Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.2013.mssvm03

Vol. **6**, No. 1, pp. 47-71
February 2013

A Fast Augmented Lagrangian Method for Euler's Elastica Models

Yuping Duan[1,*], Yu Wang[2] and Jooyoung Hahn[3]

[1] *Institute for Infocomm Research, Singapore.*
[2] *Computer Science Department, Technion, Haifa 32000, Israel.*
[3] *Institute for Mathematics and Scientific Computing, University of Graz, Austria.*

Received 6 December 2011; Accepted (in revised version) 11 September 2012

Available online 11 January 2013

Abstract. In this paper, a fast algorithm for Euler's elastica functional is proposed, in which the Euler's elastica functional is reformulated as a constrained minimization problem. Combining the augmented Lagrangian method and operator splitting techniques, the resulting saddle-point problem is solved by a serial of subproblems. To tackle the nonlinear constraints arising in the model, a novel fixed-point-based approach is proposed so that all the subproblems either is a linear problem or has a closed-form solution. We show the good performance of our approach in terms of speed and reliability using numerous numerical examples on synthetic, real-world and medical images for image denoising, image inpainting and image zooming problems.

AMS subject classifications: 68U10, 65N21, 74S20

Key words: Euler's elastica, augmented Lagrangian method, image denoising, image inpainting, image zooming.

1. Introduction

Suppose that the observed image u_0 is the original image u perturbed by an additive noise η

$$u_0 = u + \eta.$$

The image denoising problems of recovering the image u from the noisy image u_0 are often solved by variational methods and optimization techniques. Among various variational denoising methods, the Rudin-Osher-Fatemi (ROF) method [31] is probably the most successful one, which is defined by minimizing the following functional

$$\min_u \int_\Omega |\nabla u| + \frac{\mu}{2}\int_\Omega (u-u_0)^2, \tag{1.1}$$

*Corresponding author. *Email addresses:* duany@i2r.a-star.edu.sg (Y. Duan), yuwang@cs.technion.ac.il (Y. Wang), JooyoungHahn@gmail.com (J. Hahn)

where μ is a positive parameter and u is defined on a continuous domain $\Omega \subset \mathbb{R}^2$.

The success of the ROF model mainly relies on the total variation (TV) regularization, which enables the ROF model to preserve sharp edges while removing noise. Due to its nice properties, TV-based models have been further extended to vectorial models for color image denoising [6, 13] and several fast algorithms are proposed [9–11]. In spite of many advantageous properties, TV-based methods have a common disadvantage: piecewise constant images are favored over piecewise smooth images, which is the so-called staircasing effect. To overcome this drawback, high order models [7, 8, 12, 14, 15, 24, 26, 32] are proposed to yield smoother results. As one of them, Euler's elastica model, which is defined based on the curvature of the level curves of images, was first introduced into computer vision by Mumford [28] and successfully applied to a number of applications, such as image restoration [1–3, 17], image segmentation [16, 27, 29] and image inpainting [4, 5, 14].

Euler's elastica energy can be described by the curvature κ of a smooth curve Γ as the following

$$E(\Gamma) = \int_{\Gamma} \big(a + b|\kappa|^{\beta}(s)\big)\, ds, \tag{1.2}$$

where s is the arc length and a, b are two positive parameters. In the functional (1.2), the first term minimizes the total length and the second term minimizes the power of total curvature. The power β can be set to either $\beta = 1$ as in [26], or $\beta = 2$ as in [14]. In this work, we set $\beta = 2$, but the techniques developed below can be extended to the case $\beta = 1$ without many efforts. The Euler's elastica of all the level curves of an image u can be expressed as

$$E = \int_{l=0}^{L} \int_{\gamma_l : u=l} \big(a + b|\kappa|^{\beta}(s)\big)\, ds\, dl, \tag{1.3}$$

where γ_l is the level curve with $u = l$. Note that the curvature κ can be expressed as a function of u

$$\kappa(u) = \nabla \cdot \Big(\frac{\nabla u}{|\nabla u|}\Big). \tag{1.4}$$

Substituting above equation into (1.3) and using the co-area formula yields

$$E(u) = \int_{\Omega} \Big(a + b\Big|\nabla \cdot \frac{\nabla u}{|\nabla u|}\Big|^{\beta}\Big)|\nabla u|. \tag{1.5}$$

For image denoising applications, the elastica energy (1.5) can be used as a regularization term. Together with the data fitting term, we can formulate the minimization problem to approximate the noisy image u_0 by Euler's elastica energy as follows

$$\min_{u} \int_{\Omega} \Big(a + b\Big(\nabla \cdot \frac{\nabla u}{|\nabla u|}\Big)^2\Big)|\nabla u| + \frac{\mu}{t}\int_{\Omega} |u - u_0|^t, \tag{1.6}$$

the choice of t is determined by the type of noise in u_0: e.g., $t = 1$ for salt & pepper noise and $t = 2$ for Gaussian white noise.

The numerical solutions [14, 26] of Euler's elastica functional (1.6) are normally complex and time consuming due to the high nonlinearity of the partial differential equations (PDEs). In [3, 21], graph-cuts methods are applied to the high order models and Euler's elastica. Recently, the operator splitting method is greatly studied to simplify the optimization problems in image processing and the augmented Lagrangian method (ALM) [30] has been successfully implemented to solve the generated constrained models; see [20, 34, 36–38]. Along the similar idea, [33] applies the ALM to Euler's elastica model (1.6), in which great efficiency is achieved by solving the subproblems emerging from the augmented Lagrangian functional with the Fast Fourier Transform (FFT). In their approach, a quadratic penalty term in the augmented Lagrangian functional is relaxed to the first order and a frozen coefficient method with the FFT is used to solve the coupled PDEs with the variable coefficient. The dependence of the FFT gives rise to a limitation to apply the algorithm in [33] to more general problems such as the problems defined on non-flat geometries [22]. To overcome this hurdle, [19] proposes an algorithm for the elastica energy by replacing the FFT with the cheap arithmetic operation.

In this work, we use the operator splitting technique to propose a novel augmented Lagrangian algorithm for Euler's elastica model (1.6). We decompose the saddle-point problem into subproblems to simplify the structure. Instead of disposing of the quadratic term in the subproblem related to the normal vector, we apply a fixed-point method to find a closed-form solution, which is different from [19]. Furthermore, in contrast to [33], we introduce a new variable into the constrained problem to avoid solving the PDEs with variable coefficients. In the proposed formulation, all subproblems are either linear which can be solved efficiently by the iterative solver, or having closed-form solutions. Therefore, the new method is computationally efficient in terms of both memory requests and computational costs.

2. Augmented Lagrangian method for Euler's elastica model

In this section, we propose an augmented Lagrangian formulation for Euler's elastica energy (1.6). First, we introduce two extra variables $\boldsymbol{p}$ and $\boldsymbol{n}$ to (1.6) and cast it into the following constrained minimization problem

$$\begin{aligned} &\min_{u,\boldsymbol{p},\boldsymbol{n}} \int_\Omega \Big(a + b(\nabla\cdot\boldsymbol{n})^2\Big)|\boldsymbol{p}| + \frac{\mu}{t}\int_\Omega |u-u_0|^t, \\ &\text{s.t.} \quad \boldsymbol{p} = \nabla u; \quad \boldsymbol{n} = \frac{\boldsymbol{p}}{|\boldsymbol{p}|}. \end{aligned} \tag{2.1}$$

The second constraint in (2.1) is equivalent to $\boldsymbol{p} = |\boldsymbol{p}|\boldsymbol{n}$ if $\boldsymbol{p} \neq \boldsymbol{0}$. Therefore, the corresponding augmented Lagrangian functional for the constrained optimization problem (2.1) is defined as follows:

$$\begin{aligned} L(u,\boldsymbol{p},\boldsymbol{n};\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2) = &\int_\Omega \big(a + b(\nabla\cdot\boldsymbol{n})^2\big)|\boldsymbol{p}| + \frac{\mu}{t}\int_\Omega |u-u_0|^t + \int_\Omega (\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})\cdot\boldsymbol{\lambda}_1 \\ &+ \frac{r_1}{2}\int_\Omega (\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})^2 + \int_\Omega (\boldsymbol{p}-\nabla u)\cdot\boldsymbol{\lambda}_2 + \frac{r_2}{2}\int_\Omega (\boldsymbol{p}-\nabla u)^2, \end{aligned} \tag{2.2}$$

where $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ are the Lagrange multipliers and r_1, r_2 are positive penalty parameters. We aim to seek a saddle-point of the augmented Lagrangian functional (2.2), which satisfies

$$\begin{aligned}&\text{Find } (u^*,\boldsymbol{p}^*,\boldsymbol{n}^*;\boldsymbol{\lambda}_1^*,\boldsymbol{\lambda}_2^*),\\&\text{s.t. } L(u^*,\boldsymbol{p}^*,\boldsymbol{n}^*;\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)\le L(u^*,\boldsymbol{p}^*,\boldsymbol{n}^*;\boldsymbol{\lambda}_1^*,\boldsymbol{\lambda}_2^*)\le L(u,\boldsymbol{p},\boldsymbol{n};\boldsymbol{\lambda}_1^*,\boldsymbol{\lambda}_2^*),\\&\qquad\qquad\qquad\qquad\qquad\qquad\forall(u,\boldsymbol{p},\boldsymbol{n};\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2).\end{aligned}\tag{2.3}$$

The rest of the paper is organized as follows. In Section 3, we first review the existing algorithm in [33] for solving the minimization problem (2.1). In Section 4, we modify the augmented Lagrangian functional (2.2) by introducing another new variable. We also illustrate the advantages of the proposed algorithm compared to the existing algorithm in [33]. In Section 5, the numerical solution to each subproblem emerging from the augmented Lagrangian functional is discussed separately. In Section 6, we carry out numerical experiments with our method for dealing with problems in image denoising, image inpainting and image zooming to demonstrate its efficiency.

