Proximal ADMM for Euler's Elastica Based Image Decomposition Model

Zhifang Liu[1], Samad Wali[1,*], Yuping Duan[2], Huibin Chang[3], Chunlin Wu[1] and Xue-Cheng Tai[4]

[1] *School of Mathematical Sciences, Nankai University, Tianjin, P.R. China.*
[2] *Center for Applied Mathematics, Tianjin University, Tianjin, China.*
[3] *School of Mathematical Sciences, Tianjin Normal University, Tianjin, P.R. China.*
[4] *Department of Mathematics, Hong Kong Baptist University, Hong Kong, China.*

Abstract. This paper studies image decomposition models which involve functional related to total variation and Euler's elastica energy. Such kind of variational models with first order and higher order derivatives have been widely used in image processing to accomplish advanced tasks. However, these non-linear partial differential equations usually take high computational cost by the gradient descent method. In this paper, we propose a proximal alternating direction method of multipliers (ADMM) for total variation (TV) based Vese-Osher's decomposition model [L.A. Vese and S.J. Osher, J. Sci. Comput., 19.1 (2003), pp. 553-572] and its extension with Euler's elastica regularization. We demonstrate that efficient and effective solutions to these minimization problems can be obtained by proximal based numerical algorithms. In numerical experiments, we present numerous results on image decomposition and image denoising, which conforms significant improvement of the proposed models over standard models.

AMS subject classifications: 68U10, 90C25, 49M37
Key words: Alternating direction method of multipliers, total variation, Euler's elastica, proximal method, image decomposition, image denoising.

1. Introduction

Assume that $\Omega \subset \mathbb{R}^2$ is a bounded, open, and connected subset (usually a rectangle in image processing). The image decomposition task is to decompose a given image $f : \Omega \to \mathbb{R}$ as the sum of two components

$$f = u + v,$$

*Corresponding author. *Email addresses:* liuzhifang0628@163.com (Zhifang Liu), samad.walikhan@gmail.com (Samad Wali), yuping.duan@tju.edu.cn (Yuping Duan), changhuibin@gmail.com (Huibin Chang), wucl@nankai.edu.cn (Chunlin Wu), xuechengtai@hkbu.edu.hk (Xue-Cheng Tai)

where u is geometric part or "cartoon" component, and v is an oscillating one. In general, u models homogeneous regions with sharp boundaries and v contains oscillating patterns such as texture and noise; See, e.g. [3, 15, 25, 26, 34, 39, 44, 45]. The origins of these ideas are the remarkable book by Y.Meyer [34], in which the author showed that the well-known Rudin-Osher-Fatemi (ROF) model [37] does not always represent texture or oscillatory details well.

An eligible and successful choice to detect textures is the generalized functions space $G = G(\Omega)$ [3, 30, 34, 46], where

$$G(\Omega) = \left\{ v = \nabla \cdot \mathbf{g} = \partial_x g_1 + \partial_y g_2 : \mathbf{g} = (g_1, g_2),\ g_1, g_2 \in L^\infty(\Omega, \mathbb{R}^2) \right\}$$

endowed with the norm

$$|v|_G = \inf\left\{ \| |\mathbf{g}| \|_{L^\infty(\Omega,\mathbb{R}^2)} : v = \nabla\cdot\mathbf{g},\ \mathbf{g} = (g_1, g_2),\ g_1, g_2 \in L^\infty(\Omega, \mathbb{R}^2), |\mathbf{g}| = \sqrt{g_1^2 + g_2^2} \right\}$$

The (BV, G) model proposed by Meyer in [34] is to solve the problem

$$\inf_{(u,v)\in BV(\Omega)\times G(\Omega)} \left\{ \int_\Omega |u|_{BV} + \beta |v|_G, f = u + v \right\}$$

where $BV(\Omega)$ is the *bounded variation* functions space.

There is no standard calculation of the associated Euler-Lagrange equation due to the term coming from an L^∞-norm (in the G-norm), which maps a series of work out to overcome this difficulty; See, e.g., [2, 4, 45, 46].

