Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2017.m1611

Vol. **10**, No. 1, pp. 98-115
February 2017

Fast Linearized Augmented Lagrangian Method for Euler's Elastica Model

Jun Zhang[1,2,*], Rongliang Chen[3], Chengzhi Deng[1] and Shengqian Wang[1]

[1] *Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing, Nanchang Institute of Technology, Nanchang 330099, Jiangxi, China.*
[2] *College of Science, Nanchang Institute of Technology, Nanchang 330099, Jiangxi, China.*
[3] *Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, P. R. China.*

Received 24 February 2016; Accepted 8 July 2016

Abstract. Recently, many variational models involving high order derivatives have been widely used in image processing, because they can reduce staircase effects during noise elimination. However, it is very challenging to construct efficient algorithms to obtain the minimizers of original high order functionals. In this paper, we propose a new linearized augmented Lagrangian method for Euler's elastica image denoising model. We detail the procedures of finding the saddle-points of the augmented Lagrangian functional. Instead of solving associated linear systems by FFT or linear iterative methods (e.g., the Gauss-Seidel method), we adopt a linearized strategy to get an iteration sequence so as to reduce computational cost. In addition, we give some simple complexity analysis for the proposed method. Experimental results with comparison to the previous method are supplied to demonstrate the efficiency of the proposed method, and indicate that such a linearized augmented Lagrangian method is more suitable to deal with large-sized images.

AMS subject classifications: 65M55; 68U10; 94A08

Key words: Image denoising, Euler's elastica model, linearized augmented Lagrangian method, shrink operator, closed form solution.

1. Introduction

Image denoising aims to recover a noise-free image u from a noise-polluted image f. In general, it can be modeled by $f = u + \eta$, where η is the unknown noise. It has been a challenging topic and has been deeply investigated during the last two decades.

*Corresponding author. *Email addresses:* junzhang0805@126.com (J. Zhang), rl.chen@siat.ac.cn (R.-L. Chen), dengcz@nit.edu.cn (C.-Z. Deng), sqwang113@263.net (S.-Q. Wang)

 ©2017 Global-Science Press

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

Excellent results have been obtained by using the total variation (TV) model, which involves solving a second-order partial differential equation (PDE). This model was proposed by Rudin, Osher and Fatemi [21] (called the ROF model). Recently, the gap between a continuous ROF model and its discretized version was studied in [14].

The traditional way to solve the ROF model is to solve the corresponding Euler-Lagrange equation with some time marching methods [25]. Due to the stability constraint on time step size, this kind of methods usually converges slowly. To overcome the non-differentiability of the ROF model and reduce computational cost, several fast algorithms have been studied. For example, the idea of duality was first proposed by Chan *et al.* [5], later by Carter [2] and Chambolle [3, 4], and then by Zhu [29], Zhu and Chan [30], Zhu *et al.* [31]. In [11], Goldstein and Osher developed a split Bregman method where the Gauss-Seidel iteration was used to solve some linear systems. If the periodic boundary condition is used, such linear systems generated by the split Bregman method can be solved by fast Fourier transform (FFT) [27]. In [24], the authors introduced an augmented Lagrangian method for solving the ROF model. More specifically, it was shown in [26] that the CGM method, Chambolle's dual algorithm, the split Bregman method and the augmented Lagrangian method were equivalent.

Although the CGM method, the split Bregman method and the augmented Lagrangian method can deal with the ROF model efficiently, associated linear systems still need to be solved by FFT or the Gauss-Seidel method. In order to avoid solving a linear equation system and simultaneously obtain a closed form solution, Jia *et al.* proposed an efficient algorithm by applying matrix-vector multiplication [16]. Based on the same idea, that is, avoiding solving PDEs by FFT or linear iterative methods, Duan and Huang adopted a fixed-point strategy and proposed a fixed-point augmented Lagrangian method for TV minimization problems [9].

By using the ROF model, the edges and the small scale characteristics can be preserved during noise elimination. So this model has been extensively used for a variety of image restoration problems (see [6, 20]). However, the ROF model often causes staircase effects in the results. To overcome this difficulty, some high order models [8, 17–19, 22, 28, 32] have been introduced in image restoration recently. Also, Hahn *et al.* did some research on reducing staircase effects [1, 12, 13]. In [22], the following minimization problem based on a fourth-order PDE (called the Euler's elastica model) has been proposed:

$$\min_u \int_\Omega \left(a + b \left(\nabla \cdot \frac{\nabla u}{\mid \nabla u \mid} \right)^2 \right) \mid \nabla u \mid + \frac{\lambda}{2} \int_\Omega (u - f)^2, \tag{1.1}$$

where $\lambda > 0$ is a weighting parameter which controls the amount of denoising, and the term $\nabla \cdot \frac{\nabla u}{|\nabla u|}$ is the curvature of level curve $u(x, y) = c$. The use of fourth-order derivatives damps out high frequency components of images faster than the second-order PDE-based methods, so (1.1) can reduce the staircase effects and produce better approximation to the nature image. Indeed, it is also able to preserve object edges while erasing noise. The Euler's elastica model is one of the most important high order

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

models. It has been successfully applied in various problems, such as image denoising, inpainting and zooming.

However, due to the high nonlinearity of related Euler-Lagrange equation as in [22], the traditional ways to solve the Euler's elastica model (1.1) are usually time-consuming. In [23], the authors successfully introduced an augmented Lagrangian method to solve this model. In their method, the original unconstrained problem (1.1) was first transformed into a constrained optimization problem. They then concentrated on finding the saddle-points of the corresponding augmented Lagrangian functional. To this end, they solved several subproblems alternatively. Some subproblems have closed form solutions and thus can be easily solved. But there are still some other subproblems which need to be solved by applying techniques such as the FFT. With the aim of using FFT to solve one of those subproblems, a frozen coefficient method was applied to the coupled PDEs and this increased the computational cost. Moreover, if the boundary condition assigned to the minimization problem (1.1) is not periodic, it is not possible to use FFT. Last but not the least, since it is not necessary to solve the subproblems exactly for each iteration, we can use some inexpensive methods instead of FFT. Based on these observations, in [10], Duan *et al.* introduced a new variable into the constrained problem to remove the variable coefficient in one subproblem, and employed a Gauss-Seidel method to solve related Euler-Lagrange equations. Compared with the previous method, their method can reduce computational cost and can be applied on a computational domain with non-periodic boundary. However, two subproblems still needed to be solved by linear iterative method and didn't have closed form solutions.

Recently, from the same point of view (that is, avoiding solving the Euler-Lagrange equations of several subproblems by iterative methods), the linearized technique has been successfully employed to deal with quadratic penalty terms in many optimization problems. For instance, in [7], a linearized alternating direction method was applied to multiplicative noisy image denoising. In [15], Jeong *et al.* proposed a linearized alternating direction method for Poisson image restoration.

Inspired by the success of the linearized technique for dealing with these image processing problems, we check whether the pursuit of the solutions of linear equation systems can be replaced by adopting the linearized technique to approximate so as to reduce computational cost. Then based on the linearized technique, in this paper, we propose a new linearized augmented Lagrangian method for solving the Euler's elastica model (1.1). At each iteration of the proposed method, two quadratic penalty terms are linearized. Owing to the use of this linearized technique, the solutions of all subproblems have closed form expressions and thus each subproblem can be easily solved. Numerical results illustrate the efficiency of the proposed method. In particular, the newly developed method is more suitable to deal with large-sized images. Furthermore, the plots of relative residuals and errors show the convergence of the proposed method.