3. The existing algorithm

In this section, we give a brief review of the augmented Lagrangian method applied to Euler's elastica in the Tai-Hahn-Chung (THC) formulation [33]. It is difficult to solve the augmented Lagrangian functional (2.2) efficiently because of the non-differentiable quadratic term involved $|\boldsymbol{p}|$. Therefore, the authors introduce one more variable into the constrained problem (2.1), which satisfies

$$\boldsymbol{m}=\boldsymbol{n}\quad\text{and}\quad|\boldsymbol{m}|\le 1.$$

The constrained optimization problem (2.1) is reformatted as the following structure

$$\begin{aligned}&\min_{u,\boldsymbol{p},\boldsymbol{n},\boldsymbol{m}}\int_\Omega\Big(a+b(\nabla\cdot\boldsymbol{n})^2\Big)|\boldsymbol{p}|+\frac{\mu}{t}\int_\Omega|u-u_0|^t,\\&\text{s.t.}\quad\boldsymbol{p}=\nabla u;\quad\boldsymbol{n}=\boldsymbol{m};\quad|\boldsymbol{p}|=\boldsymbol{m}\cdot\boldsymbol{p};\quad|\boldsymbol{m}|\le 1.\end{aligned}\tag{3.1}$$

By the constraint $|\boldsymbol{m}|\le 1$, there exists the relationship $|\boldsymbol{p}|-\boldsymbol{m}\cdot\boldsymbol{p}\ge 0$, a.e., in Ω. Therefore, the quadratic penalty term $\int_\Omega(|\boldsymbol{p}|-\boldsymbol{m}\cdot\boldsymbol{p})^2$ in (2.2) can be relaxed to a first order penalty term. Then, the augmented Lagrangian functional for the constrained problem (3.1) is defined as follows

$$\begin{aligned}&L(u,\boldsymbol{p},\boldsymbol{n},\boldsymbol{m};\lambda_1,\boldsymbol{\lambda}_2,\boldsymbol{\lambda}_3)\\&=\int_\Omega\big(a+b(\nabla\cdot\boldsymbol{n})^2\big)|\boldsymbol{p}|+\frac{\mu}{t}\int_\Omega|u-u_0|^t+\int_\Omega(|\boldsymbol{p}|-\boldsymbol{m}\cdot\boldsymbol{p})\lambda_1+r_1\int_\Omega(|\boldsymbol{p}|-\boldsymbol{m}\cdot\boldsymbol{p})\\&\quad+\int_\Omega(\boldsymbol{p}-\nabla u)\cdot\boldsymbol{\lambda}_2+\frac{r_2}{2}\int_\Omega|\boldsymbol{p}-\nabla u|^2+\int_\Omega(\boldsymbol{n}-\boldsymbol{m})\cdot\boldsymbol{\lambda}_3+\frac{r_3}{2}\int_\Omega(\boldsymbol{n}-\boldsymbol{m})^2+\delta_R(\boldsymbol{m}),\end{aligned}\tag{3.2}$$

where $\boldsymbol{\lambda}_3$ is the Lagrange multiplier, r_3 is the positive constant and the set $R := \{\boldsymbol{m} \in \boldsymbol{L}^2(\Omega)||\boldsymbol{m}| \leq 1 \text{a.e. in } \Omega\}$ and δ_R is the indicator function defined by

$$\delta_R(\boldsymbol{m}) := \begin{cases} 0, & \boldsymbol{m} \in R, \\ +\infty, & \text{otherwise.} \end{cases}$$

In [33], the energy functional (3.2) is decomposed into a number of subproblems as follows:

$$\epsilon_1(u) = \frac{\mu}{t}\int_\Omega |u-u_0|^t - \int_\Omega \nabla u \cdot \boldsymbol{\lambda}_2 + \frac{r_2}{2}\int_\Omega (\boldsymbol{p}-\nabla u)^2, \tag{3.3a}$$

$$\epsilon_2(\boldsymbol{p}) = \int_\Omega \left(a + b(\nabla\cdot n)^2\right)|\boldsymbol{p}| + \int_\Omega (|\boldsymbol{p}| - \boldsymbol{m}\cdot\boldsymbol{p})\lambda_1 + r_1\int_\Omega (|\boldsymbol{p}| - \boldsymbol{m}\cdot\boldsymbol{p}) + \int_\Omega \boldsymbol{p}\cdot\boldsymbol{\lambda}_2 + \frac{r_2}{2}\int_\Omega (\boldsymbol{p}-\nabla u)^2, \tag{3.3b}$$

$$\epsilon_3(\boldsymbol{n}) = \int_\Omega b|\boldsymbol{p}|(\nabla\cdot\boldsymbol{n})^2 + \int_\Omega \boldsymbol{n}\cdot\boldsymbol{\lambda}_3 + \frac{r_3}{2}\int_\Omega (\boldsymbol{n}-\boldsymbol{m})^2, \tag{3.3c}$$

$$\epsilon_4(\boldsymbol{m}) = -(\lambda_1 + r_1)\int_\Omega \boldsymbol{p}\cdot\boldsymbol{m} - \int_\Omega \boldsymbol{m}\cdot\boldsymbol{\lambda}_3 + \frac{r_3}{2}\int_\Omega (\boldsymbol{n}-\boldsymbol{m})^2 + \delta_R(\boldsymbol{m}). \tag{3.3d}$$

The above subproblems are solved alternatively in one iteration of the algorithm. For the u-sub problem, since its Euler-Lagrange equation is a linear PDE, it is solved efficiently by the FFT. There are closed-form solutions for both the $\boldsymbol{p}$-sub and $\boldsymbol{m}$-sub problems referred to [33]. In the existing algorithm of THC, the most difficult and time-consuming part is to solve the Euler-Lagrange equation of $\boldsymbol{n}$-sub problem, which is

$$-2\nabla(b|\boldsymbol{p}|\nabla\cdot\boldsymbol{n}) + r_3(n-m) + \boldsymbol{\lambda}_3 = 0. \tag{3.4}$$

The coefficient of $\nabla\cdot\boldsymbol{n}$ is a variable in (3.4), which makes it difficult to handle. Aimed to use the FFT, a frozen coefficient method is applied to the coupled PDEs in [33]. However, there are some drawbacks caused by this method. First, the method needs the inner iterations. Since the coefficient of $\nabla\cdot\boldsymbol{n}$ is fixed artificially in each iteration inner iteration, it may require more iterations to achieve the convergence. Second, to solve the coupled PDEs (3.4), two FFT algorithms are necessary in an inner iteration. Considering the FFT used in u-sub problem, three times of the FFT are involved at least in one outer iteration of the algorithm in [33]. The above points will increase the computational costs of the algorithm in [33]. Besides, there exist cases that the domain or boundary condition assigned to the minimization problem (1.6) is not applicable to the FFT. Therefore, we consider to find a better way to solve the augmented Lagrangian functional (2.2).

4. The proposed algorithm

In this section, we propose a more efficient algorithm to solve the augmented Lagrangian functional (2.2). Unlike the algorithm in [33], we keep the two constraints

in the optimization problem (2.1) and introduce one more variable to denote the mean curvature, which is

$$h = \nabla \cdot \boldsymbol{n}.$$

We use the variable h to remove the variable coefficient in the n-sub problem (3.3c) in Tai-Hahn-Chung formulation [33]. Therefore, the Euler's elastica model (1.6) is reformulated as the following constrained optimization problem in our work

$$\begin{aligned} &\min_{u,\boldsymbol{p},\boldsymbol{n},h} \int_\Omega (a+bh^2)|\boldsymbol{p}| + \frac{\mu}{t}\int_\Omega |u-u_0|^t, \\ &\text{s.t.} \quad \boldsymbol{p} = \nabla u; \quad \boldsymbol{p} = |\boldsymbol{p}|\boldsymbol{n}; \quad h = \nabla\cdot\boldsymbol{n}. \end{aligned} \tag{4.1}$$

We use the augmented Lagrangian method in [34] to rewrite the constrained problem (4.1) into an unconstrained minimization problem. We define the augmented Lagrangian functional for (4.1) as follows

$$\begin{aligned} &L(u,\boldsymbol{p},\boldsymbol{n},h;\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\lambda_3) \\ &= \int_\Omega (a+bh^2)|\boldsymbol{p}| + \frac{\mu}{t}\int_\Omega |u-u_0|^t + \int_\Omega (\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})\cdot\boldsymbol{\lambda}_1 + \frac{r_1}{2}\int_\Omega (\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})^2 \\ &\quad + \int_\Omega (\boldsymbol{p}-\nabla u)\cdot\boldsymbol{\lambda}_2 + \frac{r_2}{2}\int_\Omega (\boldsymbol{p}-\nabla u)^2 + \int_\Omega (h-\nabla\cdot\boldsymbol{n})\lambda_3 + \frac{r_3}{2}\int_\Omega (h-\nabla\cdot\boldsymbol{n})^2. \end{aligned} \tag{4.2}$$

We apply an iterative algorithm to solve the saddle-point problem corresponding to the augmented Lagrangian functional (4.2); see Algorithm 4.1.

Algorithm 4.1: Augmented Lagrangian method for the Euler's elastica model

1. Initialization: u^0, $\boldsymbol{p}^0$, $\boldsymbol{n}^0$, h^0 and $\boldsymbol{\lambda}_1^0$, $\boldsymbol{\lambda}_2^0$, λ_3^0.
2. For $k = 0, 1, 2, \cdots$, compute $(u^k, \boldsymbol{p}^k, \boldsymbol{n}^k, h^k)$ as an approximate minimizer of the augmented Lagrangian functional with the Lagrange multiplier $\boldsymbol{\lambda}_1^{k-1}$, $\boldsymbol{\lambda}_2^{k-1}$ and λ_3^{k-1}, i.e.,

$$(u^k, \boldsymbol{p}^k, \boldsymbol{n}^k, h^k) \approx \arg\min L(u,\boldsymbol{p},\boldsymbol{n},h;\boldsymbol{\lambda}_1^{k-1},\boldsymbol{\lambda}_2^{k-1},\lambda_3^{k-1}). \tag{4.3}$$

3. Update the Lagrange multipliers:

$$\boldsymbol{\lambda}_1^k = \boldsymbol{\lambda}_1^{k-1} + r_1(\boldsymbol{p}^k - |\boldsymbol{p}^k|\boldsymbol{n}^k), \tag{4.4a}$$
$$\boldsymbol{\lambda}_2^k = \boldsymbol{\lambda}_2^{k-1} + r_2(\boldsymbol{p}^k - \nabla u^k), \tag{4.4b}$$
$$\lambda_3^k = \lambda_3^{k-1} + r_3(h^k - \nabla\cdot\boldsymbol{n}^k). \tag{4.4c}$$