In [45, 46], Vese and Osher proposed to model oscillatory components v as first order derivatives of vector fields in $L^p, (1 \le p < \infty)$ (approaching to the L^∞-norm) to approximate Meyer's (BV, G) model. As the first practical image decomposition model, *total variation* (TV) based Vese-Osher's decomposition model solves the following convex minimization problem

$$\min_{u,\mathbf{g}} \left\{ \int_\Omega |\nabla u| + \frac{\alpha}{2} \int_\Omega |f - u - v|^2 + \beta \left(\int_\Omega |\mathbf{g}|^\rho \right)^{\frac{1}{\rho}} \right\} \tag{1.1}$$

where $\alpha,\ \beta > 0$, are tuning parameters, $v = \nabla \cdot \mathbf{g}$, $\mathbf{g} = (g_1, g_2)$ and $1 \le \rho < \infty$. The model (1.1) was solved by sequential descent approach to its Euler-Lagrange equation. It has been shown that the advantage of the model (1.1) is that it is not sensitive to the choice of ρ. The authors recommended to set $\rho = 1$ to yield faster calculations per iteration [45]. In fact if we generate the model (1.1) to the case $\rho = \infty$, then we can easily handle the L^∞-norm in the discrete setting by the ADMM method proposed in this paper.

In [12], the authors showed that TV regularization suffers from the undesirable staircase effect for image denoising application, which also exists in image decomposition problems. To overcome this, high order models have been proposed [13, 32, 49]. Euler's elastica is one of the higher order energy functionals, which has a number of interesting

applications in elasticity, computer graphics and in image processing. To improve the quality of an image in the sense of cartoon u and texture v and improve other applications like image denoising, we can use a non-convex and non-linear regularization such as curvature based regularization.

Euler's elastica energy, which was based on the curvature of the level curve of image, can effectively discriminate local behavior and global trends of the geometry hiding in certain image [14]. Euler's elastica energy was first introduced into computer vision by Mumford [35] and successfully applied to a number of applications, such as image restoration [1, 6], image segmentation [21, 33, 36, 52] and image inpainting [7, 9, 40].

In recent years, fast numerical optimization methods, such as split Bregman method [27] and augmented Lagrangian method [38, 47, 48], have been widely used to solve variational methods arising from image processing. In [43], Tai, Hahn and Chung propose a fast and efficient algorithm for Euler's elastica model via augmented Lagrangian method. The algorithms for curvature based variational model, one can see [18, 42, 50, 51] for details.

1.1. Contribution

1. In this paper, we use the Euler's elastica as regularization for geometric component u and propose the following minimization problem

$$\min_{u,\mathbf{g}} \left\{ \int_\Omega \left(a + b \left(\nabla \cdot \frac{\nabla u}{|\nabla u|} \right)^2 \right) |\nabla u| + \frac{\alpha}{2} \int_\Omega |f - u - v|^2 + \beta \left(\int_\Omega |\mathbf{g}|^\rho \right)^{\frac{1}{\rho}} \right\} \tag{1.2}$$

   where $\alpha > 0$ and $\beta > 0$ are tuning parameters, $v = \nabla \cdot \mathbf{g}$, $\mathbf{g} = (g_1, g_2)$ and $\rho \geq 1$.

2. We propose alternating direction method of multipliers (ADMM), to solve the Vese-Osher's decomposition model (1.1). We show one subproblem can be easily solved with a closed form solution and the other subproblems can be transformed into a Fourier linear system and solved by block Gaussian elimination. Besides, we use the proximal method to further reduce the computational costs. Similarly, we implement same proximal ADMM for Euler's elastica model (1.2). Owing to the use of suitable proximal step and fixed-point technique, each subproblem can be solved easily. Numerical results illustrate the effectiveness and efficiency of the proposed method.

The paper is organized as follows. In the next section, we present the ADMM and proximal ADMM for Vese-Osher's decomposition model with total variation and Euler's elastica. In Sect. 3, we explain numerical discretization of the subproblems associated with ADMM and proximal ADMM, and give the convergence analysis in case of TV. In Sect. 4, we show some numerical experiments to illustrate the efficiency and effectiveness of our proposed algorithm. A concluding remark and future works are given in Sect. 5.

2. Proximal ADMM for Vese-Osher's decomposition model

In current paper, we study a variational decompositional model that is based on a prior model for plane curve, i.e. Euler's elastica. A remarkable feature of elasticas revealed by Mumford [35], which positively support the role of elasticas in image and vision analysis as an interpolation tool. It also shed light on the choice of "2" for curvature power in the Euler's elastica energy formula. In brief, our proposed elastica decomposition scheme combines both Vese-Osher's model and Euler's elastica energy. It thus provides a theoretical foundation for earlier works on PDE based image decomposition. In particular, we assume that our proposed fast proximal method for both TV and Euler's elastica based decomposition models will be feasible.

In this section, we propose a fast and efficient algorithm for minimization problems related to Vese-Osher's decomposition model. We assume that the image domain Ω is normally taken as a rectangle or a grid for a rectangle domain, i.e., $\Omega = [1, N] \times [1, M]$. For mathematical notation, we will use the standard inner product $\langle u, v\rangle = \int_\Omega u \cdot v$ and norm $\|u\| = \int_\Omega |u|^2$ in L^2 space. The adjoint operator of ∇ denote by ∇^*, i.e. $\nabla^* \mathbf{g} = -\nabla \cdot \mathbf{g}$.