The organization of this paper is as follows. In Section 2, we review the previous augmented Lagrangian method for solving the Euler's elastica image denoising model (1.1). The newly developed linearized augmented Lagrangian method for solving this

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

problem is introduced in Section 3. We adopt a linearized strategy to approximate two quadratic penalty terms and study the complexity of the proposed method in this section. In Section 4, some numerical results are reported to show the efficiency of the proposed method. We end the paper with some conclusions in Section 5.

2. The augmented Lagrangian method

As is well-known, the augmented Lagrangian method (ALM) is a fast method for solving many regularization problems. It has been successfully employed in the minimization of non-differentiable and high order functionals [23,24,26,33]. This is noted that non-differentiable and high order functionals may cause some numerical difficulties, but the ALM is capable of handling these. At present, ALM has been widely applied in many fields.

Before reviewing related work on the ALM for the Euler's elastica model (1.1), we first list a definition and a lemma.

Definition 2.1. *The shrink operator $S_\tau(\cdot) : X \to X$ is defined by*

$$S_\tau(x) := \max\left\{0, 1 - \frac{\tau}{|x|}\right\} x, \tag{2.1}$$

where $\tau > 0$ is a given constant.

Lemma 2.1. ([23]) *Let x_0 be a given scalar or vector, μ, r be two parameters with $r > 0$, then the minimizer of the following functional*

$$\min_x \int_\Omega \mu|x| + \frac{r}{2}|x - x_0|^2$$

has an explicit solution

$$x = S_{\frac{\mu}{r}}(x_0). \tag{2.2}$$

It is easy to see that problem (1.1) is equivalent to the following constrained optimization problem:

$$\begin{aligned} &\min_{u,\mathbf{p},\mathbf{n},h} \int_\Omega (a + bh^2)\,|\mathbf{p}| + \frac{\lambda}{2}\int_\Omega (u - f)^2, \\ &s.t. \quad \mathbf{p} = \nabla u, \quad \mathbf{n} = \frac{\mathbf{p}}{|\mathbf{p}|}, \quad h = \nabla \cdot \mathbf{n}. \end{aligned} \tag{2.3}$$

Note that the second constraint in (2.3) is equivalent to $\mathbf{p} = |\mathbf{p}|\mathbf{n}$. Therefore, the above constrained problem can be reformulated as follows:

$$\begin{aligned} &\min_{u,\mathbf{p},\mathbf{n},h} \int_\Omega (a + bh^2)\,|\mathbf{p}| + \frac{\lambda}{2}\int_\Omega (u - f)^2, \\ &s.t. \quad \mathbf{p} = \nabla u, \quad \mathbf{p} = |\mathbf{p}|\mathbf{n}, \quad h = \nabla \cdot \mathbf{n}. \end{aligned} \tag{2.4}$$

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

The augmented Lagrangian functional for (2.4) is defined as:

$$\begin{aligned}&\mathcal{L}(u,\mathbf{p},\mathbf{n},h;\lambda_1,\lambda_2,\lambda_3)\\ =&\int_\Omega (a+bh^2)\,|\mathbf{p}| + \frac{\lambda}{2}\int_\Omega (u-f)^2 + \int_\Omega \lambda_1\cdot(\mathbf{p}-|\mathbf{p}|\mathbf{n}) + \frac{r_1}{2}\int_\Omega |\mathbf{p}-|\mathbf{p}|\mathbf{n}|^2 \\ &+\int_\Omega \lambda_2\cdot(\mathbf{p}-\nabla u) + \frac{r_2}{2}\int_\Omega |\mathbf{p}-\nabla u|^2 + \int_\Omega \lambda_3(h-\nabla\cdot\mathbf{n}) + \frac{r_3}{2}\int_\Omega (h-\nabla\cdot\mathbf{n})^2,\end{aligned} \tag{2.5}$$

where $\mathbf{p},\mathbf{n}\in\mathbb{R}^2$ are auxiliary vectors, $\lambda_1,\lambda_2\in\mathbb{R}^2$, and $\lambda_3\in\mathbb{R}$ are Lagrangian multipliers. Note that the saddle-points of the functional (2.5) correspond to the minimizers of the constrained optimization problem (2.3) and the minimizers of the Euler's elastica model (1.1). Hence one just needs to find the saddle-points of (2.5). To this end, an alternative minimization procedure is adopted as in [10]:

$$u^{k+1} = \arg\min_u \mathcal{L}(u,\mathbf{p}^k,\mathbf{n}^k,h^k;\lambda_1^k,\lambda_2^k,\lambda_3^k), \tag{2.6a}$$
$$\mathbf{p}^{k+1} = \arg\min_{\mathbf{p}} \mathcal{L}(u^{k+1},\mathbf{p},\mathbf{n}^k,h^k;\lambda_1^k,\lambda_2^k,\lambda_3^k), \tag{2.6b}$$
$$\mathbf{n}^{k+1} = \arg\min_{\mathbf{n}} \mathcal{L}(u^{k+1},\mathbf{p}^{k+1},\mathbf{n},h^k;\lambda_1^k,\lambda_2^k,\lambda_3^k), \tag{2.6c}$$
$$h^{k+1} = \arg\min_h \mathcal{L}(u^{k+1},\mathbf{p}^{k+1},\mathbf{n}^{k+1},h;\lambda_1^k,\lambda_2^k,\lambda_3^k), \tag{2.6d}$$
$$\lambda_1^{k+1} = \lambda_1^k + r_1(\mathbf{p}^{k+1}-|\mathbf{p}^{k+1}|\mathbf{n}^{k+1}), \tag{2.6e}$$
$$\lambda_2^{k+1} = \lambda_2^k + r_2(\mathbf{p}^{k+1}-\nabla u^{k+1}), \tag{2.6f}$$
$$\lambda_3^{k+1} = \lambda_3^k + r_3(h^{k+1}-\nabla\cdot\mathbf{n}^{k+1}). \tag{2.6g}$$

For the **p**-subproblem (the second problem of (2.6)), due to the presence of the nonlinear and non-differentiable quadratic penalty term involving $|\mathbf{p}|$, it is difficult to solve. Therefore, as discussed in [10], $\mathbf{p}-|\mathbf{p}|\mathbf{n}$ in the quadratic penalty term is replaced with $\mathbf{p}-|\mathbf{p}^k|\mathbf{n}$. Then the **p**-subproblem is reformulated as follows:

$$\begin{aligned}\mathbf{p}^{k+1} = \arg\min_{\mathbf{p}} &\int_\Omega (a+b(h^k)^2)\,|\mathbf{p}| + \int_\Omega \lambda_1^k\cdot(\mathbf{p}-|\mathbf{p}|\mathbf{n}^k)\\ &+\frac{r_1}{2}\int_\Omega |\mathbf{p}-|\mathbf{p}^k|\mathbf{n}^k|^2 + \int_\Omega \lambda_2^k\cdot\mathbf{p} + \frac{r_2}{2}\int_\Omega |\mathbf{p}-\nabla u^{k+1}|^2.\end{aligned} \tag{2.7}$$