Since the variables u, $\boldsymbol{p}$, $\boldsymbol{n}$, h are coupled together in the minimization problem (4.3), it is difficult to solve all the variables simultaneously. Therefore, we use the decomposition technique to separate the problem (4.3) into a number of sub minimization problems. The

energy functionals corresponding to each variable are given separately as follows:

$$E_1(u) = \frac{\mu}{t}\int_\Omega |u-u_0|^t - \int_\Omega \nabla u\cdot\boldsymbol{\lambda}_2 + \frac{r_2}{2}\int_\Omega (\boldsymbol{p}-\nabla u)^2, \tag{4.5a}$$

$$E_2(\boldsymbol{p}) = \int_\Omega (a+bh^2)|\boldsymbol{p}| + \int_\Omega (\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})\cdot\boldsymbol{\lambda}_1 + \frac{r_1}{2}\int_\Omega (\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})^2 + \int_\Omega \boldsymbol{p}\cdot\boldsymbol{\lambda}_2 + \frac{r_2}{2}\int_\Omega (\boldsymbol{p}-\nabla u)^2, \tag{4.5b}$$

$$E_3(\boldsymbol{n}) = -\int_\Omega |\boldsymbol{p}|\boldsymbol{n}\cdot\boldsymbol{\lambda}_1 + \frac{r_1}{2}\int_\Omega (\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})^2 - \int_\Omega \lambda_3\nabla\cdot\boldsymbol{n} + \frac{r_3}{2}\int_\Omega (h-\nabla\cdot\boldsymbol{n})^2, \tag{4.5c}$$

$$E_4(h) = \int_\Omega b|\boldsymbol{p}|h^2 + \int_\Omega \lambda_3 h + \frac{r_3}{2}\int_\Omega (h-\nabla\cdot\boldsymbol{n})^2. \tag{4.5d}$$

All subproblems (4.5a) to (4.5d) are efficiently solved by either closed-form solution or fast iteration method. For the minimization of (4.3), we alternatively solve the minimizer to each subproblem by using Algorithm 4.2.

Algorithm 4.2: Alternating minimization method for (4.3)

1. Initialization: $\widetilde{u}^0 = u^{k-1}$, $\widetilde{\boldsymbol{p}}^0 = \boldsymbol{p}^{k-1}$, $\widetilde{\boldsymbol{n}}^0 = \boldsymbol{n}^{k-1}$ and $\widetilde{h}^0 = h^{k-1}$.
2. For $l = 0,1,2,\cdots,L$ and fixed Lagrange multipliers $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_1^{k-1}$, $\boldsymbol{\lambda}_2 = \boldsymbol{\lambda}_2^{k-1}$ and $\lambda_3 = \lambda_3^{k-1}$: Compute $\widetilde{u}^l$ from (4.5a), $\widetilde{\boldsymbol{p}}^l$ from (4.5b), $\widetilde{\boldsymbol{n}}^l$ from (4.5c) and $\widetilde{h}^l$ from (4.5d).
3. $(u^k,\boldsymbol{p}^k,\boldsymbol{n}^k,h^k) = (\widetilde{u}^L,\widetilde{\boldsymbol{p}}^L,\widetilde{\boldsymbol{n}}^L,\widetilde{h}^L)$.

Here L can be chosen using some convergence test techniques. In this paper, we simply set $L = 1$. We will discuss the specific solution to each subproblem in the forthcoming section.

5. Numerical solutions for subproblems

In this section, we explain how to find the minimizer of each subproblem. We use a staggered grid system as in Fig. 1 to solve the energy functional minimization (4.5a) to (4.5d) and update the Lagrange multipliers from (4.4a) to (4.4c).

5.1. Notations

We first give some basic notations at the beginning. An image u is regarded as a function

$$u : \{1,\cdots,M\}\times\{1,\cdots,N\},$$

where $M,N \geq 2$.

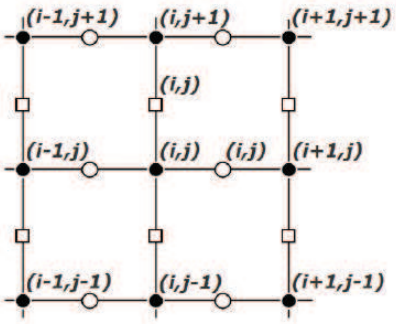


Figure 1: Grid definition. The rule of indexing variables in the augmented Lagrangian functional (2.2): u, h and λ_3 are defined on $\bullet$-nodes. The first and second component of $\boldsymbol{p}$, $\boldsymbol{n}$, $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are defined on $\circ$-nodes and $\square$-nodes, respectively.

We denote the Euclidean space $\mathbb{R}^{M\times N}$ as V and define another inner product vector space: $Q = V \times V$. For a given $(i,j) \in [1,M] \times [1,N]$, we see that

$$u \in V, \quad u(i,j) \in \mathbb{R}, \quad \text{and} \quad \boldsymbol{p} \in Q, \quad \boldsymbol{p}(i,j) = (p_1(i,j), p_2(i,j)) \in \mathbb{R}^2.$$

We also equip the space V and Q with the standard Euclidean inner products as follows

$$(u,v)_V = \sum_{i,j} u(i,j)v(i,j) \quad \text{and} \quad (\boldsymbol{p},\boldsymbol{q})_Q = (p_1,q_1)_V + (p_2,q_2)_V.$$

We will use the discrete backward and forward differential operators for $u \in V$, which are defined with periodic boundary condition as follows:

$$\partial_x^- u(i,j) = \begin{cases} u(i,j) - u(i-1,j), & 1 < i \le M, \\ u(1,j) - u(M,j), & i = 1, \end{cases}$$
$$\partial_y^- u(i,j) = \begin{cases} u(i,j) - u(i,j-1), & 1 < j \le N, \\ u(i,1) - u(i,N), & j = 1, \end{cases}$$
$$\partial_x^+ u(i,j) = \begin{cases} u(i+1,j) - u(i,j), & 1 \le i < M, \\ u(1,j) - u(M,j), & i = M, \end{cases}$$
$$\partial_y^+ u(i,j) = \begin{cases} u(i,j+1) - u(i,j), & 1 \le j < N, \\ u(i,1) - u(i,N), & j = N. \end{cases}$$

The discrete forward and backward gradient operator $\nabla : V \to Q$ are given as follows:

$$\nabla u(i,j) := (\partial_x^+ u(i,j), \partial_y^+ u(i,j)).$$

The corresponding discrete backward and forward adjoint operator $\mathrm{div} : Q \to V$ of $-\nabla$ is obtained:

$$\mathrm{div}\, \boldsymbol{p}(i,j) := \partial_x^- p_1(i,j) + \partial_y^- p_2(i,j).$$

When a variable defined on $\circ$-nodes (or $\square$-nodes) needs to be evaluated at $(i,j) \in \square$-nodes (or $\circ$-nodes), for example $\boldsymbol{p} = (p_1, p_2)$, we use the average operators:

$$A^{\square}_{i,j}(p_1) = \frac{p_1(i,j+1) + p_1(i-1,j+1) + p_1(i,j) + p_1(i-1,j)}{4},$$
$$A^{\circ}_{i,j}(p_2) = \frac{p_2(i+1,j) + p_2(i,j) + p_2(i+1,j-1) + p_2(i,j-1)}{4},$$

where p_1 and p_2 are defined on $\circ$-nodes and $\square$-nodes respectively. We need to define a special operator to measure the magnitude of $\boldsymbol{p}$ at $(i,j) \in \bullet$-nodes as follows:

$$|A|^{\bullet}_{i,j}(\boldsymbol{p}) = \Big(\Big(\frac{p_1(i,j) + p_1(i-1,j)}{2} \Big)^2 + \Big(\frac{p_2(i,j) + p_2(i,j-1)}{2} \Big)^2 \Big)^{\frac{1}{2}}.$$

We compute the divergence of $\boldsymbol{p}$ at $(i,j) \in \bullet$-nodes using the following operator:

$$\mathrm{div}^{\bullet}_{i,j}(\boldsymbol{p}) = p_1(i,j) - p_1(i-1,j) + p_2(i,j) - p_2(i,j-1).$$

5.2. Subproblems

In the following, for each subproblem, we denote the fixed Lagrange multipliers in the previous $(k-1)^{th}$ iteration as $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_1^{k-1}$, $\boldsymbol{\lambda}_2 = \boldsymbol{\lambda}_2^{k-1}$ and $\lambda_3 = \lambda_3^{k-1}$.

5.2.1. The u-sub problem

We denote a fixed variable $\widetilde{\boldsymbol{p}}^{l-1}$ as $\boldsymbol{p}$. For the u-sub problem (4.5a), the choice of t is either $t = 1$ or $t = 2$ for different applications. We discuss the minimization of u according to the value of t separately:

Case 1: $t = 1$

For this case, we introduce a new variable v into the u-sub problem and rewrite it as a constrained problem as follows

$$\min_{u,v} \mu \int_\Omega |v - u_0| - \int_\Omega \nabla u \cdot \boldsymbol{\lambda}_2 + \frac{r_2}{2} \int_\Omega (\boldsymbol{p} - \nabla u)^2, \qquad \text{s.t.} \quad v = u. \tag{5.1}$$

We apply the augmented Lagrangian method to (5.1) and obtain the following saddle-point problem

$$\begin{aligned} L(u,v;\lambda_4) =& \mu \int_\Omega |v - u_0| - \int_\Omega \nabla u \cdot \boldsymbol{\lambda}_2 + \frac{r_2}{2} \int_\Omega (\boldsymbol{p} - \nabla u)^2 \\ &+ \int_\Omega \lambda_4 (v - u) + \frac{r_4}{2} \int_\Omega (v - u)^2, \end{aligned} \tag{5.2}$$

where λ_4 is the Lagrange multiplier and r_4 is a positive constant. We can regard (5.2) as a self-contained minimization problem and use the following iterative algorithm to solve the variables u, v and λ_4; see Algorithm 5.1.

Algorithm 5.1: Augmented Lagrangian method for the u-sub problem

1. Initialization: u^0 and v^0.
2. For $k = 0, 1, 2, \cdots$, compute (u^k, v^k) as an approximate minimizer of the augmented Lagrangian functional with the Lagrange multiplier λ_4^{k-1}, i.e.,

$$(u^k, v^k) \approx \arg\min L(u, v; \lambda_4^{k-1}). \tag{5.3}$$

3. Update the Lagrange multipliers:

$$\lambda_4^k = \lambda_4^{k-1} + r_4(v^k - u^k). \tag{5.4}$$

Algorithm 5.1 can be easily incorporated into Algorithm 4.1. Let us consider the minimization (5.3) under the structure of Algorithm 4.1. Therefore, for fixed $\widetilde{\boldsymbol{p}}$, we aim to find $\widetilde{u}^l$ and $\widetilde{v}^l$ in Algorithm 4.2. We denote the fixed Lagrange multiplier λ_4^{k-1} as λ_4 for ease of the explanation. To minimize u and v, we separate the functional (5.2) into the following two subproblems:

$$\min_u \int_\Omega \nabla u \cdot \boldsymbol{\lambda}_2 + \frac{r_2}{2} \int_\Omega (\boldsymbol{p} - \nabla u)^2 - \int_\Omega \lambda_4 u + \frac{r_4}{2} \int_\Omega (v - u)^2, \tag{5.5}$$

for a given v, and

$$\min_v \mu \int_\Omega |v - u_0| + \int_\Omega \lambda_4 v + \frac{r_4}{2} \int_\Omega (v - u)^2, \tag{5.6}$$

for a given u.

