k-1})$ for some $t_{k-1} \in [0, 1)$, increase t with some predictor steplength h_{k-1} to reach $t_k = t_{k-1} + h_{k-1}$ and the solution of $H(\mathbf{u}, t_k) = 0$ is provided with the initial guess \mathbf{u}^{k-1} .

Corrector steps. From the initial point \mathbf{u}^{k-1} , approximately solve $H(\mathbf{u}, t_k) = 0$ by the fixed point method. Because the parameter t is introduced into (16) for compute $1/\sqrt{|A_j^T(A_x^T \mathbf{u})|^2 + |A_j^T(A_y^T \mathbf{u})|^2 + (1-t)/t^2}$, the fixed point method used here is slightly different from Algorithms 2 in replacing the parameter $\beta = \beta(t)$ by t .

As we know, the prediction phases and the correction phases mutually affect each other. The predictor steplength h is adjusted according to the performance of the corrector procedure as done below in Algorithm 4: when a corrector step terminates

within prescribed steps of $it1$, h is considered too small for the next predictor and is increased, when the iterations terminate over some $it2 > it1$ steps, h is considered too large and will be decreased. The procedure of predictor-corrector path following is shown as follows:

Algorithm 4. $\mathbf{u} \leftarrow \text{homotopy1}(\mathbf{f}, \alpha, \beta, \mu, \text{maxit}, \text{maxit}^*, \text{tol})$

step 1. Set $k := 1$, $\mathbf{u}^{k-1} := \mathbf{f}$, $t_{k-1} := 0$.
step 2. Set $t_k := t_{k-1} + \min(\mu, 0.5(1 - t_{k-1}))$.
 If $t_k \geq 2/(1 + \sqrt{1 + 4\beta})$, then
 $t_k = 2/(1 + \sqrt{1 + 4\beta})$,
 $\mathbf{u} \leftarrow \text{FP2_method}(\mathbf{f}, \mathbf{u}^{k-1}, \alpha, t_k, \text{maxit}^*, \text{tol})$.
 Return with \mathbf{u} .
 Else $[\mathbf{u}^k, \text{iter}] \leftarrow \text{FP2_method}(\mathbf{f}, \mathbf{u}^{k-1}, \alpha, t_k, \text{maxit}, \text{tol})$.
step 3. Set $k := k - 1$.
 If $\text{iter} \leq it1$, then
 set $\mu := \min(1.2\mu, 0.9)$.
 If $\text{iter} \geq it2$, then
 set $\mu := \mu/1.2$.
 If μ is unreasonably small, then
 return with an error flag.
 Return to step 2.

Here we may replace the step `FP2_method` by `FP1_method` if we hope to accelerate Algorithm 1. In this case, the continuous map for Algorithm 1 (i.e. TV model) is

$$(17) \quad F(\mathbf{u}, t) = \alpha \left(\sum_j A_j \left(\frac{t A_j^T \mathbf{u}}{\sqrt{t^2 |A_j^T \mathbf{u}|^2 + (1-t)}} \right) \right) + K^*(K\mathbf{u} - \mathbf{f}) = 0.$$

3. Alternative convex combinations of ROF model and LLT model

With our accelerated LLT method (Algorithm 4), we may propose different convex combinations of ROF model and LLT model using the same homotopy framework. It turns out that such new combinations are competitive in performance (see next section).

3.1. Homotopy fixed point scheme 2. Notice that before the final solution is obtained, a convex combination is done in each iteration for both the LT1 method and the LT2 method. Below we will introduce a simple approach which finds the solutions of (11) and (12) independent of each other in homotopy framework, and a convex combination of the two solutions is done at their convergence. Observe that θ plays an important role in the combination, we apply a simple iterative method which arises after each solution of (11) and (12). The aim is to update θ for the convex combination so that it can take the best out of each of both models efficiently. Assume that the required (small) β is prescribed, α_1 and α_2 are regularization parameters for (11) and (12), respectively. The details of our first alternative algorithm are given in the following

Algorithm 5 (Homotopy fixed point scheme 2).

step 1. Compute (11) by homotopy method, via (17), to get \mathbf{v} ,
 Compute (12) by homotopy method, via (16), to get \mathbf{u} .

step 2. Set $k:=0$, $\mathbf{w}^k := \mathbf{f}$ (or $\mathbf{w}^k := \mathbf{v}$, or $\mathbf{w}^k := \mathbf{u}$).

step 3. Using \mathbf{w}^k to get θ by (10),
 set $\mathbf{w}^{k+1} = \theta\mathbf{v} + (1 - \theta)\mathbf{u}$.

step 4. If $k = \text{maxit}$, then
 return with \mathbf{w}^{k+1} .
 Else set $k := k + 1$, then return to step 3.

Here we use the function (10) to get θ so function (7) offers another choice. Numerical tests indicate that θ obtained by (10) is more effective and accurate than that by (7) for this algorithm. Meanwhile, we remark that in step 3 the first θ is usually given by $\mathbf{w}^0 = \mathbf{f}$, but $\mathbf{w}^0 = \mathbf{v}$ is better for images composed of many flat subregions, and for images with many smooth subregions $\mathbf{w}^0 = \mathbf{u}$ would be better.

3.2. Homotopy fixed point scheme 3. In Algorithm 5, we solve ROF and LLT models separately before combining them. Below our second alternative algorithm consists of solving two homotopy equations (16) and (17) simultaneously over the complete homotopy path.