Then there is a closed form solution for the **p**-subproblem (2.7). By utilizing Lemma 2.1, the solution of the **p**-subproblem can be given by

$$\mathbf{p}^{k+1} = S_{\frac{c}{r_1+r_2}}\left(\frac{r_1|\mathbf{p}^k|\mathbf{n}^k + r_2\nabla u^{k+1} - \lambda_1^k - \lambda_2^k}{r_1+r_2}\right), \tag{2.8}$$

where

$$c = a + b(h^k)^2 - \lambda_1^k\cdot\mathbf{n}^k. \tag{2.9}$$

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

The h-subproblem also has a closed form solution which can be obtained by solving the corresponding Euler-Lagrange equation. And it is simple to show that

$$h^{k+1} = \frac{r_3 \nabla \cdot \mathbf{n}^{k+1} - \lambda_3^k}{2b|\mathbf{p}^{k+1}| + r_3}. \tag{2.10}$$

For the **n**-subproblem, in order to avoid the singularity arising from the operator $\nabla(\nabla\cdot)$, a quadratic penalty term is added. Therefore, the **n**-subproblem is rewritten as follows:

$$\begin{aligned}\mathbf{n}^{k+1} = \arg\min_{\mathbf{n}} \int_\Omega -(\lambda_1^k \cdot |\mathbf{p}^{k+1}|\mathbf{n} + \lambda_3^k \nabla \cdot \mathbf{n}) + \frac{r_1}{2}\int_\Omega |\mathbf{p}^{k+1} - |\mathbf{p}^{k+1}|\mathbf{n}|^2 \\ + \frac{r_3}{2}\int_\Omega (h^k - \nabla \cdot \mathbf{n})^2 + \frac{\gamma}{2}\int_\Omega |\mathbf{n} - \mathbf{n}^k|^2,\end{aligned} \tag{2.11}$$

where $\gamma > 0$ is a penalty parameter. In [10], the u-subproblem and **n**-subproblem were solved by applying a Gauss-Seidel method to their related Euler-Lagrange equations.

Algorithm 2.1. (*ALM for Solving* (1.1))

Step 1. Input $a, b, \lambda, \gamma, r_1, r_2, r_3$. *Initialization:* $u^0 = f, \boldsymbol{p}^0, \boldsymbol{n}^0, h^0$, *and* $\lambda_1^0, \lambda_2^0, \lambda_3^0$. *Let* $k := 0$.

Step 2. Compute $(u^{k+1}, \boldsymbol{p}^{k+1}, \boldsymbol{n}^{k+1}, h^{k+1}; \lambda_1^{k+1}, \lambda_2^{k+1}, \lambda_3^{k+1})$ *by solving* (2.6).

Step 3. Stop if the stopping criterion is satisfied. Otherwise, let $k := k + 1$ *and go to Step 2.*

Note that the ALM for solving the Euler's elastica model (1.1) requires solving two systems of PDEs related to the u-subproblem and the **n**-subproblem at each iteration. As a result, the computational cost is expensive.

3. The linearized augmented Lagrangian method

In the augmented Lagrangian method (ALM) for the model (1.1), the way to solve the associated Euler-Lagrange equations of the u-subproblem and **n**-subproblem is usually time-consuming. Concretely speaking, the ALM still needs to solve them by using the Gauss-Seidel iterative method, which demands a high computational cost at each iteration. This problem is mainly caused by the fact that these two subproblems have no closed form solutions. Therefore, in this section, we introduce a linearized ALM where the u-subproblem and **n**-subproblem also have closed form solutions. Namely, we use matrix-vector multiplication to reduce computational cost instead of pursuing the solutions of the systems of equations by the Gauss-Seidel method.

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

3.1. Linearization for the u-subproblem and the n-subproblem

In [10], the time-consuming parts of the ALM are to solve the Euler-Lagrange equations of the u-subproblem and **n**-subproblem, which are given as

$$-r_2\Delta u + \lambda u = \lambda f - \nabla\cdot(r_2\mathbf{p}^k + \lambda_2^k) \tag{3.1}$$

and

$$\begin{aligned}&\left(\gamma + r_1|\mathbf{p}^{k+1}|^2 - r_3\nabla(\nabla\cdot)\right)\mathbf{n}\\ =&\gamma\mathbf{n}^k + r_1\mathbf{p}^{k+1}|\mathbf{p}^{k+1}| + \lambda_1^k|\mathbf{p}^{k+1}| - r_3\nabla h^k - \nabla\lambda_3^k,\end{aligned} \tag{3.2}$$

respectively.

There are no closed form solutions for the above equations and hence the computational costs of u^{k+1} and $\mathbf{n}^{k+1}$ are expensive. Fortunately, we find that these difficulties can be avoided by the linearization of the penalty terms $\frac{r_2}{2}\int_\Omega|\mathbf{p}^k - \nabla u|^2$ and $\frac{r_3}{2}\int_\Omega(h^k - \nabla\cdot\mathbf{n})^2$. Using the second-order Taylor expansion of these two terms at u^k and $\mathbf{n}^k$, respectively, we get

$$\begin{aligned}&\frac{r_2}{2}\int_\Omega|\mathbf{p}^k - \nabla u|^2\\ \approx&\frac{r_2}{2}\int_\Omega|\mathbf{p}^k - \nabla u^k|^2 + r_2\int_\Omega\langle\nabla\cdot(\mathbf{p}^k - \nabla u^k), u - u^k\rangle + \frac{1}{2\delta_1}\int_\Omega(u-u^k)^2\end{aligned} \tag{3.3}$$

$$\begin{aligned}&\frac{r_3}{2}\int_\Omega(h^k - \nabla\cdot\mathbf{n})^2\\ \approx&\frac{r_3}{2}\int_\Omega(h^k - \nabla\cdot\mathbf{n}^k)^2 + r_3\int_\Omega\langle\nabla(h^k - \nabla\cdot\mathbf{n}^k), \mathbf{n} - \mathbf{n}^k\rangle + \frac{1}{2\delta_2}\int_\Omega|\mathbf{n} - \mathbf{n}^k|^2,\end{aligned} \tag{3.4}$$

where $\delta_1, \delta_2 > 0$ are two constants. Substituting (3.3)-(3.4) into the u-subproblem of (2.6) and the **n**-subproblem (2.11) respectively, according to the optimality conditions, we deduce that the corresponding Euler-Lagrange equations are

$$\frac{u-u^k}{\delta_1} + \lambda u = \lambda f - \nabla\cdot(r_2\mathbf{p}^k + \lambda_2^k) + r_2\Delta u^k \tag{3.5}$$

and

$$\begin{aligned}&\frac{\mathbf{n}-\mathbf{n}^k}{\delta_2} + \left(\gamma + r_1|\mathbf{p}^{k+1}|^2\right)\mathbf{n}\\ =&\gamma\mathbf{n}^k + r_1\mathbf{p}^{k+1}|\mathbf{p}^{k+1}| + \lambda_1^k|\mathbf{p}^{k+1}| - r_3\nabla h^k - \nabla\lambda_3^k + r_3\nabla(\nabla\cdot\mathbf{n}^k).\end{aligned} \tag{3.6}$$

Then it is easy to see that the u-subproblem and the **n**-subproblem have closed form solutions, which are defined as

$$u^{k+1} = \frac{u^k + \delta_1 g_1}{1 + \delta_1\lambda}, \tag{3.7a}$$

$$\mathbf{n}^{k+1} = \frac{\mathbf{n}^k + \delta_2 g_2}{1 + \delta_2(\gamma + r_1|\mathbf{p}^{k+1}|^2)}, \tag{3.7b}$$

where g_1 and g_2 denote the right-hand sides of (3.5) and (3.6), respectively.