First, the Euler-Lagrange equation of (5.5) gives us a linear PDE

$$(r_4 - r_2\Delta)u = \lambda_4 + r_4 v - \nabla \cdot \boldsymbol{\lambda}_2 - r_2 \nabla \cdot \boldsymbol{p}. \tag{5.7}$$

The PDE problem (5.7) can be solved efficiently by a wide range of linear iterative methods, such as Jacobi method, Gauss-Seidel method. In this work, we use one sweep of the Gauss-Seidel iteration to approximate the solution $\widetilde{u}^l$.

On the other hand, we consider the solution to the v-sub problem. Let $w = u - \lambda_4/r_4$. We can rewrite (5.6) as follows

$$\min_v \mu \int_\Omega |v - u_0| + \frac{r_4}{2} \int_\Omega (v - w)^2. \tag{5.8}$$

The optimal value of the variable v in (5.8) can be easily computed by using the shrinkage operator. For fixed u, we simply compute

$$\widetilde{v}^l(i,j) = \max\Big\{0, 1 - \frac{\mu}{r_4 |w(i,j) - u_0(i,j)|}\Big\} \cdot \big(w(i,j) - u_0(i,j)\big) + u_0(i,j).$$

Case 2: $t = 2$

For this case, the minimization problem related to u can be simplified as

$$\min_u \frac{\mu}{2}\int_\Omega (u-u_0)^2 - \int_\Omega \nabla u \cdot \boldsymbol{\lambda}_2 + \frac{r_2}{2}\int_\Omega (\boldsymbol{p} - \nabla u)^2. \tag{5.9}$$

Therefore, the optimal condition of (5.9) gives us a linear equation of u, which is

$$(\mu - r_2\Delta)u = \mu u_0 - \nabla\cdot\boldsymbol{\lambda}_2 - r_2\nabla\cdot\boldsymbol{p}. \tag{5.10}$$

Similarly, for fixed $\boldsymbol{p}$, we use one sweep of the Gauss-Seidel iteration to compute the updated $\widetilde{u}^l$.

5.2.2. The p-sub problem

We denote fixed variables $\widetilde{u}^l$, $\widetilde{\boldsymbol{n}}^{l-1}$ and $\widetilde{h}^{l-1}$ as u, $\boldsymbol{n}$ and h, respectively. For the $\boldsymbol{p}$-sub problem, it is difficult to solve the Euler-Lagrange equation of (4.5b) due to the non-differentiability element $|\boldsymbol{p}|$ in the quadratic term. To avoid this situation, we consider to apply a fixed-point formulation to the constraint $\boldsymbol{p} = |\boldsymbol{p}|\boldsymbol{n}$ in the k^{th} iteration, which gives

$$\boldsymbol{p} = |\boldsymbol{p}^{k-1}|\boldsymbol{n}.$$

To get rid of the nonlinearity and non-differentiability term, we use $\boldsymbol{p} - |\boldsymbol{p}^{k-1}|\boldsymbol{n}$ to replace $\boldsymbol{p} - |\boldsymbol{p}|\boldsymbol{n}$ in the quadratic penalty term in (4.5b). Therefore, we reformulate the energy functional of the variable $\boldsymbol{p}$ as follows

$$E_2(\boldsymbol{p}) = \int_\Omega (a + bh^2 - \boldsymbol{n}\cdot\boldsymbol{\lambda}_1)|\boldsymbol{p}| + \frac{r_1+r_2}{2}\int_\Omega \Big(\boldsymbol{p} - \frac{r_1|\boldsymbol{p}^{k-1}|\boldsymbol{n} + r_2\nabla u - \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2}{r_1+r_2}\Big)^2.$$

For the simplicity, let

$$c = a + bh^2 - \boldsymbol{n}\cdot\boldsymbol{\lambda}_1 \quad \text{and} \quad \boldsymbol{q} = r_1|\boldsymbol{p}^{k-1}|\boldsymbol{n} + r_2\nabla u - \boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2.$$

Here, c can be either positive or negative. For each case, there is the closed-form solution for solving $\boldsymbol{p}$-sub problem. If c is positive, then we have the following closed-form solution for $\boldsymbol{p}$

$$\boldsymbol{p}(i,j) = \frac{1}{r_1+r_2}\max\Big\{0, 1 - \frac{c}{|\boldsymbol{q}(i,j)|}\Big\}\cdot\boldsymbol{q}(i,j).$$

And if c is negative, the solution for $\boldsymbol{p}$ is

$$\boldsymbol{p}(i,j) = \frac{1}{r_1+r_2}\Big(1 - \frac{c}{|\boldsymbol{q}(i,j)|}\Big)\cdot\boldsymbol{q}(i,j),$$

which belongs to the case when c is positive.

Therefore, for fixed u, $\boldsymbol{n}$ and h, the closed-form solution for $\boldsymbol{p}$ in (4.5b) can be summarized as

$$\widetilde{\boldsymbol{p}}^l = \frac{1}{r_1+r_2}\max\Big\{0, 1-\frac{c}{|\boldsymbol{q}|}\Big\}\cdot\boldsymbol{q}. \tag{5.11}$$

According to the rule of indexing variables in Fig. 1, the first and second component of $\boldsymbol{p}$, $\boldsymbol{n}$, $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ are defined on $\circ$-nodes and $\square$-nodes, respectively. The discretization of c and $\boldsymbol{q}$ at $(i,j)\in\circ$-node is obtained as follows:

$$\begin{aligned}
c(i,j) &= a + b\Big(\frac{h(i+1,j)+h(i,j)}{2}\Big)^2 - \lambda_{11}(i,j)n_1(i,j) - A^{\circ}_{i,j}(\lambda_{12})A^{\circ}_{i,j}(n_2),\\
q_1(i,j) &= r_2\Big(u(i+1,j)-u(i,j)\Big) + r_1|\boldsymbol{p}^{k-1}|n_1(i,j) - \lambda_{11}(i,j) - \lambda_{21}(i,j),\\
q_2(i,j) &= \frac{r_2}{2}\Big(\frac{u(i+1,j+1)+u(i,j+1)}{2} - \frac{u(i+1,j-1)+u(i,j-1)}{2}\Big)\\
&\quad + r_1|\boldsymbol{p}^{k-1}|A^{\circ}_{i,j}(n_2) - A^{\circ}_{i,j}(\lambda_{12}) - A^{\circ}_{i,j}(\lambda_{22}).
\end{aligned}$$

Similarly, a discretization of c and $\boldsymbol{q}$ at $(i,j)\in\square$-node is obtained as follows:

$$\begin{aligned}
c(i,j) &= a + b\Big(\frac{h(i,j+1)+h(i,j)}{2}\Big)^2 - A^{\square}_{i,j}(\lambda_{11})A^{\square}_{i,j}(n_1) - \lambda_{12}(i,j)n_2(i,j),\\
q_1(i,j) &= \frac{r_2}{2}\Big(\frac{u(i+1,j+1)+u(i+1,j)}{2} - \frac{u(i-1,j+1)+u(i-1,j)}{2}\Big)\\
&\quad + r_1|\boldsymbol{p}^{k-1}|A^{\square}_{i,j}(n_1) - A^{\square}_{i,j}(\lambda_{11}) - A^{\square}_{i,j}(\lambda_{21}),\\
q_2(i,j) &= r_2\Big(u(i,j+1)-u(i,j)\Big) + r_1|\boldsymbol{p}^{k-1}|n_2(i,j) - \lambda_{12}(i,j) - \lambda_{22}(i,j).
\end{aligned}$$

5.2.3. The $\boldsymbol{n}$-sub problem

We denote fixed variables $\widetilde{\boldsymbol{p}}^l$ and $\widetilde{h}^{l-1}$ as $\boldsymbol{p}$ and h, respectively. For the $\boldsymbol{n}$-sub problem, the Euler-Lagrange equation for the energy (4.5c) is the following linear coupled PDEs

$$-r_3\nabla(\nabla\cdot\boldsymbol{n}) + r_1|\boldsymbol{p}|^2\boldsymbol{n} = r_1\boldsymbol{p}|\boldsymbol{p}| + \boldsymbol{\lambda}_1|\boldsymbol{p}| - r_3\nabla h - \nabla\lambda_3. \tag{5.13}$$

Note that the operator $\nabla(\nabla\cdot)$ is singular. Due to the possibility of $|\boldsymbol{p}|=0$, we add a quadratic penalty term to (4.5c) to avoid the singularity. Therefore, we rewrite $E_3(\boldsymbol{n})$ as follows

$$\begin{aligned}
E_3(\boldsymbol{n}) = -\int_\Omega \boldsymbol{\lambda}_1\cdot|\boldsymbol{p}|\boldsymbol{n} + \frac{r_1}{2}\int_\Omega(\boldsymbol{p}-|\boldsymbol{p}|\boldsymbol{n})^2 - \int_\Omega\lambda_3\nabla\cdot\boldsymbol{n}\\
+ \frac{r_3}{2}\int_\Omega(h-\nabla\cdot\boldsymbol{n})^2 + \frac{\gamma}{2}\int_\Omega(\boldsymbol{n}-\boldsymbol{n}^{k-1})^2,
\end{aligned} \tag{5.14}$$

where γ is a positive constant. In the experiments, γ could be chosen to be a very small number.