This is achieved by firstly reducing parameter β in (16) and (17) simultaneously with the formula $(1-t)/t^2$ as the homotopy parameter t varies from 0 to 1. Secondly, a convex combination is done after each correction and we allow both equations to start with the same initial solution for a new parameter t (i.e. new β).

The overall scheme is summarized by the following algorithm.

Algorithm 6 (Homotopy fixed point scheme 3).

$\mathbf{w} \leftarrow \text{homotopy2}(\mathbf{f}, \alpha_1, \alpha_2, \beta, \mu, \text{maxit}_1, \text{maxit}_2, \text{maxit}^*, \text{tol})$

step 1. Set $t_0 := 0$, $\mathbf{w}^0 := \mathbf{f}$.

step 2. Set $t := t_0 + \min(\mu, 0.5(1 - t_0))$, and $\beta_t := (1 - t)/t^2$.
 If $t \geq 2/(1 + \sqrt{1 + 4\beta})$, then
 $\beta_t = \beta$,
 $\mathbf{v} \leftarrow \text{FP1_method}(\mathbf{f}, \mathbf{w}^0, \alpha_1, \beta_t, \text{maxit}^*, \text{tol})$,
 $\mathbf{u} \leftarrow \text{FP2_method}(\mathbf{f}, \mathbf{w}^0, \alpha_2, \beta_t, \text{maxit}^*, \text{tol})$,
 using \mathbf{w}^0 to get θ by (10), set $\mathbf{w} = \theta\mathbf{v} + (1 - \theta)\mathbf{u}$.
 Else $[\mathbf{v}, \text{iter1}] \leftarrow \text{FP1_method}(\mathbf{f}, \mathbf{w}^0, \alpha_1, \beta_t, \text{maxit}_1, \text{tol})$,
 $[\mathbf{u}, \text{iter2}] \leftarrow \text{FP2_method}(\mathbf{f}, \mathbf{w}^0, \alpha_2, \beta_t, \text{maxit}_2, \text{tol})$,
 using \mathbf{w}^0 to get θ by (10), set $\mathbf{w}^* = \theta\mathbf{v} + (1 - \theta)\mathbf{u}$.

step 3. Set $(\mathbf{w}^0, t_0) := (\mathbf{w}^*, t)$, and $\text{iter} = \max(\text{iter1}, \text{iter2})$.
 If $\text{iter} \leq \text{it1}$, then
 set $\mu := \min(1.2\mu, 0.9)$.
 If $\text{iter} \geq \text{it2}$, then
 set $\mu := \mu/1.2$.
 If μ is unreasonably small, then
 return with an error flag.
 Return to step 2.

4. Numerical experiments and discussions

In this section we use the signal to noise ratio (SNR), and the \mathbb{L}^2 - norm of the difference between the recovered image and the original image to measure the quality of the restored images, both of them are defined by

$$\text{SNR} = \frac{\sum_{i=1}^n \sum_{j=1}^n u_{i,j}^2}{\sum_{i=1}^n \sum_{j=1}^n (u_{i,j} - \tilde{u}_{i,j})^2}, \quad \text{and} \quad \mathbb{L}^2\text{-norm} = \sum_{i=1}^n \sum_{j=1}^n (u_{i,j} - \tilde{u}_{i,j})^2,$$

where u and \tilde{u} are the original image and the restored image, respectively.

We test and compare our restoration algorithms on “Lena”, “Aircraft” and “Cameraman” images, with 256×256 pixels and an intensity range of $[0, 255]$. All of the original images are shown in Figure 1. The test images with their SNR and \mathbb{L}^2 -norm are listed in Table 2. The corresponding regularization parameters used in the following tests can also be seen in Table 2. For ease of distinction, we use α_1 for ROF model, α_2 for LLT model and α for CTX model (see [9]). Here “Lena I” and “Lena II” (“Aircraft I” and “Aircraft II”) are noisy “Lena” (“Aircraft”) images with weak and strong noise respectively, and “Cameraman” is contaminated with Gaussian noise and Gaussian blur.

TABLE 2. SNR, \mathbb{L}^2 -norm and the corresponding regularizing parameters for each noisy image.

noisy image f	SNR of f	\mathbb{L}^2 - norm	regularize parameters
Lena I	40	4.5348e+003	$\alpha_1 = 6, \alpha_2 = 4$
Lena II	8	1.0141e+004	$\alpha_1 = 14, \alpha_2 = 10$
Aircraft I	50	4.4219e+003	$\alpha_1 = 6, \alpha_2 = 4, \alpha = 4$
Aircraft II	10	9.8877e+003	$\alpha_1 = 15, \alpha_2 = 10, \alpha = 10$
Cameraman	20	7.6870e+003	$\alpha_1 = 8, \alpha_2 = 3$

We aim to compare both the restoration quality (to highlight the usefulness of combined models) as well as the solution speed (to highlight the advantage of our homotopy algorithms).

4.1. Comparisons of Algorithm 4 with time marching (TM) method and fixed point method for LLT model for image denoising. To compare the effect of Algorithm 4 for LLT model, we conduct our experiments on the standard “Lena” image which is well suited for processing images with smooth transitions. We take $\beta = 10^{-8}$ for all algorithms in this section, and set $\mu = 0.2$ for our homotopy method. The maximum iteration steps for a time marching method (i.e. a gradient descent method) are 10000 for “Lena I” and 40000 for “Lena II”, and the tolerance for fixed point methods is 10^{-4} . The summary of computational results is listed in Table 3.