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

3.2. The proposed algorithm

By combining (2.8), (2.10), (3.7) and the last three equations of (2.6), we obtain a linearized ALM for solving the Euler's elastica model (1.1), which can be expressed as the following simple form:

$$u^{k+1} = \frac{u^k + \delta_1 g_1}{1 + \delta_1 \lambda}, \tag{3.8a}$$

$$\mathbf{p}^{k+1} = S_{\frac{c}{r_1+r_2}}\left(\frac{r_1|\mathbf{p}^k|\mathbf{n}^k + r_2\nabla u^{k+1} - \lambda_1^k - \lambda_2^k}{r_1 + r_2}\right), \tag{3.8b}$$

$$\mathbf{n}^{k+1} = \frac{\mathbf{n}^k + \delta_2 g_2}{1 + \delta_2(\gamma + r_1|\mathbf{p}^{k+1}|^2)}, \tag{3.8c}$$

$$h^{k+1} = \frac{r_3\nabla\cdot\mathbf{n}^{k+1} - \lambda_3^k}{2b|\mathbf{p}^{k+1}| + r_3}, \tag{3.8d}$$

$$\lambda_1^{k+1} = \lambda_1^k + r_1(\mathbf{p}^{k+1} - |\mathbf{p}^{k+1}|\mathbf{n}^{k+1}), \tag{3.8e}$$

$$\lambda_2^{k+1} = \lambda_2^k + r_2(\mathbf{p}^{k+1} - \nabla u^{k+1}), \tag{3.8f}$$

$$\lambda_3^{k+1} = \lambda_3^k + r_3(h^{k+1} - \nabla\cdot\mathbf{n}^{k+1}). \tag{3.8g}$$

Algorithm 3.1. (*Linearized ALM for Solving* (1.1))

Step 1. Input $a, b, \lambda, \gamma, r_1, r_2, r_3, \delta_1, \delta_2$. *Initialization:* $u^0 = f, \boldsymbol{p}^0, \boldsymbol{n}^0, h^0$, *and* $\lambda_1^0, \lambda_2^0, \lambda_3^0$. *Let* $k := 0$.

Step 2. Compute $(u^{k+1}, \boldsymbol{p}^{k+1}, \boldsymbol{n}^{k+1}, h^{k+1}; \lambda_1^{k+1}, \lambda_2^{k+1}, \lambda_3^{k+1})$ *by solving* (3.8).

Step 3. Stop if the stopping criterion is satisfied. Otherwise, let $k := k + 1$ *and go to Step 2.*

Compared with Algorithm 2.1, Algorithm 3.1 requires less computational cost because it only involves the shrink operator and all subproblems have closed form solutions.

3.3. Complexity analysis

In the following, we will give some simple analysis of complexity for the proposed method.

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

For our proposed method, the cost of updating u^k is $16N+2$ at each iteration, where $N = mn$ is the total number of grid points. At the same time, if a root operation is equal to 12 times addition, or subtraction, or multiplication, or division, it will spend $58N+1$ ($2N$ times root operations), $48N$ (N times root operations), $7N+1$, $8N$, $6N$ and $3N$ operations for each iteration on updating $\mathbf{p}^k, \mathbf{n}^k, h^k, \lambda_1^k, \lambda_2^k, \lambda_3^k$, respectively. Then the total cost of one step is $146N+4$, which gives rise to the $O(N)$ complexity.

4. Numerical experiments

In this section, some numerical experiments are presented to illustrate the effectiveness and efficiency of the proposed method for Euler's elastica model. For simplicity of presentation, we denote the newly developed method as LALM. Specifically, we compare our results with those of the augmented Lagrangian method proposed in [10] which is denoted as GSALM. Moreover, to show the significance of the linearization for the n-subproblem, we also compare with the LnALM in which the linearization technique will only be applied to the n-subproblem but not the u-subproblem. Experimental results indicate that the proposed method converges faster, especially for large-sized images.

The programs were coded in MATLAB and run on a personal computer with a 3.0GHZ CPU processor. For each experiment, we use the relative error of the solution u^k defined by

$$\frac{\| u^k - u^{k-1} \|_2}{\| u^{k-1} \|_2} \tag{4.1}$$

as the stopping criterion. That is, once the relative error is less than a given error tolerance, the iteration process will be terminated and the solution is obtained. In order to check whether the iteration converges, we monitor the relative error (4.1). We also observe how the numerical energy of the Euler's elastica model evolves during the iteration process. To this end, we track the amount E^k defined as follows:

$$E^k = \int_\Omega \left(a + b \left(h^k \right)^2 \right) \mid \mathbf{p}^k \mid + \frac{\lambda}{2} \int_\Omega (u^k - f)^2. \tag{4.2}$$

To assess the restoration performance qualitatively, the peak signal to noise ratio (PSNR) defined as

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} (u_{i,j} - I_{i,j})^2}, \tag{4.3}$$

is used, where $u_{i,j}$ and $I_{i,j}$ denote the pixel values of the restored image and the original clean image, respectively.

In the following, three synthetic and a real images used in [10] are chosen as the test images. The noisy images are generated by adding random noise with mean zero and the standard deviation 10 to the clean images. The same parameters as did in [10] are used in experiments. In this way, it will be more reasonable when we compare with the GSALM proposed in [10].