We have the following optimality condition for the $\boldsymbol{n}$-sub problem (5.14) by its Euler-Lagrange equation

$$\Big(\gamma + r_1|\boldsymbol{p}|^2 - r_3\nabla\mathrm{div}\Big)\boldsymbol{n} = \gamma\boldsymbol{n}^{k-1} + r_1\boldsymbol{p}|\boldsymbol{p}| + \boldsymbol{\lambda}_1|\boldsymbol{p}| - r_3\nabla h - \nabla\lambda_3. \tag{5.15}$$

Eq. (5.15) is coupled PDEs of the variable $\boldsymbol{n} = (n_1, n_2)$. When we compute the component n_1, we use the n_2 in previous iteration

$$(\gamma + r_1|\boldsymbol{p}|^2 - r_3\partial_x^2)n_1 = \gamma n_1^{k-1} + r_3\partial_x\partial_y n_2 + r_1 p_1|\boldsymbol{p}| + \lambda_{11}|\boldsymbol{p}| - r_3\partial_x h - \partial_x\lambda_3,$$

and vice versa, when solve n_2, we use the n_1 in previous iteration

$$(\gamma + r_1|\boldsymbol{p}|^2 - r_3\partial_y^2)n_2 = \gamma n_2^{k-1} + r_3\partial_y\partial_x n_1 + r_1 p_2|\boldsymbol{p}| + \lambda_{12}|\boldsymbol{p}| - r_3\partial_y h - \partial_y\lambda_3.$$

Similarly to the u-sub problem, the one sweep Gauss-Seidel iteration gives the updated $\widetilde{\boldsymbol{n}}^l$. Compared to the frozen coefficient FFT method in [33], the proposed method for the $\boldsymbol{n}$-sub problem is easy to implement and solves the PDEs with low computational cost.

5.2.4. The h-sub problem

We denote fixed variables $\widetilde{\boldsymbol{p}}^l$ and $\widetilde{\boldsymbol{n}}^l$ as $\boldsymbol{p}$ and $\boldsymbol{n}$, respectively. For the h-sub problem, we have the Euler-Lagrange equation of the functional (4.5d) as follows

$$(2b|\boldsymbol{p}| + r_3)h = r_3\nabla\cdot\boldsymbol{n} - \lambda_3. \tag{5.16}$$

We can obtain a closed-form solution of h by solving the first-order equation (5.16). For fixed $\boldsymbol{p}$ and $\boldsymbol{n}$, the minimizer of $E_4(h)$ is solved as follows

$$\widetilde{h}^l = \frac{r_3\mathrm{div}^{\bullet}_{i,j}(\boldsymbol{n}) - \lambda_3}{2b|A|^{\bullet}_{i,j}(\boldsymbol{p}) + r_3}.$$

5.2.5. Update Lagrange multipliers

We update Lagrange multipliers $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, λ_3 and λ_4 using the staggered grid as shown in Fig. 1. The discretized form of equations from (4.4a) to (4.4c) and (5.4) is written as:

$$\begin{aligned}
&\lambda_{11}^k = \lambda_{11}^{k-1} + r_1(p_1^k - |\boldsymbol{p}^k|n_1^k) && \text{at } \circ\text{-nodes},\\
&\lambda_{12}^k = \lambda_{12}^{k-1} + r_1(p_2^k - |\boldsymbol{p}^k|n_2^k) && \text{at } \square\text{-nodes},\\
&\lambda_{21}^k = \lambda_{21}^{k-1} + r_2(p_1^k - \partial_1^+ u^k) && \text{at } \circ\text{-nodes},\\
&\lambda_{22}^k = \lambda_{22}^{k-1} + r_2(p_2^k - \partial_2^+ u^k) && \text{at } \square\text{-nodes},\\
&\lambda_3^k = \lambda_3^{k-1} + r_3(h^k - \mathrm{div}^{\bullet}_{i,j}(\boldsymbol{n})) && \text{at } \bullet\text{-nodes},\\
&\lambda_4^k = \lambda_4^{k-1} + r_4(v^k - u^k) && \text{at } \bullet\text{-nodes}.
\end{aligned}$$

We summarize the solutions to subproblems of both the existing algorithm in [33] and the proposed algorithm in Table 1.

Table 1: Summary of methods for solving subproblems in the existing algorithm in [33] and the proposed algorithm.

Subproblem	Our proposed algorithm	The existing algorithm in [33]
v-sub problem	closed-form solution	closed-form solution
u-sub problem	GS of linear PDE	FFT of linear PDE
$\boldsymbol{p}$-sub problem	closed-form solution	closed-form solution
$\boldsymbol{n}$-sub problem	GS of coupled linear PDEs	FFT of coupled linear PDEs
h/m-sub problem	first order equation	closed-form solution

6. Numerical examples

In this section, we present the numerical results to demonstrate the efficiency of the proposed algorithm. Although we emphasis image denoising in our work, the examples of image inpainting and zooming are also included. In all the experiments, we set $\gamma = 0.01$. The system is a 2.4 GHz CPU and 4GB memory.

During the iterations, we define the relative error of the solution $\{u^k | k = 1, 2, \cdots\}$ and we stop the iteration when the relative error is less than the given error tolerance, that is

$$\frac{\|u^k - u^{k-1}\|_{L^1}}{\|u^{k-1}\|_{L^1}} \leq \epsilon. \tag{6.1}$$

We also define the numerical energy of the Euler's elastica model as follows

$$E^k = \int_\Omega \big(a + b(h^k)^2\big)|\boldsymbol{p}^k| + \frac{\mu}{2}\int_\Omega (v^k - u_0)^2. \tag{6.2}$$

We monitor the relative residuals of variables in (4.3), which are defined as

$$\begin{aligned}&(R_1^k, R_2^k, R_3^k, R_4^k)\\ =&\Big(\frac{\|\boldsymbol{p}^k - |\boldsymbol{p}^k|\boldsymbol{n}^k\|_{L^1}}{|\Omega|}, \frac{\|\boldsymbol{p}^k - \nabla u^k\|_{L^1}}{|\Omega|}, \frac{\|h - \nabla\cdot\boldsymbol{n}\|_{L^1}}{|\Omega|}, \frac{\|v-u\|_{L^1}}{|\Omega|}\Big).\end{aligned} \tag{6.3}$$

Correspondingly, the relative errors of the Lagrange multipliers are obtained by

$$\begin{aligned}&(L_1^k, L_2^k, L_3^k, L_4^k)\\ =&\Big(\frac{\|\boldsymbol{\lambda}_1^k - \boldsymbol{\lambda}_1^{k-1}\|_{L^1}}{\|\boldsymbol{\lambda}_1^{k-1}\|_{L^1}}, \frac{\|\boldsymbol{\lambda}_2^k - \boldsymbol{\lambda}_2^{k-1}\|_{L^1}}{\|\boldsymbol{\lambda}_2^{k-1}\|_{L^1}}, \frac{\|\lambda_3^k - \lambda_3^{k-1}\|_{L^1}}{\|\lambda_3^{k-1}\|_{L^1}}, \frac{\|\lambda_4^k - \lambda_4^{k-1}\|_{L^1}}{\|\lambda_4^{k-1}\|_{L^1}}\Big).\end{aligned} \tag{6.4}$$

In addition, we use the signal-to-noise ratio (SNR) to estimate the difference between restored image u and original image u_c for image denoising applications

$$SNR(u^k, u_c) = 10\log_{10}\Big(\frac{\sum_{i,j}(u^k(i,j) - a_1)^2}{\sum_{i,j}(|u^k(i,j) - u_c(i,j)| - a_2)^2}\Big), \tag{6.5}$$

where a_1 and a_2 are the average of u^k and $u^k - u_c$, respectively.

In the Euler's elastica functional (1.6), there are three parameters: a, b and μ. By using the augmented Lagrangian method, we have four more parameters r_1, r_2, r_3 and r_4 associated with Lagrange multipliers $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, λ_3 and λ_4, respectively. We set the values of these parameters according to different applications. Most notably, the parameter μ is chosen to be quite different for image denoising, inpainting and zooming examples. We will indicate the choice of these parameters in each example.

6.1. Image denoising

In this subsection, we explore the numerical results of the proposed algorithm for image denoising applications. We consider denoising examples with Gaussian white noise and Salt & pepper noise.

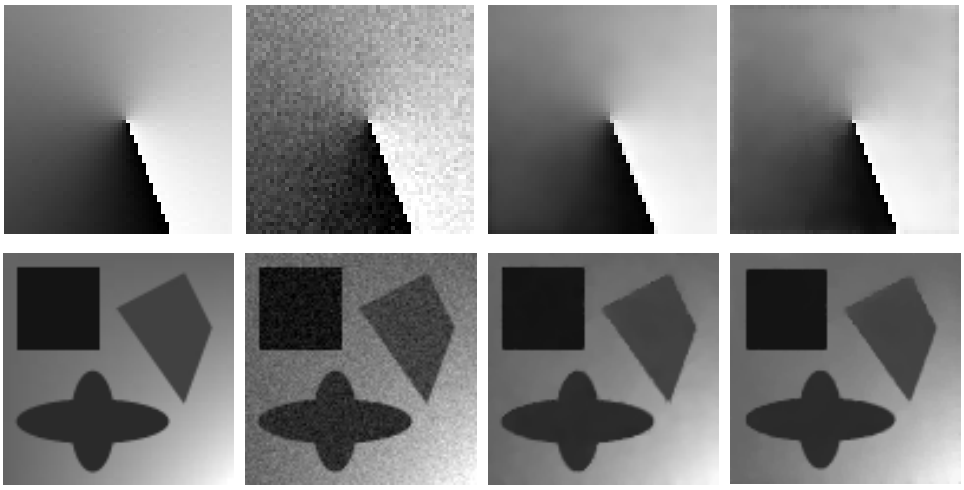
6.1.1. Comparison examples with the existing algorithm (THC)

In the first place, we compare the proposed algorithm with the existing algorithm (THC) in [33]. Our algorithm is implemented in C++ and THC algorithm is implemented in Matlab environment. For a fair comparison, we set $a = 1$, $b = 10$, $\mu = 100$ and $r_1 = 2$ for both algorithms and stop the iteration if the relative error of the solution $\{u^k | k = 1, 2, \cdots\}$ satisfies the same error tolerance. In Fig. 2, we choose two test images: Test (Row one) and Shapes (Row two) and add Gaussian white noise with mean zero and standard deviation 10 to them. We use $\epsilon = 3\times10^{-4}$ for example Test and $\epsilon = 2\times10^{-4}$ for example Shapes and display the results from the Euler's elastica energy (1.6) in Fig. 2. We list the comparison results of the computational time, SNR and iteration number in Table 2.

Table 2: The computational time (sec) of our proposed algorithm is compared with the method (THC) in [33] for the results in Fig. 2. The computational time is measured in seconds.