As far as the quality of the denoised images is concerned, the fixed point method and Algorithm 4 are better than time marching methods and the same result can be seen in Figures 2 and 3. And we also see that our Algorithm 4 is about four times

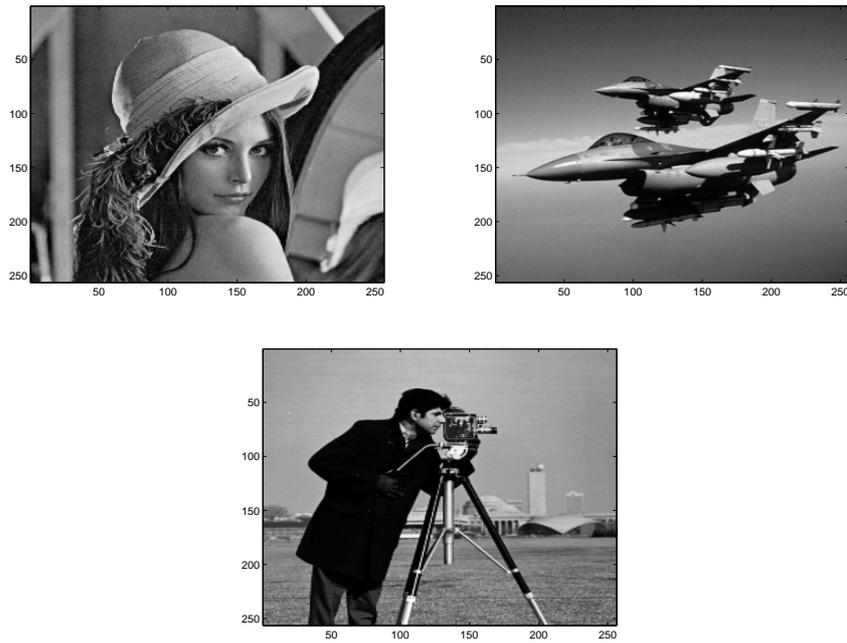


FIGURE 1. Above Left Plot: The original “Lena” image. Above Right Plot: The original “Aircraft” image. Bottom Plot: The original “Cameraman” image.

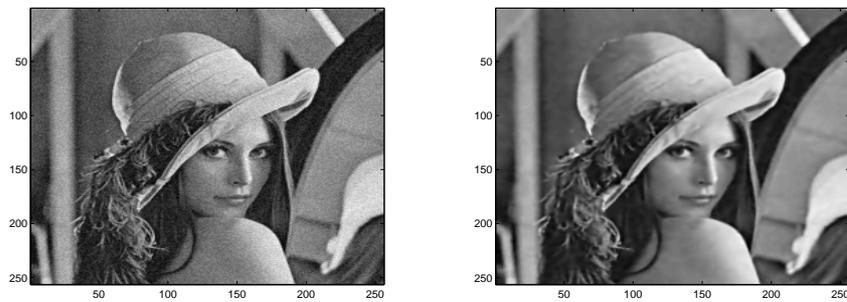


FIGURE 2. Left Plot: Noisy image of “Lena I”. Right Plot: Image recovered by time marching method with SNR=46.5281.

more efficient than the fixed point method and five times than the time marching for this test.

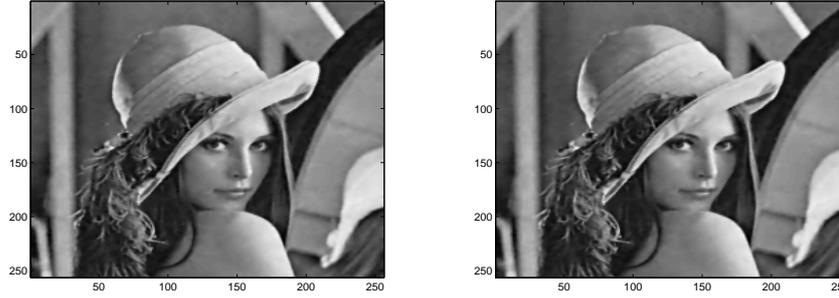


FIGURE 3. Left Plot: Image recovered by fixed point method with SNR=47.4484. Right Plot: By Algorithm 4 with SNR=47.4485.

TABLE 3. Comparison of the time marching (TM) method, fixed point (FP) method, and Algorithm 4 for the same LLT model on “Lena I” and “Lena II”.

method	Lena I			Lena II		
	SNR	$\mathbb{L}^2 - \text{norm}$	CPU	SNR	$\mathbb{L}^2 - \text{norm}$	CPU
TM method	46.5281	4.2048e+003	974	10.0139	9.0636e+003	3807
FP method	47.4484	4.1638e+003	829	10.0358	9.0537e+003	3606
Algorithm 4	47.4485	4.1638e+003	212	10.0358	9.0537e+003	356

4.2. Comparisons of new algorithms with ROF model and LLT model.

We do our first comparison for pure image denoising problem. Due to ROF model does well in “blocky” images and LLT model works almost perfectly for smooth images, we still choose the standard “Lena” image as the test image which is composed of flat subregions, subregions with a smooth change in intensity value and jumps. We take the parameter $\beta = 10^{-6}$ for all these comparison algorithms. Table 4 shows numerical results on SNR and the \mathbb{L}^2 -norm of the difference between the denoised image and the true image and CPU time needed for each algorithms.