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

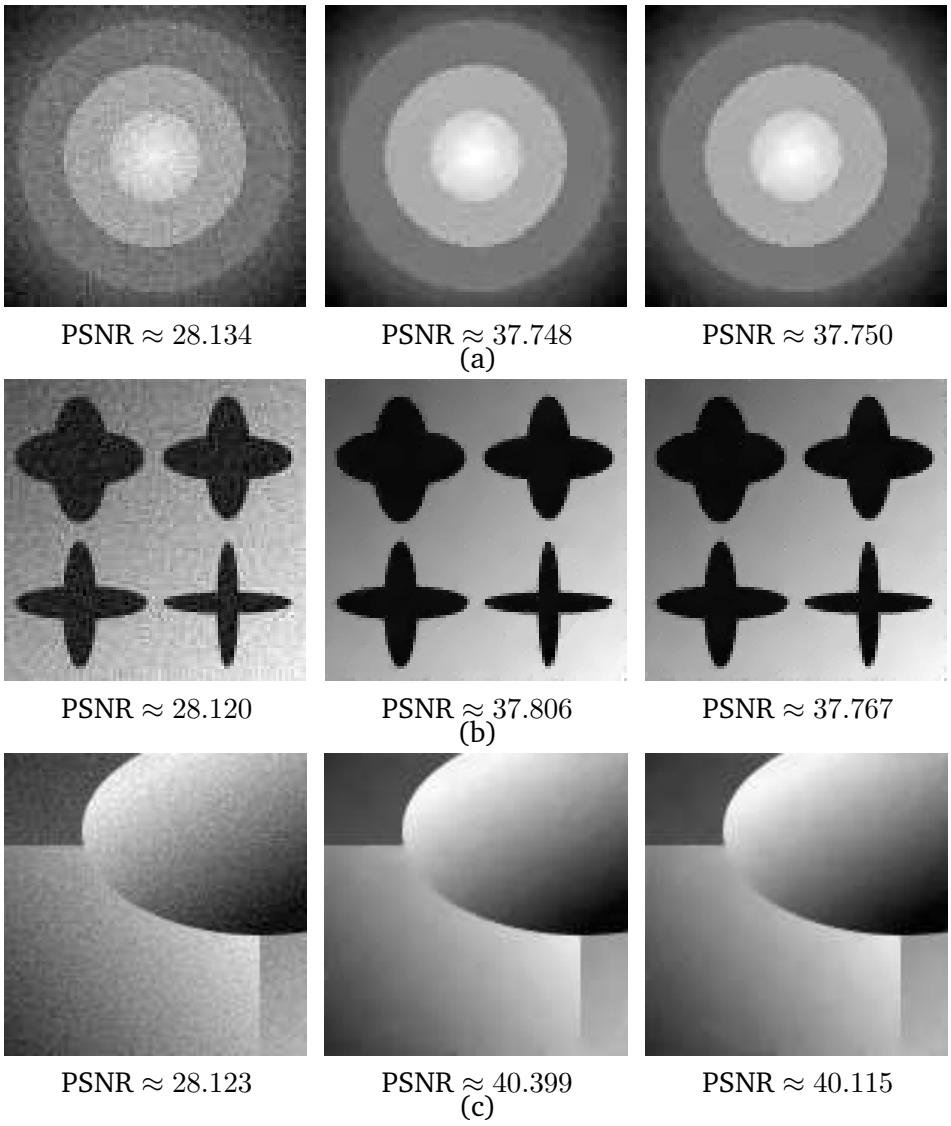


Figure 1: The denoising results for three synthetic images. Here $a = 1, b = 10, \lambda = 1, r_1 = 0.03, r_2 = 10, r_3 = 10000, \gamma = 0.01$. Specifically, in the LALM, $\delta_1 = 2.3 \cdot 10^{-2}, \delta_2 = 4.5 \cdot 10^{-5}$ are chosen and the tolerance is $5.5 \cdot 10^{-4}$ for (a). For (b), $\delta_1 = 2.51 \cdot 10^{-2}, \delta_2 = 4.7 \cdot 10^{-5}$ and the tolerance $5.9 \cdot 10^{-4}$ are used. And for (c), set $\delta_1 = 2.1 \cdot 10^{-2}, \delta_2 = 4.4 \cdot 10^{-5}$ and the tolerance is $4 \cdot 10^{-4}$. The noisy image, the denoised images by the GSALM and the LALM are shown from the first column to the third column, respectively.

The results of the synthetic images are shown in Fig. 1 and the result of the real image is displayed in Fig. 2. For all four experiments, the related numerical results are listed in Table 1, where iter., cpu(s) and Δ denote the number of iterations, the CPU time in second required for the whole denoising process and the saved relative time indicated as a percentage, respectively.

From the first and third columns of Fig. 1 and those of Fig. 2, we see that the visual

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

Table 1: Numerical results.

Image	Algorithm	iter.	cpu(s)	PSNR	Δ
Fig.1(a)	GSALM	310	3.728	37.748	
100×100	LnALM	397	1.357	37.800	63.6%
	LALM	363	1.201	37.750	67.8%
Fig.1(b)	GSALM	135	1.716	37.806	
100×100	LnALM	132	0.483	37.723	71.9%
	LALM	138	0.436	37.767	74.6%
Fig.1(c)	GSALM	96	1.981	40.399	
128×128	LnALM	100	0.717	40.711	63.8%
	LALM	99	0.639	40.115	67.7%
Fig.2(a)	GSALM	88	8.486	32.118	
256×256	LnALM	81	3.229	32.161	61.9%
	LALM	83	2.948	32.150	65.3%



PSNR ≈ 28.131 PSNR ≈ 32.118 PSNR ≈ 32.150

Figure 2: The denoising result for a real image. We set $a = 1, b = 2, \lambda = 1, r_1 = 0.02, r_2 = 10, r_3 = 5000, \gamma = 0.01$. Specifically, we choose $\delta_1 = 2.25 \cdot 10^{-2}, \delta_2 = 10^{-4}$ for the LALM. The tolerance is $7.75 \cdot 10^{-4}$. The noisy image, the denoised images by the GSALM and the LALM are shown from the first column to the third column, respectively.

results are impressive. In other words, our proposed algorithm (LALM) is effective. Specifically, we can observe from the second and third columns of Fig. 1 and those of Fig. 2 that the visual results of our LALM are comparable to the GSALM. Furthermore, from Table 1, we find that the LALM performs better than the GSALM in terms of CPU time, while delivering almost the same quality (PSNR) of restoration. Notice from this table that for the above four images, both LnALM and LALM are faster than GSALM and the LALM is a little faster than the LnALM, while the PSNR values obtained by these three methods are almost equal. This means that the linearization for the n-subproblem saves more computational cost than that for the u-subproblem.

The results also show that the CPU time grows fast when the size of the image increases. In order to further show the advantage of the LALM, we adopt four larger images as examples. The Gaussian white noise with mean zero and standard deviation 20 is added to the test images. Specifically, to test the convergence of the LALM, we

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

Table 2: Numerical results for the LALM with $\delta_1 = 0.4$.

δ_2	$8 \cdot 10^{-5}$	10^{-4}	$3 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$7 \cdot 10^{-4}$	$9 \cdot 10^{-4}$	$11 \cdot 10^{-4}$
PSNR	28.414	28.503	28.889	28.753	28.615	28.509	28.441

Table 3: Numerical results for the LALM with $\delta_2 = 3 \cdot 10^{-4}$.

δ_1	0.2	0.25	0.3	0.35	0.4	0.45	0.5
PSNR	28.647	28.756	28.829	28.858	28.868	28.857	28.749

Table 4: Numerical results.

Image	Algorithm	iter.	cpu(s)	PSNR	Δ
Fig.3 cameraman	GSALM	312	29.437	28.855	
512×512	LALM	328	11.871	28.956	59.7%
Fig.4 lena	GSALM	146	63.461	30.537	
512×512	LALM	167	25.412	31.201	60.0%
Fig.5 man	GSALM	501	220.070	29.472	
512×512	LALM	220	33.821	29.525	84.6%
Fig.6 boat	GSALM	460	827.569	32.415	
1024×1024	LALM	212	132.366	32.315	84.0%

track the relative error in u (4.1), the numerical energy (4.2), the PSNR (4.3) of the images. And the plots are displayed in Figs. 3 and 4. While we show numerical results obtained at the iteration when the stopping criterion is met, all graphs related to the LALM and GSALM are drawn up to 1000 iterations.

In the Euler's elastica model (1.1), there are three parameters: a, b and λ. By using the previous augmented Lagrangian method (GSALM), there are four more parameters r_1, r_2, r_3, γ. In our improved augmented Lagrangian method (LALM), we have two more parameters δ_1, δ_2. We will study the behavior and effect of parameters δ_1, δ_2. In our fifth experiment for 'cameraman' image, we choose $a = 1, b = 0.1, \lambda = 0.12, r_1 = 0.08, r_2 = 0.4, r_3 = 1000, \gamma = 0.01$ and our proposed LALM is terminated after 1000 steps. The PSNR values for various values of δ_2 are listed in Table 2. From this table, we observe that the PSNR value of the denoised image with $\delta_2 = 3 \cdot 10^{-4}$ is the biggest. Then for the case of $\delta_2 = 3 \cdot 10^{-4}$, we further study the effect of parameter δ_1 on the LALM. We list the corresponding results for various values of δ_1 in Table 3. From this table, we know that we would better choose $\delta_1 = 0.4$. It is necessary to emphasize that in order to choose better values of parameters, we did the same work while using the GSALM. Once the better values of parameters in the GSALM are set, we choose the same values of parameters in the LALM in most cases. Then we study how to choose the parameters δ_1, δ_2.