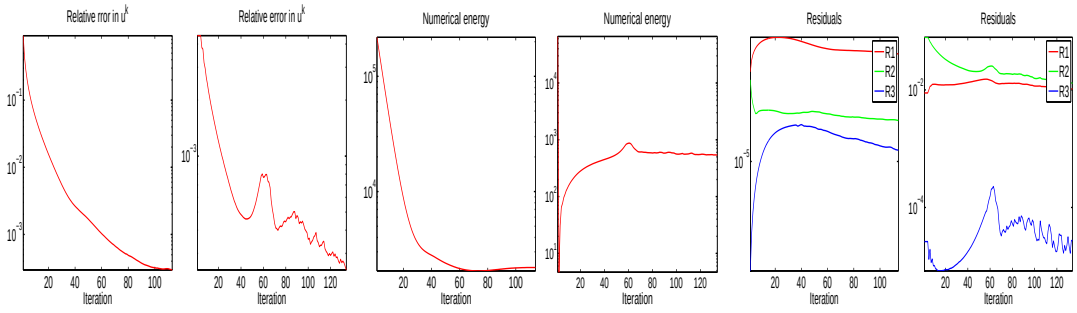
Images	Size	Our proposed algorithm			The existing algorithm (THC)		
		SNR	Iteration	Time	SNR	Iteration	Time
Test	60×60	26.60	113	1.16	25.60	134	30.68
Shapes	100×100	26.17	111	3.03	26.30	145	94.77

In comparison to the existing algorithm in [33], our algorithm can usually save the outer iteration to achieve the same relative error; see Table 2. Besides, the subproblem for the variable n in THC requires inner iterations for the convergence, while the inner iteration is only once for u-sub problem and n-subproblem in the proposed Euler's elastica formulation. Therefore, our algorithm can save computational cost in both inner iteration and outer iteration, which makes it superior to the previous algorithm in [33]. In Fig. 3, plots of relative error, numerical energy and residuals versus iteration numbers are shown for Test and Shapes in Table 2. From these plots, we can see that the relative error in u^k (6.1), numerical energy (6.2) and residuals (6.3) of the proposed algorithm converge more stable compared to THC algorithm, which demonstrate that the proposed algorithm is more reliable and efficient in practice.

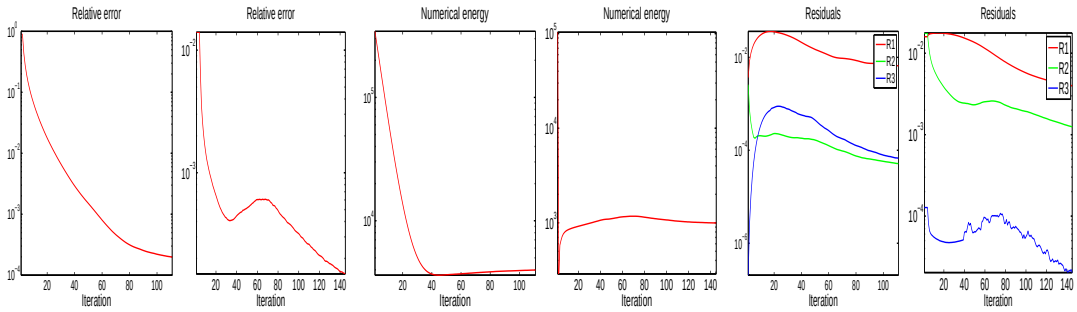


(a) Clean image (b) Noisy image (c) Our Algorithm (d) THC

Figure 2: Denosing comparison results. For Test (Row one), we set $r_2 = 500$, $r_3 = 100$ in proposed algorithm and $r_2 = 2$, $r_3 = 600$ in THC. For Shapes (Row two), we set $r_2 = 500$, $r_3 = 10$ in proposed algorithm and $r_2 = 50$, $r_3 = 100$ in THC.



(a) Relative error from Test (b) Numerical energy from Test (c) Residuals from Test

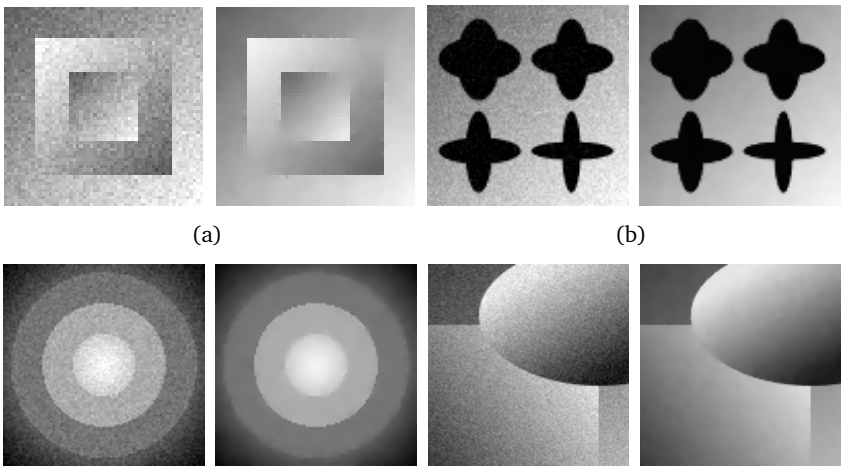


(d) Relative error from Shapes (e) Numerical energy from Shapes (f) Residuals from Shapes

Figure 3: Plots of (6.1), (6.2) and (6.3) values versus iteration numbers for examples shown in Fig. 2. In each pair, the left one is from our proposed algorithm and the right one is from THC.

6.1.2. Examples on synthetic and real-world images

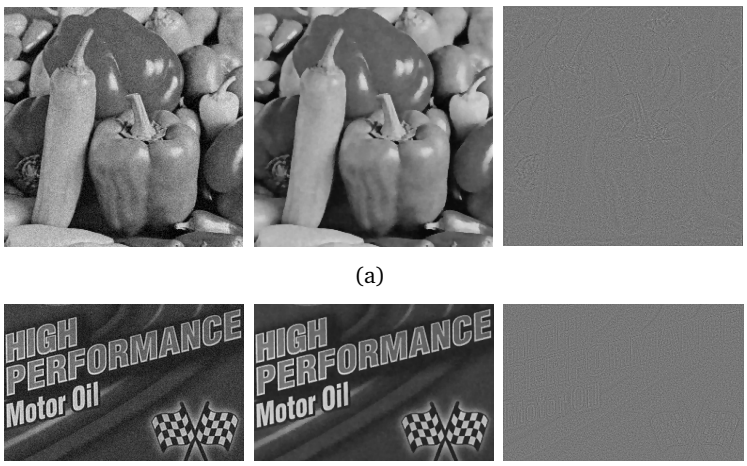
In Figs. 4 and 5, we show the denoising results of our proposed algorithm for synthetic images and real-world images, respectively. We add the Gaussian white noise with mean zero and the standard deviation 10 to the test images and use the L^2 data fitting term ($t = 2$) to remove the noise. In Fig. 4, we display the noisy images and the restored images and in Fig. 5, from left to right, we show the noisy images, the restored images



(a) (b)

(c) (d)

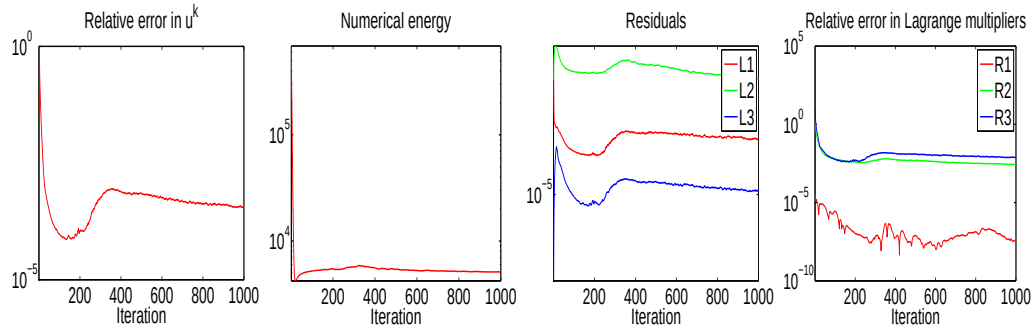
Figure 4: Euler's elastica based image denoising for Gaussian noise. We set $a = 1$, $b = 2$, $\mu = 100$, $r_1 = 2$, $r_2 = 200$ and $r_3 = 20$. The tolerance is $1.2 \cdot 10^{-4}$ for (a), $1 \cdot 10^{-4}$ for (b), (c) and 3.5×10^{-4} for (d).



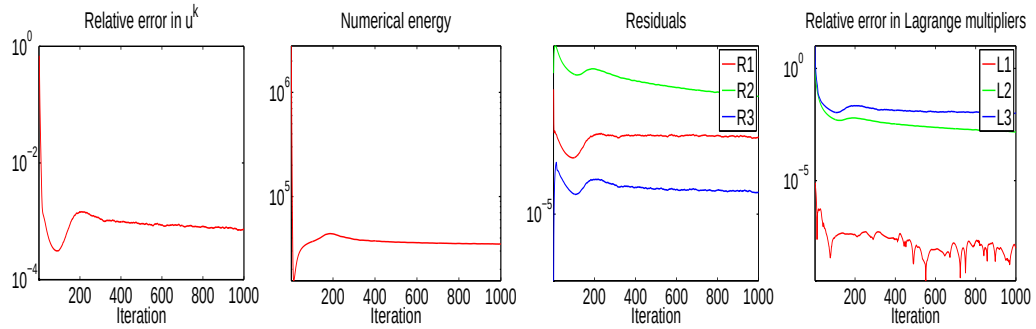
(a)

(b)

Figure 5: Euler's elastica based image denoising for Gaussian noise. We set $a = 1$, $b = 2$, $\mu = 180$, $r_1 = 2$, $r_2 = 200$ and $r_3 = 20$. The tolerance is 3.5×10^{-4}.



(a) Example in Fig. 4 (d)



(b) Example in Fig. 5 (a)

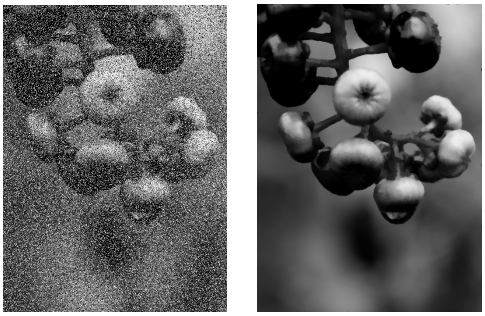
Figure 6: Plots of (6.1), (6.2), (6.3) and (6.4) values versus iteration numbers for examples shown in Fig. 4 (d) and Fig. 5 (a).

and the superimposed differences between the noisy images and the restored images. We summarize the size of images, SNR, number of outer iteration k and computational time in Table 3 for the test images in Figs. 4 and 5.