It is evident to draw the conclusion that our proposed methods do much better than ROF model and LLT model for “Lena I” and “Lena II”, and they do well in recovering both of jumps and smooth signals by observing the flat subregion and smooth subregion of the “Lena” image (see Figures 4-12). We can also see the strengths and weakness about both ROF model and LLT model.

TABLE 4. Comparison of ROF model, LLT model, Algorithm 5, and Algorithm 6 for “Lena I” and “Lena II”.

method	Lena I			Lena II		
	SNR	$\mathbb{L}^2 - \text{norm}$	CPU	SNR	$\mathbb{L}^2 - \text{norm}$	CPU
ROF model	47.5896	4.1576e+003	87	10.0479	9.0482e+003	91
LLT model	47.4485	4.1638e+003	89	10.0358	9.0537e+003	138
Algorithm 5	48.1440	4.1336e+003	173	10.0657	9.0402e+003	226
Algorithm 6	47.9564	4.1417e+003	83	10.0791	9.0342e+003	182

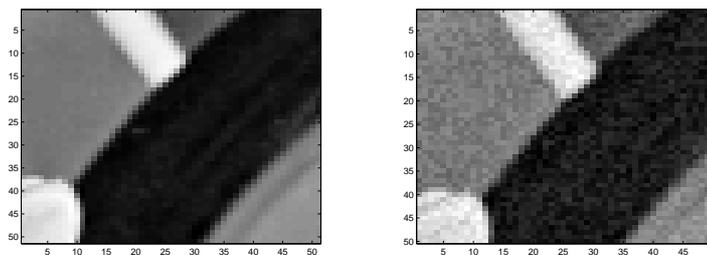


FIGURE 4. Left Plot: The flat subregion of “Lena” image. Right Plot: The subregion of “Lena I”.

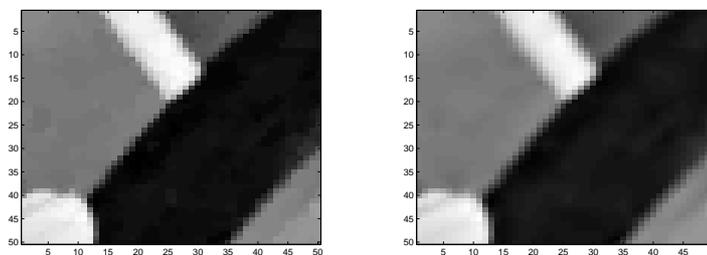


FIGURE 5. Left Plot: The subregion recovered by ROF model. Right Plot: By LLT model.

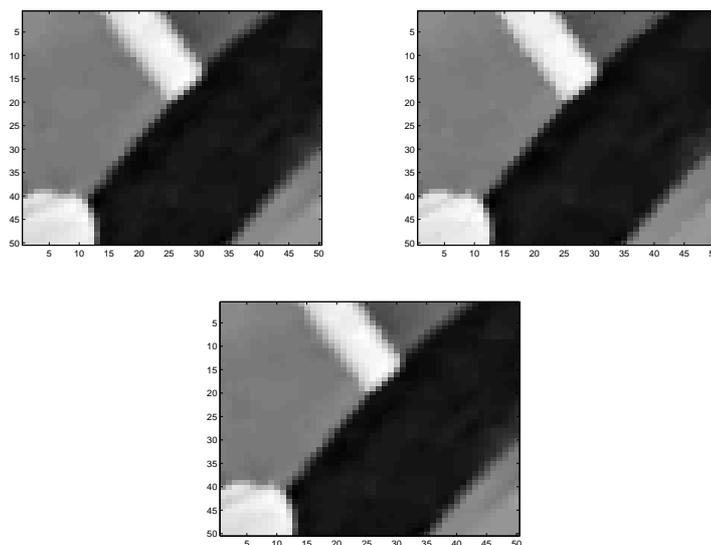


FIGURE 6. Left and Middle Plots: The subregion recovered by LT2 method and Algorithm 5. Right Plot: By Algorithm 6.

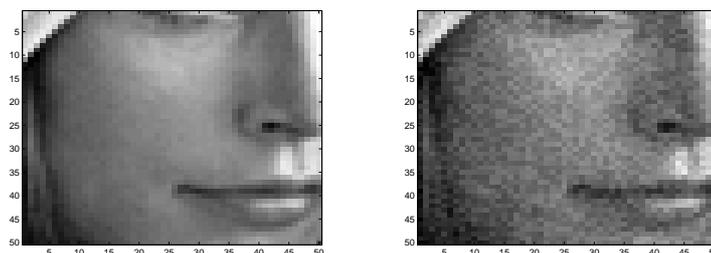


FIGURE 7. Left Plot: The smooth subregion of “Lena” image. Right Plot: The subregion of “Lena I”.

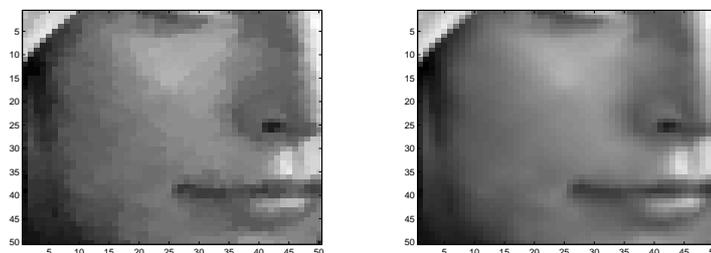


FIGURE 8. Left Plot: The subregion recovered by ROF model. Right Plot: By LLT model.