It is easy to see from Figs. 3-6 that the visual results of the LALM are also comparable to the GSALM. And numerical results in Table 4 demonstrate that the LALM spends less CPU time than the GSALM, while the PSNR values obtained by these two methods

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

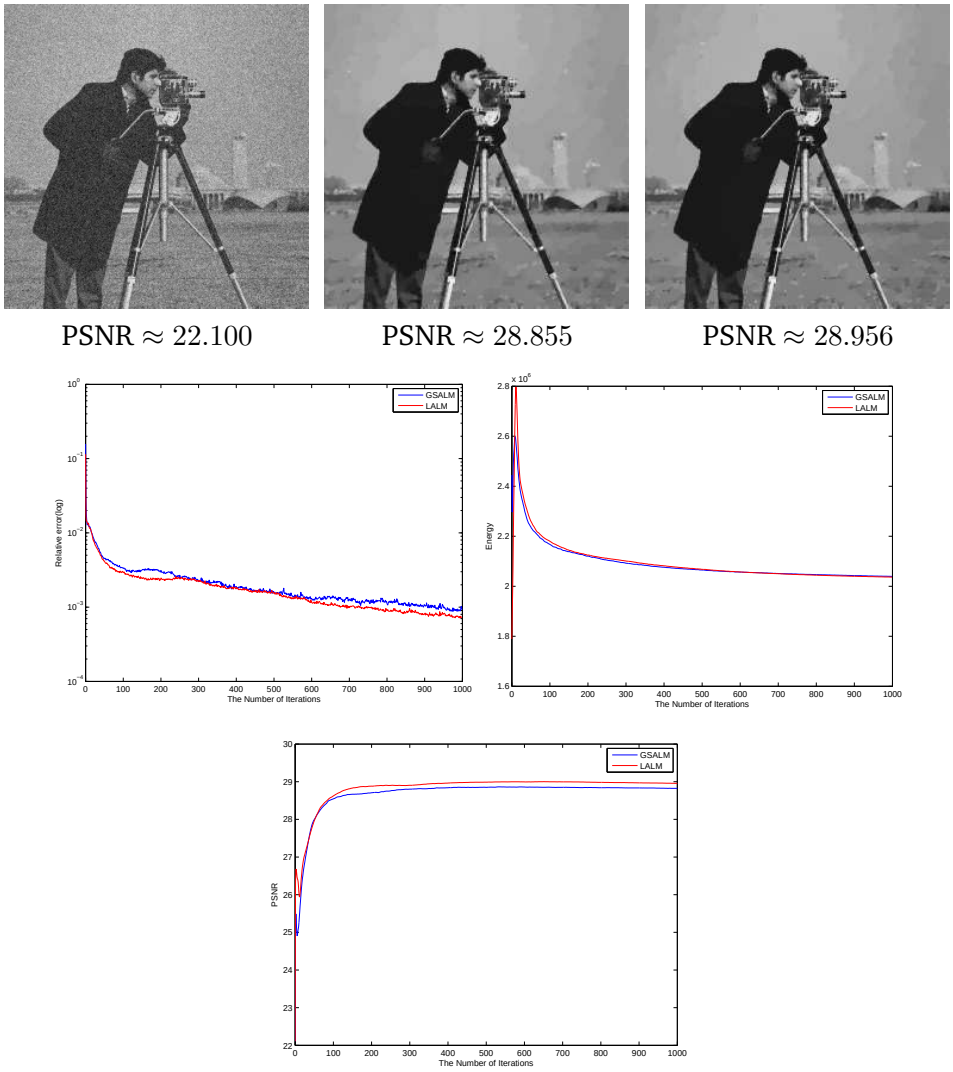


Figure 3: The denoising results and the plots of (4.1), (4.2) and (4.3) for the real cameraman image. We set $a = 1, b = 0.1, \lambda = 0.12, r_1 = 0.08, r_2 = 0.4, r_3 = 1000, \gamma = 0.01$. Specifically, in the LALM, $\delta_1 = 0.4, \delta_2 = 3 \cdot 10^{-4}$ are chosen and the tolerance is $2 \cdot 10^{-3}$. The noisy image, the denoised images by the GSALM and the LALM are shown on the first row, respectively. The plots of relative error in u^k and numerical energy are displayed on the second row while the plot of PSNR is shown on the third row.

are almost equal. Since the Euler's elastica energy functional is not convex, we can not expect monotonic decrease in the values of the numerical energy and relative error in u. From these plots in Figs. 3-6, it is shown that the LALM is stable and has converged, just as the GSALM did. It is actually clear that the energy and PSNR have converged to a steady state.

The experimental results in Tables 1 and 4 show that the LALM is almost 2 times faster than the GSALM in terms of the CPU time. It can restore noisy images quite well in an efficient manner, especially for some large-sized images.

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press



PSNR ≈ 22.113 PSNR ≈ 30.537 PSNR ≈ 31.201

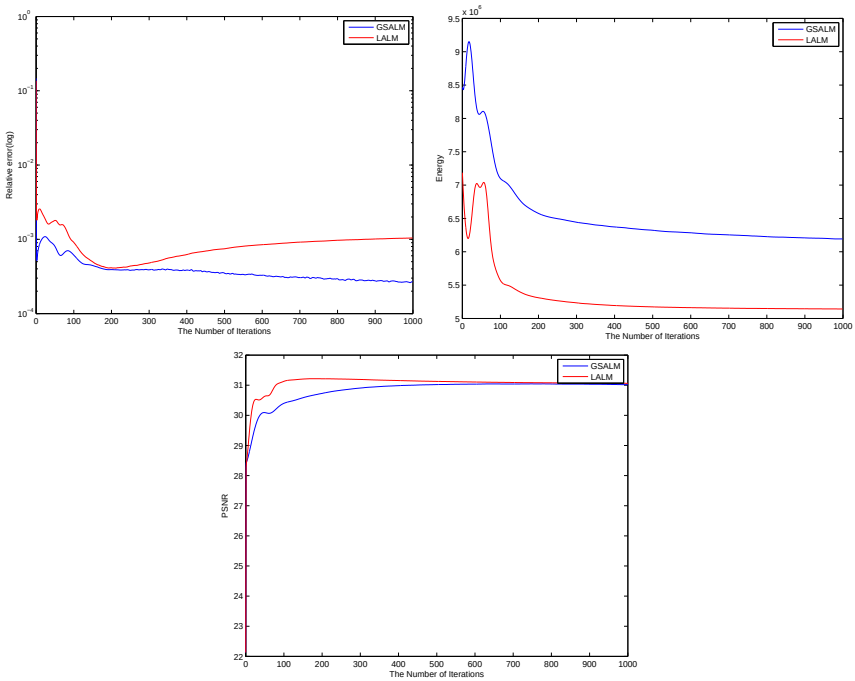


Figure 4: The denoising results and the plots of (4.1), (4.2) and (4.3) for the real lena image. For the GSALM, we choose $a = 1, b = 1.8, \lambda = 0.1, r_1 = 0.03, r_2 = 15, r_3 = 600, \gamma = 0.01$. The same parameters are used except $\lambda = 0.08, r_2 = 4$ for the LALM. Specifically, in the LALM, $\delta_1 = 0.05, \delta_2 = 6 \cdot 10^{-4}$ are chosen. The tolerance is $4.5 \cdot 10^{-4}$. The noisy image, the denoised images by the GSALM and the LALM are shown on the first row, respectively. The plots of relative error in u^k and numerical energy are displayed on the second row while the plot of PSNR is shown on the third row.