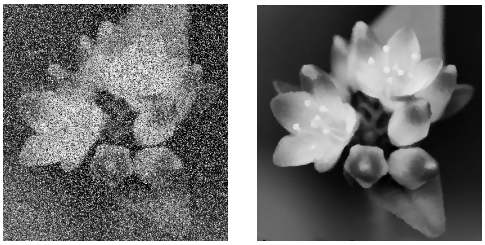
Table 3: The size of images and the SNR for images in Figs. 4 and 5 are shown. The computational time is measured in seconds.

Images	Size	SNR	# of iteration	Time (sec)
Fig. 4(a)	60×60	21.54	92	0.99
Fig. 4(b)	100×100	26.20	104	2.92
Fig. 4(c)	100×100	23.12	83	2.33
Fig. 4(d)	128×128	25.59	93	4.27
Fig. 5(a)	256×256	18.81	67	12.10
Fig. 5(b)	332×216	17.40	93	18.14

Based on synthetic image (d) in Fig. 4 and real image (a) in Fig. 5, we plot the relative error in u^k (6.1), numerical energy (6.2), relative residuals (6.3), relative errors of Lagrange multipliers (6.4) and SNR (6.5) versus the outer iterations k in Fig. 6. Here, all the graphs are drawn up to 1000 iterations. By doing so, we can easily observe that our algorithm converges well in the numerical experiments.



(a)



(b)

Figure 7: Euler's elastica based image denoising for salt & pepper noise. We set $a = 5$, $b = 20$, $\mu = 5$, $r_1 = 1$, $r_2 = 10$, $r_3 = 1$ and $r_4 = 20$ for both (a) and (b). The tolerance is 1.7×10^{-3} for (a) and 1.8×10^{-3} for (b).

In Fig. 7, we show the numerical results for image denoising applications with images corrupted by the salt & pepper noise. We add the salt & pepper noise with a noise density 0.4 to both test images. For this case, we use the L^1 data fitting term ($t = 1$) in the functional (1.6). We display the noisy images and restored images in Fig. 7 and list the size of images, SNR, number of outer iteration k and computational time in Table 4.

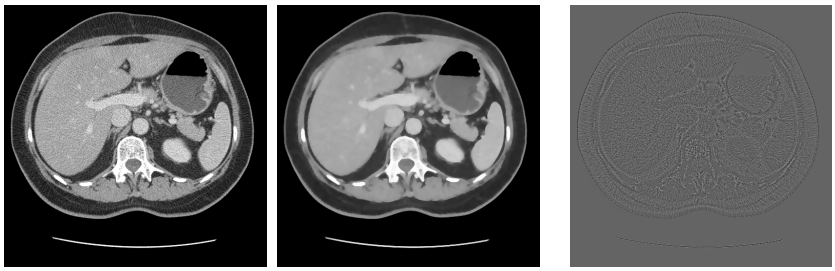
Table 4: The size of images and the SNR for images in Fig. 7 are shown. The computational time is measured in seconds.

Images	Size	SNR	# of iteration	Time (sec)
Fig. 7(a)	700×512	19.53	50	51.87
Fig. 7(b)	420×400	19.51	54	26.16

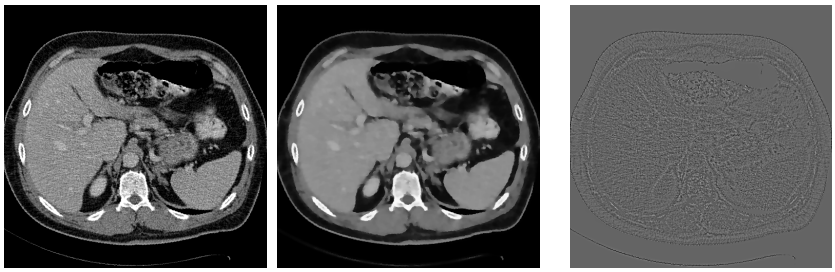
In these experiments, for both L^1 and L^2 data fidelity term, the proposed algorithm can produce the restored results of good quality within 100 iteration in most cases. In comparison to the previous algorithms for handling Euler's elastica, our algorithm has the advantages of fast convergence, which is very crucial for solving the high order model.

6.1.3. Examples on medical images

It is well-known that noise exists in computed tomography (CT) images. We conduct two experiments of Euler's elastica model on really liver CT data. Two slices of liver CT images



(a)



(b)

Figure 8: Euler's elastica based image denoising for CT liver images. We set $a=1$, $b=10$, $\mu=50$, $r_1=2$, $r_2=200$ and $r_3=10$ for both (a) and (b). The tolerance is 6×10^{-4} for (a) and 1.3×10^{-3} for (b).

are selected from two data set, which are displayed in the fist column of Fig. 8. We use the L^2 data fidelity term ($t=2$) in (4.1) for both experiments. The restored images and the superimposed differences between original images and restored images are shown in the second and third column of Fig. 8, respectively. As illustrated by these two experiments results, our Euler's elastica model can successfully remove the noise containing in CT images.

6.2. Image inpainting

In this subsection, we illustrate the efficiency of the proposed algorithm via examples in image inpainting. Suppose an image u_0 has a local patch damaged or missing, the desirable task of inpainting is to reconstruct the domain by using the information from surrounding areas. For the inpainting domain $D\subset\Omega$, we use the Euler's elastica energy to interpolate the image values by solving the following minimization problem

$$\min_u \int_\Omega \Big(a+b\Big(\nabla\cdot\frac{\nabla u}{|\nabla u|}\Big)^2\Big)|\nabla u|+\frac{\overline{\mu}}{t}\int_\Omega |u-u_0|^t, \tag{6.6}$$

where $\overline{\mu}$ is defined as μ times an indicator function

$$\overline{\mu}=\begin{cases}\mu, & \text{if } u\in\Omega\backslash D,\\ 0, & \text{if } u\in D.\end{cases} \tag{6.7}$$

For image inpainting, we can use either the L^1 ($t=1$) or L^2 ($t=2$) data fitting term in the functional (6.6). Compared to the image denoising case, the difference lies in the v-sub problem if the L^1 data fitting is used otherwise the difference exists in the u-sub problem when the L^2 data fitting is concerned. More specifically, we give the solution to the v-sub problem and the u-sub problem under these two cases as follows:

- $t=1$:
  In the discrete setting, for a grid point $(i,j)\in\Omega\backslash D$, the minimizer of $v(i,j)$ is

$$v(i,j)=\text{shrink}\Big(u(i,j)-\frac{\lambda_4(i,j)}{r_4}-u_0(i,j),\frac{\mu}{r_4}\Big)+u_0(i,j).$$

  For a grid point $(i,j)\in D$, the minimizer of $v(i,j)$ is

$$v(i,j)=u(i,j)-\frac{\lambda_4(i,j)}{r_4}.$$

- $t=2$:
  For a grid point $(i,j)\in\Omega\backslash D$, the minimizer of $u(i,j)$ is solved by the equation

$$(\mu-r_2\Delta)u(i,j)=\mu u_0(i,j)-r_2\text{div}\boldsymbol{p}(i,j)-\text{div}\boldsymbol{\lambda}_2(i,j).$$

  For a grid point $(i,j)\in D$, the minimizer of $u(i,j)$ is

$$-r_2\Delta u(i,j)=-r_2\text{div}\boldsymbol{p}(i,j)-\text{div}\boldsymbol{\lambda}_2(i,j).$$

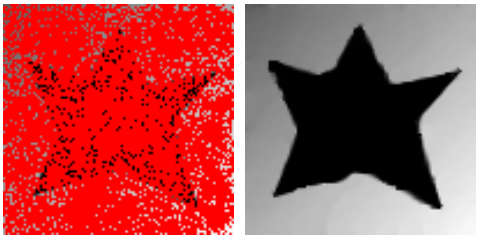
  Similar to the case of the $\boldsymbol{n}$-sub problem, we avoid the singularity of the above linear system by adding a quadratic penalty term to the energy functional.

In Figs. 9 and 10, we display the inpainting results from the elastica energy (6.6) with $t=1$ and $t=2$, respectively. Since we need u to be close to u_0 in $\Omega\backslash D$, we set μ to be 1000 for the test images in Figs. 9 and 10. In Table 5, we show the number of unknowns in the inpainting domain, the size of images, SNR, number of outer iteration k and computational time corresponding to the results in Figs. 9 and 10.

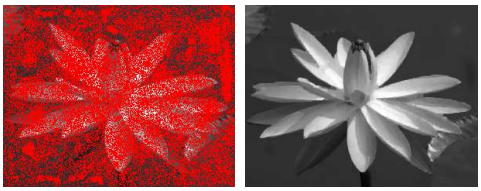
From these experiments, we can see that Euler's elastica energy can successfully recover the shapes for the inpainting domain even when the portion of unknown pixels are very large in the test images, i.e., 84.9% in Fig. 9 and 69.96% in Fig. 10.

Table 5: The number of unknowns in the inpainting domains, the size of images and the SNR for images in Figs. 9 and 10 are shown. The computational time is measured in seconds.

Images	Size	Unknowns	# of iteration	Time (sec)	Percentage of unknown pixels
Fig. 9 (a)	100×100	8496	213	6.04	84.96%
Fig. 9 (b)	300×235	42114	137	26.55	59.74%
Fig. 10 (a)	100×100	6996	105	2.88	69.96%
Fig. 10 (b)	484×404	14258	197	103.40	7.29%



(a)



(b)

Figure 9: Euler's elastica based image inpainting with $t=1$ in (6.6). We set $a=10$, $b=20$, $\mu=1000$, $r_1=1$ and $r_2=50$, $r_3=20$, $r_4=30$ for (a); $r_2=50$, $r_3=10$, $r_4=30$ for (b). The tolerance is $1.8\cdot 10^{-3}$ for (a) and 9×10^{-4} for (b).