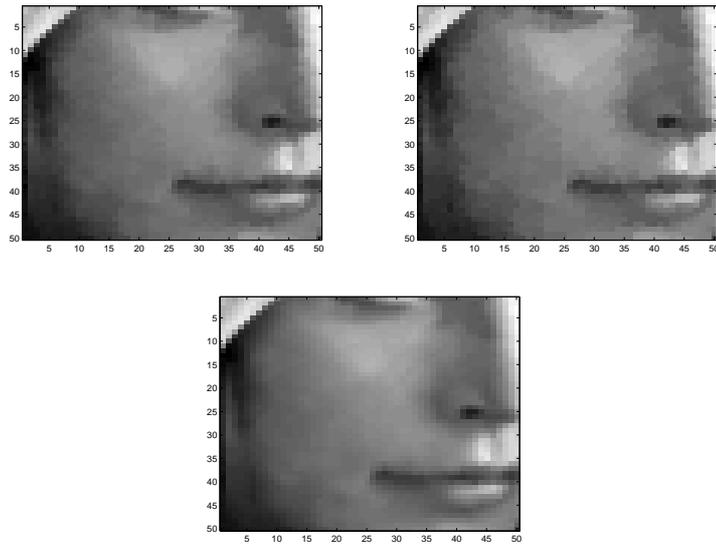


FIGURE 9. Left and Middle Plots: The subregion recovered by LT2 method and Algorithm 5. Right Plot: By Algorithm 6.

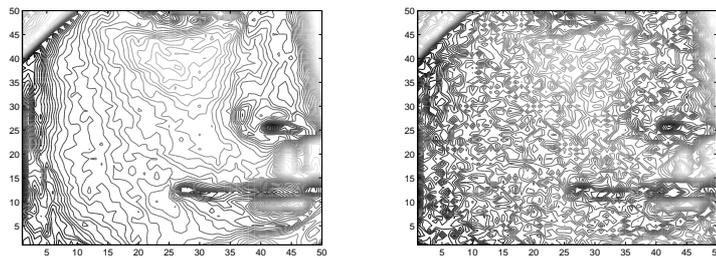


FIGURE 10. Left Plot: The contour of smooth subregion of Lena image. Right Plot: The contour of subregion of “Lena I”.

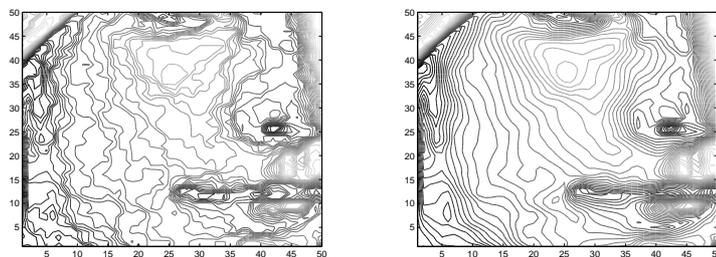


FIGURE 11. Left Plot: The contour of subregion recovered by ROF model. Right Plot: by LLT model.

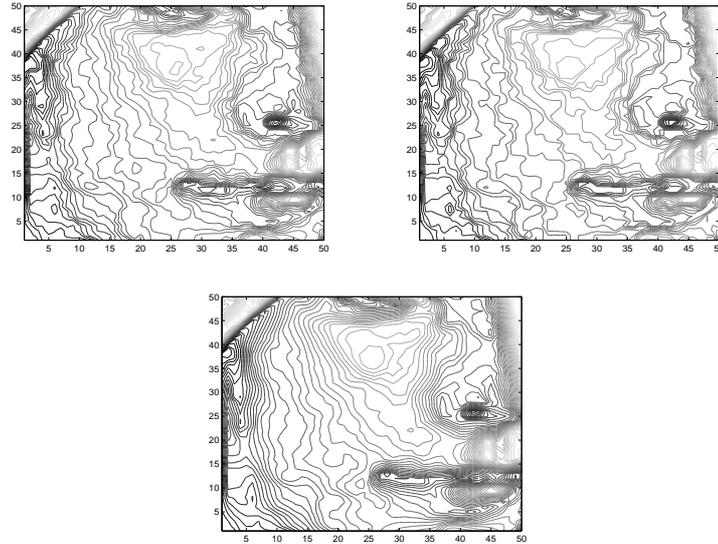


FIGURE 12. Left and Middle Plots: The contour of subregion recovered by LT2 method and Algorithm 5. Right Plot: By Algorithm 6.

Our next comparison is for image deblurring and uses the “Cameraman” image added with Gaussian noise and Gaussian blur; see Figure 13. Table 5 reports the numerical results for each algorithm. According to Table 5, as visualized in Figures 14-15, we find that images recovered by Algorithms 5 and 6 have better quality than that recovered by ROF and LLT models. This observation is similar to the denoising case discussed.