5. Conclusion

In this paper, we presented a linearized augmented Lagrangian method for solving the Euler's elastica image denoising model. We applied the linearlization technique to two subproblems such that all subproblems in the newly developed method had closed form solutions and thus could be solved rather easily. Furthermore, we analyzed the complexity of the proposed method. We tested this method by comparing it with

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

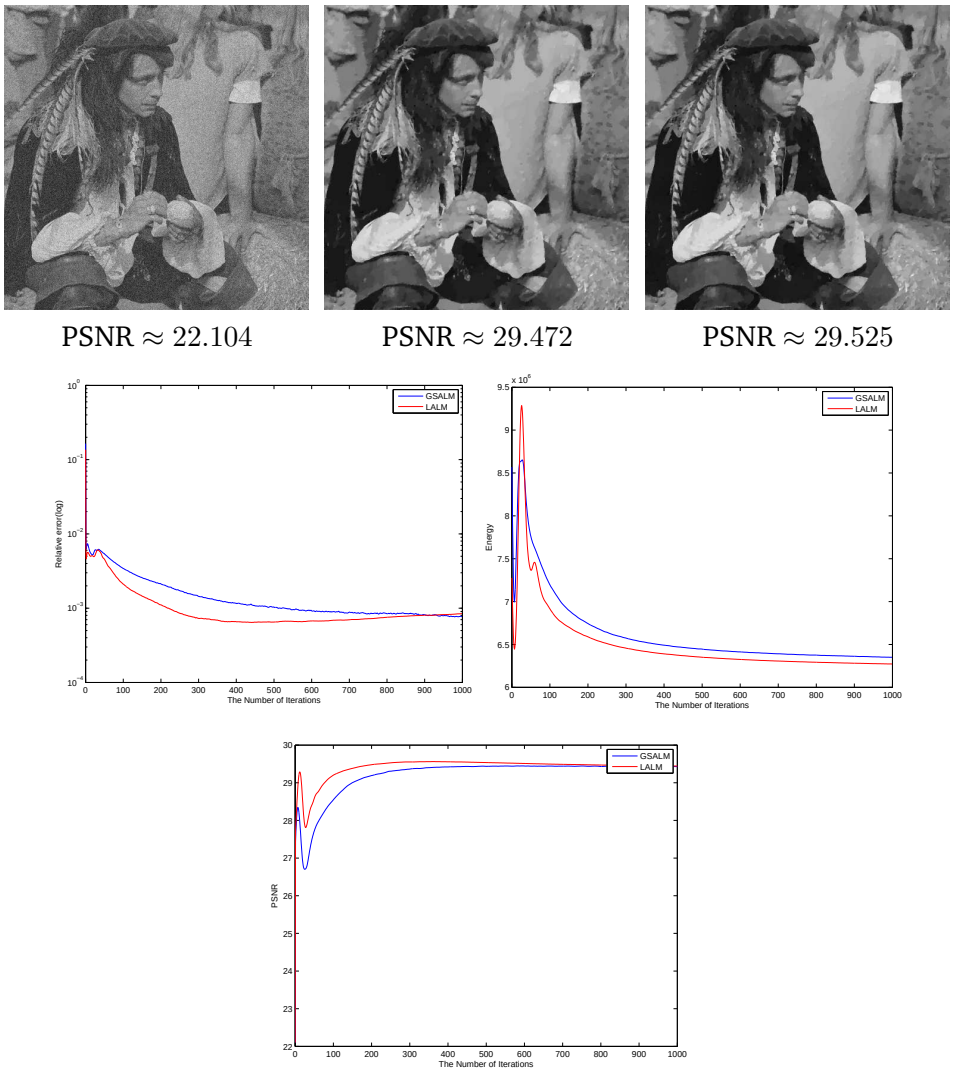


Figure 5: The denoising results and the plots of (4.1), (4.2) and (4.3) for the real man image. For the GSALM, we set $a = 1, b = 0.12, \lambda = 0.1, r_1 = 0.06, r_2 = 1.8, r_3 = 1600, \gamma = 0.01$. The same parameters are used for the LALM. Specifically, in the LALM, $\delta_1 = 0.1, \delta_2 = 3 \cdot 10^{-4}$ are chosen. The tolerance is 10^{-3}. The noisy image, the denoised images by the GSALM and the LALM are shown on the first row, respectively. The plots of relative error in u^k and numerical energy are displayed on the second row while the plot of PSNR is shown on the third row.

the previous augmented Lagrangian method in order to demonstrate its efficiency. As numerical results demonstrated, our method performed better in terms of CPU time. Especially, it is shown that our method is more suitable to deal with large-sized images. The plots of relative error in u^k, numerical energy and PSNR illustrate that it can converge to a steady state.

Acknowledgments We would like to thank Dr. Yuping Duan of Nankai University for sharing her codes. The authors also thank her, Dr. Zhi-Feng Pang of Henan University