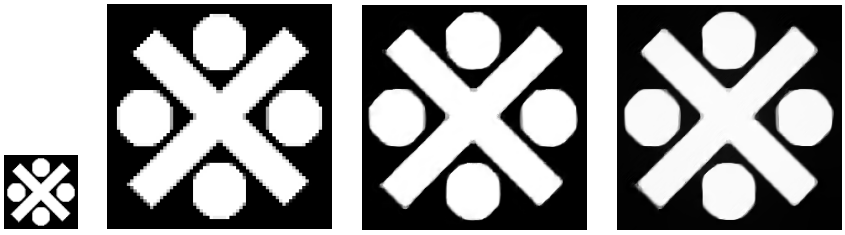
(a)

(b)

Figure 10: Euler's elastica based image inpainting with $t=2$ in (6.6). We set $a=5$, $b=20$, $\mu=1000$ and $r_1=1$, $r_2=20$, $r_3=5$ for (a); $r_1=1$, $r_2=1$, $r_3=20$, $r_4=2$ for (b). The tolerance is $4\cdot 10^{-3}$ for (a) and 1.5×10^{-4} for (b).

6.3. Image zooming

Image zooming is an interpolation problem of changing a given image into a higher or lower resolution image. Suppose the initial image u_0 of size $[1,M_1]\times[1,M_2]$ is enhanced by the factor $r\in N$, which means that the domain is interpolated onto the domain $\Omega=$



(a) (b) (c) (d)

Figure 11: Euler's elastica based image zooming. (b), (c) and (d) are resized by a factor of 4 from (a). (b) is obtained by the linearization interpolation. (c) and (d) are obtained by Euler's elastica with L^1 and L^2 data fitting term, respectively. We set $r_1 = 30$, $r_2 = 10$, $r_3 = 1$, $r_4 = 10$ for (c) and $r_1 = 30$, $r_2 = 10$, $r_3 = 5$ (d). The tolerance is $3.5 \cdot 10^{-3}$ for (c) and 4×10^{-3} for (d).

$[1, r \times M_1] \times [1, r \times M_2]$. The interpolation methods for image zooming problems are studied and discussed in [18, 23, 25, 35]. By using Euler's elastica energy, we solve the following minimization problem

$$\min_u \int_\Omega \Big(a + b\Big(\nabla \cdot \frac{\nabla u}{|\nabla u|}\Big)^2\Big)|\nabla u| + \frac{\overline{\mu}}{t}\int_\Omega |u - \overline{u}_0|^t, \tag{6.8}$$

where $\overline{\mu}$ is defined as (6.7) and $\overline{u}_0$ is obtained by

$$\overline{u}_0(i,j) = \begin{cases} u_0\Big(\frac{i}{r}, \frac{j}{r}\Big), & \text{if } (i,j) \in \Omega \backslash D, \\ 0, & \text{if } (i,j) \in D. \end{cases} \tag{6.9}$$

The values in zooming domain D is interpolated via Euler's elastica energy. Similarly, the energy functional (6.8) with both $t = 1$ and $t = 2$ are tested in the experiment. We choose the input image with size 64×64 and enlarge the image by a factor of 4. In Fig. 11, we show the zooming result of the linear interpolation in (b) and results of the proposed algorithm with L^1 and L^2 data fitting term in (c) and (d), respectively. We can see the superiority of Euler's elastica in preserving corners and shapes for image zooming. For this experiment, we set $a = 2$, $b = 20$, $\mu = 100$ for both cases with $t = 1$ and $t = 2$ and we have

- $t = 1$: Iteration: 300 and Time: 57.24s.
- $t = 2$: Iteration: 299 and Time: 56.71s.

7. Conclusions

In this work, we proposed a simple and efficient augmented Lagrangian approach for Euler's elastica. The operator splitting method and decomposition technique are used to solve the saddle-point problem arising in the augmented Lagrangian formulation. We applied the fixed-point method to one subproblem to get a closed-form solution. All the

subproblems in our method can be solved efficiently by fast iterative methods. As the numerical results of image denoising, image inpainting and image zooming demonstrated, our method yields better results in terms of computational time than the algorithm in [33] in the same error setting.

Acknowledgments We would like to thank Dr. Wei Zhu (University of Alabama) for sharing his Matlab code in order to do the comparison experiments. The authors are also indebted to Dr. Weimin Huang (Institute for Infocomm Research) and Dr. Huibin Chang (Tianjin Normal University) for their useful suggestion and valuable discussion.

References

[1] L. Ambrosio and S. Masnou, *A direct variational approach to a problem arising in image reconstruction*, Interfaces and Free Boundaries, 5(1) (2003), pp. 63–82.
[2] L. Ambrosio and S. Masnou, *On a variational problem arising in image reconstruction*, Free Boundary Problems, 147(1) (2005), pp. 17–26.
[3] E. Bae, J. Shi and X. C. Tai, *Graph cuts for curvature based image denoising*, IEEE Trans. Image Proc., 20(5) (2011), pp. 1199–1210.
[4] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro and J. Verdera, *Filling-in by joint interpolation of vector fields and gray levels*, IEEE Trans. Image Proc., 10(8) (2002), pp. 1200–1211.
[5] C. Ballester, V. Caselles and J. Verdera, *Disocclusion by joint interpolation of vector fields and gray levels*, Multiscale Model. Sim., 2(1) (2004), pp. 80–123.
[6] X. Bresson and T. F. Chan, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Problems Imaging, 2(4) (2008), pp. 455–484.
[7] C. Brito-Loeza and K. Chen, *Fast numerical algorithms for eulers elastica inpainting model*, Int. J. Modern Math., 5 (2010), pp. 157–182.
[8] C. Brito-Loeza and K. Chen, *Multigrid algorithm for high order denoising*, SIAM J. Imaging Sci., 3(3) (2010), pp. 363–389.
[9] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vision, 20(1) (2004), pp. 89–97.
[10] A. Chambolle and P. L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76(2) (1997), pp. 167–188.
[11] T. Chan, G. Golub and P. Mulet, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM J. Sci. Comput., 20(6) (1999), pp. 1964–1977.
[12] T. Chan, A. Marquina and P. Mulet, *High-order total variation-based image restoration*, SIAM J. Sci. Comput., 22(2) (2000), pp. 503–516.
[13] T. F. Chan, S. H. Kang and J. Shen, *Total variation denoising and enhancement of color images based on the CB and HSV color models*, J. Visual Commun. Image Representation, 12(4) (2001), pp. 422–435.
[14] T. F. Chan, S. H. Kang and J. Shen, *Euler's elastica and curvature-based inpainting*, SIAM J. Appl. Math., 63(2) (2002), pp. 564–592.
[15] M. Droske and A. Bertozzi, *Higher-order feature-preserving geometric regularization*, SIAM J. Imaging Sci., 3(1) (2010), pp. 21–51.
[16] S. Esedoglu and R. March, *Segmentation with depth but without detecting junctions*, J. Math. Imaging Vision, 18(1) (2003), pp. 7–15.
[17] S. Esedoglu, S. Ruuth and R. Tsai, *Threshold dynamics for shape reconstruction and disocclusion*, in IEEE Int. Conference Image Processing, 2 (2005), pp. 502–505.

[18] R. GAO, J. P. SONG AND X. C. TAI, *Image zooming algorithm based on partial differential equations technique*, Int. J. Numer. Anal. Modeling, 6(2) (2009), pp. 284–292.

[19] J. HAHN, G. J. CHUNG, Y. WANG AND X. C. TAI, *Fast algorithms for p-elastica energy with the application to image inpainting and curve reconstruction*, in SSVM'11: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision, pp. 169–182. Springer, 2011.

[20] J. HAHN, C. WU AND X. C. TAI, *Augmented lagrangian method for generalized TV-stokes model*, J. Sci. Computing, 50(2) (2012), pp. 235–264.

[21] N. KOMODAKIS AND N. PARAGIOS, *Beyond pairwise energies: efficient optimization for higher-order MRFs*, in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pages 2985–2992, 2009.

[22] R. LAI AND T. F. CHAN, *A framework for intrinsic image processing on surfaces*, Comput. Vision Image Understanding, 115(12) (2011), pp. 1647–1661.

[23] T. M. LEHMANN, C. GONNER AND K. SPITZER, *Survey: interpolation methods in medical image processing*, IEEE Trans. Medical Imaging, 18(11) (1999), pp. 1049–1075.

[24] 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 Processing, 12(12) (2003), pp. 1579–1590.

[25] E. MAELAND, *On the comparison of interpolation methods*, IEEE Trans. Medical Imaging, 7(3) (1988), pp. 213–217.

[26] S. MASNOU AND J. M. MOREL, *Level lines based disocclusion*, in IEEE Int. Conference Image Processing, pp. 259–263, 1998.

[27] S. MASNOU AND J. M. MOREL, *Level lines based disocclusion*, in Int. Conference Image Processing, pp. 259–263, 1998.

[28] D. MUMFORD AND CENTER FOR INTELLIGENT CONTROL SYSTEMS (US), *Elastica and computer vision*. Center for Intelligent Control Systems, Massachusetts Institute of Technology, 1991.

[29] M. NITZBERG, D. MUMFORD AND T. SHIOTA, *Filtering, segmentation and depth*, Lecture Notes Computer Sci., 1993.

[30] R. T. ROCKAFELLAR, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Math. Operations Research, 1(2) (1976), pp. 97–116.

[31] L. I. RUDIN, S. OSHER AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Phys. D Nonlinear Phenomena, 60(1-4) (1992), pp. 259–268.

[32] C. SCHÖNLIEB AND A. BERTOZZI, *Unconditionally stable schemes for higher order inpainting*, Commun. Math. Sci., 9(2) (2011), pp. 413–457.

[33] X. C. TAI, J. HAHN AND G. J. CHUNG, *A fast algorithm for Euler's elastica model using augmented lagrangian method*, SIAM J. Imaging Sci., 4(1) (2011), pp. 313–344.

[34] X. C. TAI AND C. WU, *Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model*, in *SSVM'09: Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision*, pages 502–513. Springer, 2009.

[35] M. UNSER, A. ALDROUBI AND M. EDEN, *Enlargement or reduction of digital images with minimum loss of information*, IEEE Trans. Image Processing, 4(3) (1995), pp. 247–258.

[36] 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 Sci., 3(3) (2010), pp. 300–339.

[37] C. WU, J. ZHANG, Y. DUAN AND X. C. TAI, *Augmented lagrangian method for total variation based image restoration and segmentation over triangulated surfaces*, J. Sci. Computing, 50(1) (2012), pp. 145–166.

[38] C. WU, J. ZHANG AND X. C. TAI, *Augmented Lagrangian method for total variation restoration with non-quadratic fidelity*, Inverse Problems Imaging, 5(1) (2011), pp. 237–261.