TABLE 5. Comparison of ROF model, LLT model, Algorithm 5 and Algorithm 6 for image restoration.

method	SNR	L^2 - norm	CPU
ROF model	45.0604	5.1140e+003	361
LLT model	44.1073	5.1690e+003	395
Algorithm 5	45.5557	5.0861e+003	754
Algorithm 6	45.1404	5.1095e+003	643

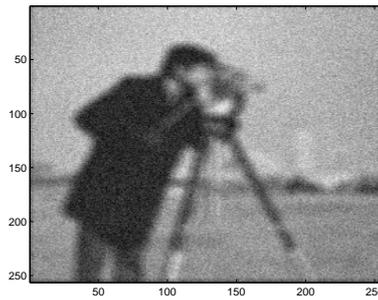


FIGURE 13. The “Cameraman” image degraded by Gaussian noise and Gaussian blur with $\text{SNR}=20$ and $L^2\text{-norm}=7.6870e+003$.

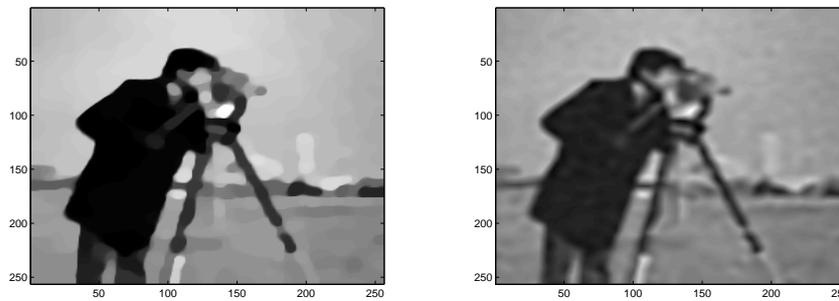


FIGURE 14. Left Plot: Image restored by ROF model with $\text{SNR}=45.0604$. Right Plot: By LLT model with $\text{SNR}=44.1073$.

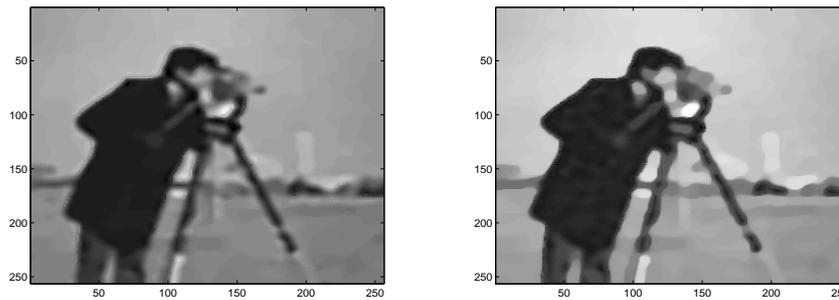


FIGURE 15. Left Plot: Image restored by Algorithm 5 with $\text{SNR}=45.5557$. Right Plot: By Algorithm 6 with $\text{SNR}=45.1404$.

4.3. Comparisons of new convex combination methods with others. In this section, we conduct further comparisons on “Aircraft I” and “Aircraft II” (see Figure 16 left and Figure 19 left). The parameter β is still taken 10^{-6} for all cases. LT1 method terminates by the maximum iteration steps 500, both CTX model and

LT2 method terminate by 100 steps. Table 6 reports the numerical results of all methods. We find our methods attains much better quality of image restorations than LT1 method , CTX model and LT2 method in comparable complexity.

TABLE 6. Comparison of CTX model, LT1 method, LT2 method, Algorithm 5, and Algorithm 6 for “Aircraft I” and “Aircraft II” .

method	Aircraft I			Aircraft II		
	SNR	$\mathbb{L}^2 - \text{norm}$	CPU	SNR	$\mathbb{L}^2 - \text{norm}$	CPU
LT1 method	58.8291	4.0766e+003	212	12.8078	8.7369e+003	211
CTX model	61.8681	3.9752e+003	253	12.7083	8.7710e+003	271
LT2 method	62.5060	3.9549e+003	284	12.8244	8.7312e+003	309
Algorithm 5	62.9111	3.9421e+003	195	12.9328	8.6946e+003	234
Algorithm 6	62.7001	3.9488e+003	101	12.8320	8.7286e+003	191

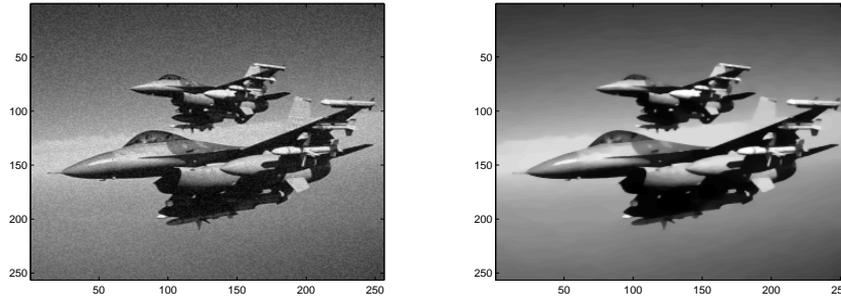


FIGURE 16. Left Plot: Noisy image of “Aircraft I”. Right Plot: Image recovered by LT1 method with SNR=58.8291.