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

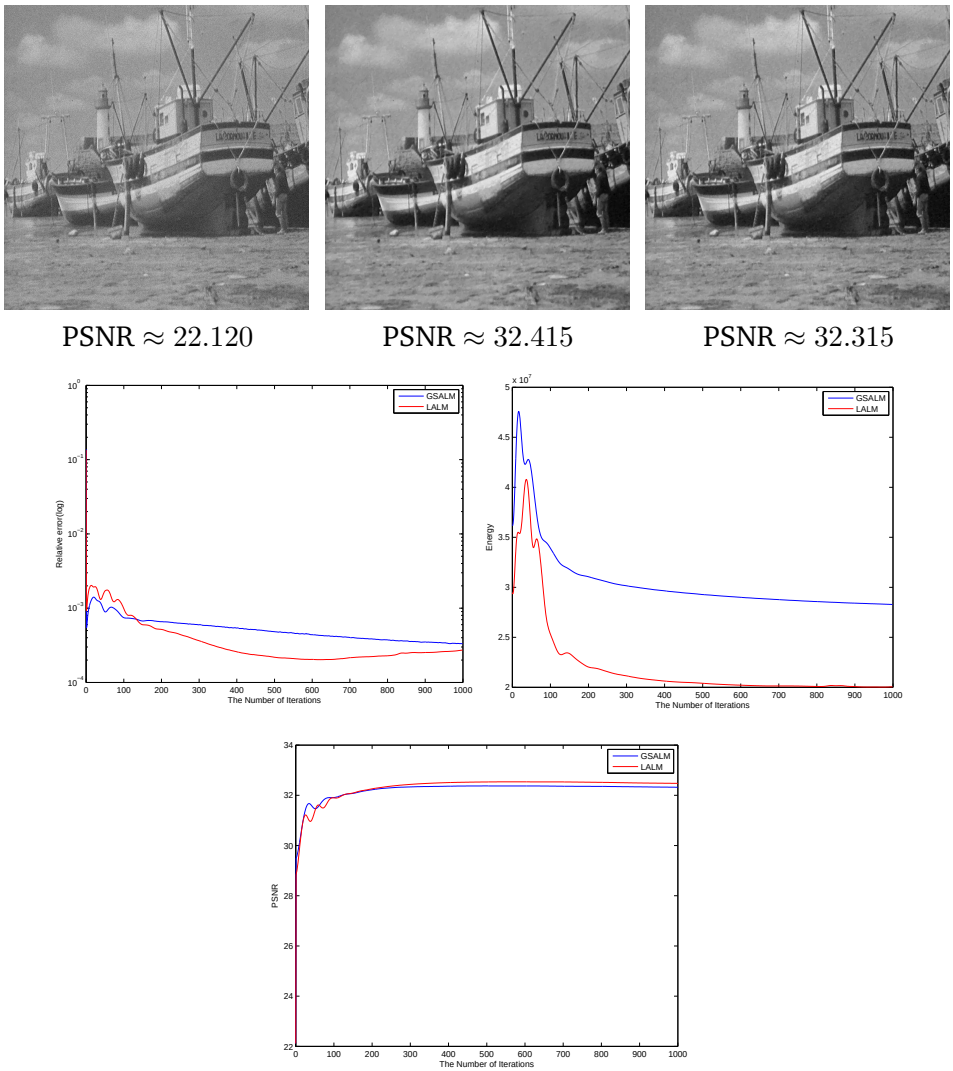


Figure 6: The denoising results and the plots of (4.1), (4.2) and (4.3) for the real boat image. For the GSALM, we set $a = 1, b = 5.5, \lambda = 0.12, r_1 = 0.085, r_2 = 25, r_3 = 1100, \gamma = 0.01$. The same parameters are used except $\lambda = 0.08, r_2 = 16$ for the LALM. Specifically, in the LALM, $\delta_1 = 0.013, \delta_2 = 3.15 \cdot 10^{-4}$ are chosen. The tolerance is $5 \cdot 10^{-4}$. The noisy image, the denoised images by the GSALM and the LALM are shown on the first row, respectively. The plots of relative error in u^k and numerical energy are displayed on the second row while the plot of PSNR is shown on the third row.

and Dr. Huibin Chang of Tianjin Normal University for their valuable comments and suggestions.

The research has been supported by the NNSF of China grants 11526110, 11271069, 61362036 and 61461032, the 863 Program of China grant 2015AA01A302, the Open Research Fund of Jiangxi Province Key Laboratory of Water Information Cooperative Sensing and Intelligent Processing (2016WICSIP013), and the Youth Foundation of Nanchang Institute of Technology (2014KJ021).

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

References

[1] K. Bredies, K. Kunisch, and T. Pock, *Total generalized variation*, SIAM J. Imaging Sciences, vol. 3 (2010), pp. 492-526.

[2] J. L. Carter, *Dual methods for total variation-based image restoration*, Technical report, UCLA CAM Report 02-13, 2002.

[3] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vis., vol. 20 (2004), pp. 89-97.

[4] A. Chambolle, *Total variation minimization and a class of binary MRF models*, EMMCVPR 2005, LNCS 3757 (2005), pp. 136-152.

[5] T. F. Chan, G. H. Golub, and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., vol. 20 (1999), pp. 1964-1977.

[6] T. F. Chan and J. Shen, *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*, SIAM Publisher, Philadelphia, 2005.

[7] D. Q. Chen and Y. Zhou, *Multiplicative denoising based on linearized alternating direction method using discrepancy function constraint*, J. Sci. Comput., vol. 60 (2014), pp. 483-504.

[8] Y. Chen, S. Levine, and M. Rao, *Variable exponent, linear growth functionals in image restoration*, SIAM J. Appl. Math., vol. 66 (2006), pp. 1383-1406.

[9] Y. Duan and W. Huang, *A fixed-point augmented lagrangian method for total variation minimization problems*, J. Vis. Commun. Image R., vol. 24 (2013), pp. 1168-1181.

[10] Y. Duan, Y. Wang, and J. Hahn, *A fast augmented Lagrangian method for Euler's elastica models*, Numer. Math. Theor. Meth. Appl., vol. 6 (2013), pp. 47-71.

[11] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sciences, vol. 2 (2009), pp. 323-343.

[12] J. Hahn, X. C. Tai, S. Borok, and A. M. Bruckstein, *Orientation-matching minimization for image denoising and inpainting*, Int. J. Comput. Vision, vol. 92 (2011), pp. 308-324.

[13] J. Hahn, C. Wu, and X. C. Tai, *Augmented Lagrangian method for generalized TV-Stokes model*, J. Sci. Comput., vol. 50 (2012), pp. 235-264.

[14] M. Hintermüller, C. N. Rautenberg, and J. Hahn, *Functional-analytic and numerical issues in splitting methods for total variation-based image reconstruction*, Inverse Probl., vol. 30 (2014), pp. 055014.

[15] T. Jeong, H. Woo, and S. Yun, *Frame-based Poisson image restoration using a proximal linearized alternating direction method*, Inverse Probl., vol. 29 (2013), pp. 075007.

[16] R. Q. Jia and H. Zhao, *A fast algorithm for the total variation model of image denoising*, Adv. Comput. Math., vol. 33 (2010), pp. 231-241.

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

[18] S. Masnou and J. M. Morel, *Level lines based disocclusion*, IEEE Int. Conf. Image Process., vol. 1998 (1998), pp. 259-263.

[19] M. Myllykoski, R. Glowinski, T. Kärkkäinen, and T. Rossi, *A new augmented Lagrangian approach for L^1-mean curvature image denoising*, SIAM J. Imaging Sciences, vol. 8 (2015), pp. 95-125.

[20] N. Paragios, Y. Chen, and O. Faugeras, *Handbook of Mathematical Models in Computer Vision*, Springer, Heidelberg, 2005.

[21] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D, vol. 60 (1992), pp. 259-268.

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press

[22] J. SHEN, S. H. KANG, AND T. F. CHAN, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., vol. 63 (2003), pp. 564-592.
[23] X. C. TAI, J. HAHN, AND G. J. CHUNG, *A fast algorithm for Euler's elastica model using augmented Lagrangian method*, SIAM J. Imaging Sciences, vol. 4 (2011), pp. 313-344.
[24] X. C. TAI AND C. WU, *Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model*, SSVM 2009, LNCS 5567 (2009), pp. 502-513.
[25] C. R. VOGEL AND M. E. OMAN, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., vol. 17 (1996), pp. 227-238.
[26] 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. Imaging Sciences, vol. 3 (2010), pp. 300-339.
[27] J. YANG, Y. ZHANG, AND W. YIN, *An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise*, SIAM J. Sci. Comput., vol. 31 (2009), pp. 2842-2865.
[28] Y. L. YOU AND M. KAVEH, *Fourth-order partial differential equations for noise removal*, IEEE Trans. Image Process., vol. 9 (2000), pp. 1723-1730.
[29] M. ZHU, *Fast Numerical Algorithms for Total Variation Based Image Restoration*, Ph.D. Thesis, UCLA, 2008.
[30] M. ZHU AND T. F. CHAN, *An efficient primal-dual hybrid gradient algorithm for total variation image restoration*, Technical report, UCLA CAM Report 08-34, 2008.
[31] M. ZHU, S. J. WRIGHT, AND T. F. CHAN, *Duality-based algorithms for total-variation-regularized image restoration*, Comput. Optim. Appl., vol. 47 (2010), pp. 377-400.
[32] W. ZHU AND T. CHAN, *Image denoising using mean curvature of image surface*, SIAM J. Imaging Sciences, vol. 5 (2012), pp. 1-32.
[33] W. ZHU, X. C. TAI, AND T. CHAN, *Image segmentation using Eulers elastica as the regularization*, J. Sci. Comput., vol. 57 (2013), pp. 414-438.
[34] W. ZHU, X. C. TAI, AND T. CHAN, *A fast algorithm for a mean curvature based image denoising model using augmented lagrangian method*, Global Optimization Methods, LNCS 8293 (2014), pp. 104-118.

https://doi.org/10.4208/nmtma.2017.m1611 Published online by Cambridge University Press