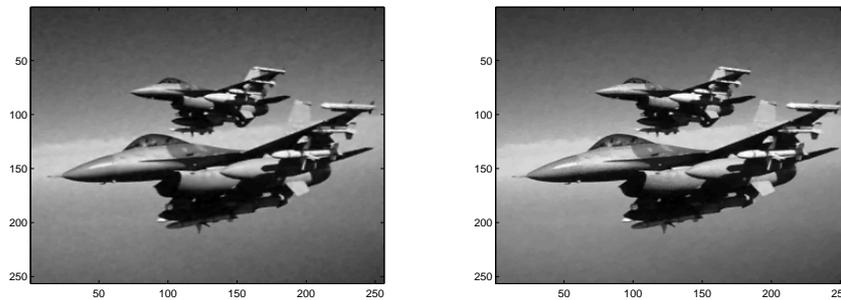


FIGURE 17. Left Plot: Images recovered by CTX model with SNR=61.8681. Right Plot: By LT2 method with SNR=62.5060.

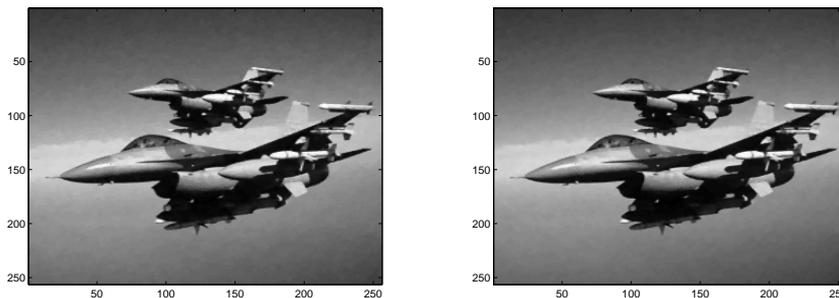


FIGURE 18. Left Plot: Images recovered by algorithm 5 with SNR=62.9111. Right Plot: By Algorithm 6 with SNR=62.7001.

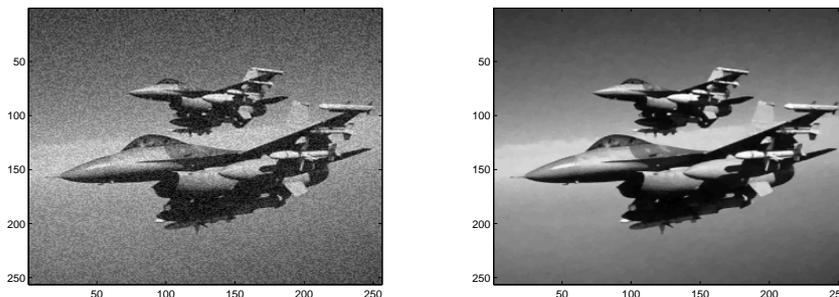


FIGURE 19. Left Plot: Noisy image of "Aircraft II". Right Plot: Image recovered by LT1 method with SNR=12.8078.

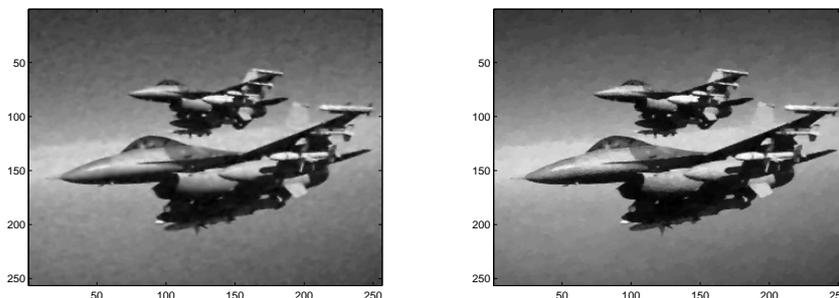


FIGURE 20. Left Plot: Images recovered by CTX model with SNR=12.7083. Right Plot: By LT2 method with SNR=12.8244.

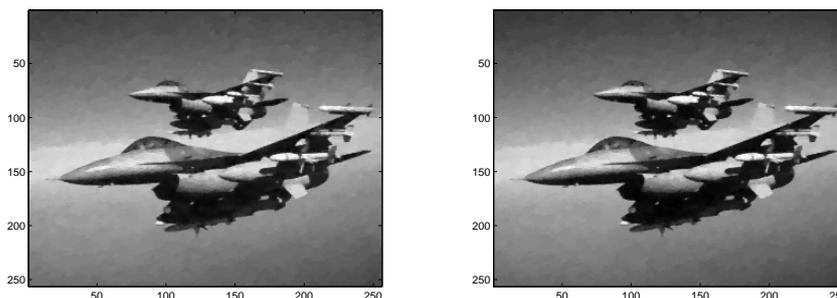


FIGURE 21. Left Plot: Images recovered by Algorithm 5 with SNR=12.9328. Right Plot: By Algorithm 6 with SNR=12.8320.

5. Conclusions

Image restoration combining total variation minimization and a second-order functional can restore effectively both the blocky subregion (of piecewise constant intensities) and smooth subregion (with no clear jumps) of an image. The resulting model can inherit the advantages of the ROF model and the LLT model, and avoid the disadvantages of both models. Construction of an effective combination and finding an efficient algorithm are both important tasks. This paper first gave a homotopy based fixed point method for *directly* solving the fourth order LLT model by curve tracking to adaptively choose the regularizing parameter β to achieve a fast convergence, and then presented new alternative combination schemes for image restoration to take the best out of ROF model, LLT model and the homotopy robustness to recover both jumps and smooth signals accurately. The resulting methods turn out to be quite effective for test images. Numerical experiments can demonstrate advantages of our methods over image iterative restoration method proposed in [16] and CTX model in [9]. Future work will consider how to apply our methods to other models and applications where combination may be used.

References

- [1] Eugene L. Allgower and Kurt Georg, Numerical Path Following, North-Holland, Amsterdam, 1997.
- [2] Carlos Brito-Loeza and Ke Chen, Multigrid algorithm for high order denoising, *SIAM J. Imaging Sciences*, No. 3, 3(2010), pp. 363-389.
- [3] 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), pp. 887-899.
- [4] Antonin Chambolle and Pierre-Louis Lions, Image recovery via total variation minimization and related problems, *Numer. Math.*, 76(1997), pp. 167-188.
- [5] Antonin Chambolle, An algorithm for total variation minimization and applications, *J. Math. Imaging Vis.*, 20(2004), pp. 89-97.
- [6] Tony F. Chan, Gene H. Golub and Pep Mulet, A nonlinear primal-dual method for total variation-based image restoration, *SIAM J. Sci. Comput.*, No. 6, 20(1999), pp. 1964-1977.
- [7] Tony F. Chan, Antonio Marquina and Pep Mulet, High-order total variation-based image restoration, *SIAM J. Sci. Comput.*, 22(2000), pp. 503-516.
- [8] Tony F. Chan and J. H. Shen, Image Processing and Analysis-Variational, PDE, Wavelet, And Stochastic Methods, SIAM, Philadelphia, 2005.
- [9] Qianshun Chang, Xue-Cheng Tai and Lily Xing, A compound algorithm of denoising using second-order and fourth-order partial differential equations, *Numer. Math. Theor. Meth. Appl.*, No.4, 2(2009), pp. 353-376.

- [10] S. Didas, B. Burgeth, A. Imiya and J. Weickert, Regularity and Scale-Space Properties of Fractional High Order Linear Filtering, Scale-Space and PDE Methods in Computer Vision, volume 3459 of Lecture Notes in Computer Science, Springer, Berlin, 2005.
- [11] C. B. Garcia and W. I. Zangwill, Pathways to Solutions, Fixed Points and Equilibria, Prentice-Hall, Englewood Cliffs, N.J., 1981.
- [12] R. B. Kellogg, T. Y. Li and J. A. Yorke, A constructive proof of the Brouwer fixed-point theorem and computational results, *SIAM J. Numerical Analysis*, 18(1976), pp. 473-483.
- [13] Zhenghua Lin, Bo Yu and Daoli Zhu, A continuation method for solving fixed points of self-mappings in general nonconvex sets, *Nonlinear Analysis*, 52(2003), pp. 905-915.
- [14] M. Lysaker, Arvid Lundervold and Xue-Cheng Tai, Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time, *IEEE Transactions on Image Processing*, No. 12, 12(2003), pp.1579-1590.
- [15] M. Lysaker, Stanley Osher and Xue-Cheng Tai, Noise removal using smoothed normals and surface fitting, *IEEE Transactions on Image Processing*, No. 10, 13(2004), pp. 1345-1357.
- [16] M. Lysaker and Xue-Cheng Tai, Iterative image restoration combining total variation minimization and a second-order functional, *Int. J. Comput. Vision*, No. 1, 66(2006), pp. 5-18.
- [17] Antonio Marquina and Stanley 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), pp. 387-405.
- [18] Stanley Osher and Ronald Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York, 2003.
- [19] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu and Wotao Yin, An iterative regularization method for total variation-based image restoration, *Multiscale Modeling and Simulation*, No. 2, 4(2005), pp. 460-489.
- [20] Leonid I. Rudin, Stanley Osher and Emad Fatemi, Nonlinear total variation based noise removal algorithms, *Phys. D*, 60(1992), pp. 259-268.
- [21] Joseph Savage and Ke Chen, On multigrids for solving a class of improved total variation based staircasing reduction models, *Springer-Verlag, Berlin*, (2007), pp. 69-94.
- [22] C. R. Vogel and M. E. Oman, Iterative methods for total variation denoising, *SIAM J. Sci. Comput.*, 17(1996), pp. 227-238.
- [23] C. R. Vogel and M. E. Oman, Fast, robust total variation-based reconstruction of noisy, blurred images, *IEEE Transactions on Image Processing*, No. 6, 7(1998), pp. 813-824.
- [24] C. R. Vogel, Computational Methods for Inverse Problems, SIAM, Philadelphia, 2002.
- [25] L. T. Watson, C. Billups and P. Morgan, Algorithm 652 hompack: A suite of codes for globally convergent homotopy algorithms, *ACM Transactions on Mathematical software*, No. 3, 13(1987), pp. 281-310.
- [26] Wei Zhu and Tony F. Chan, Image denoising using mean curvature, <http://www.math.nyu.edu/~wzhu/>, (2008), *Preprint*.

School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning 116024, PR China

E-mail: yangfenlinlin@126.com

Centre for Mathematical Imaging Techniques and Department of Mathematical Sciences, The University of Liverpool, Peach Street, Liverpool L69 7ZL, United Kingdom.

E-mail: k.chen@liverpool.ac.uk.

URL: <http://www.liv.ac.uk/~cmchenke/cmit>

School of Mathematical Sciences, Dalian University of Technology, Dalian, Liaoning 116024, PR China

E-mail: yubo@dlut.edu.cn

URL: <http://math.dlut.edu.cn/>