2.1. ADMM for TV based Vese-Osher's decomposition model

In order to use ADMM, we should transfer the unconstrained minimization problem (1.1) to an equivalent constrained minimization problem by using an operator-splitting technique. We introduce two new variables $\mathbf{p} = \nabla u$ and $\mathbf{q} = \mathbf{g}$ and reformulate the problem (1.1) to be the following constrained minimization problem

$$\begin{aligned} &\min_{u,\mathbf{g}} \left\{ \int_\Omega |\mathbf{p}| + \frac{\alpha}{2} \int_\Omega |f - u - \nabla \cdot \mathbf{g}|^2 + \beta \left(\int_\Omega |\mathbf{q}|^\rho \right)^{\frac{1}{\rho}} \right\} \\ &\text{s.t.} \quad \mathbf{p} = \nabla u, \mathbf{q} = \mathbf{g}. \end{aligned} \tag{2.1}$$

For simplicity of presentation, we let $\mathbf{x} = \begin{pmatrix} u \\ \mathbf{g} \end{pmatrix}$, $\mathbf{y} = \begin{pmatrix} \mathbf{p} \\ \mathbf{q} \end{pmatrix}$ and $\mathcal{A} = \begin{pmatrix} \nabla & 0 \\ 0 & \mathcal{I} \end{pmatrix}$ with identity maps $\mathcal{I}$. The adjoint operator of $\mathcal{A}$ is given by $\mathcal{A}^* = \begin{pmatrix} \nabla^* & 0 \\ 0 & \mathcal{I} \end{pmatrix}$. It is straightforward to see that the minimization problem (2.1) is equivalent to the following minimization problem:

$$\begin{aligned} &\min_{\mathbf{x},\mathbf{y}} \{H(\mathbf{x}) + J(\mathbf{y})\} \\ &\text{s.t.} \quad \mathbf{y} = \mathcal{A}\mathbf{x}. \end{aligned} \tag{2.2}$$

where

$$H(\mathbf{x}) = \frac{\alpha}{2} \int_\Omega |f - u - \nabla \cdot \mathbf{g}|^2, \; J(\mathbf{y}) = \int_\Omega |\mathbf{p}| + \beta \left(\int_\Omega |\mathbf{q}|^\rho \right)^{\frac{1}{\rho}}.$$

We define the augmented Lagrangian function of (2.2) as

$$\mathcal{L}(\mathbf{x},\mathbf{y};\boldsymbol{\lambda}) = H(\mathbf{x}) + J(\mathbf{y}) + \langle \boldsymbol{\lambda}, \mathbf{y} - \mathcal{A}\mathbf{x}\rangle + \frac{\gamma}{2}\|\mathbf{y} - \mathcal{A}\mathbf{x}\|^2. \tag{2.3}$$

where $\boldsymbol{\lambda} = \begin{pmatrix} \lambda_{\mathbf{p}} \\ \lambda_{\mathbf{q}} \end{pmatrix}$ is the Lagrange multiplier and γ is a positive constant. Then the ADMM for solving (2.2) can be described in Algorithm 2.1.

Algorithm 2.1: ADMM for TV based decomposition model

Initialization: $\mathbf{x}^0, \mathbf{y}^0, \boldsymbol{\lambda}^0$;

while *stopping criteria not satisfied* **do**

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}^k; \boldsymbol{\lambda}^k), \tag{2.4}$$

$$\mathbf{y}^{k+1} \in \arg\min_{\mathbf{y}} \mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}; \boldsymbol{\lambda}^k), \tag{2.5}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \gamma(\mathbf{y}^{k+1} - \mathcal{A}\mathbf{x}^{k+1}). \tag{2.6}$$

end

The $\mathbf{x}-$sub problem (2.4) is a quadratic optimization problem,

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}^k; \boldsymbol{\lambda}^k) = \arg\min_{\mathbf{x}} \left\{ H(\mathbf{x}) - \langle \boldsymbol{\lambda}^k, \mathcal{A}\mathbf{x}\rangle + \frac{\gamma}{2}\|\mathbf{y}^k - \mathcal{A}\mathbf{x}\|^2 \right\},$$

whose optimality condition gives a linear equation

$$\nabla H(\mathbf{x}) - \mathcal{A}^*\boldsymbol{\lambda}^k + \gamma\mathcal{A}^*(\mathcal{A}\mathbf{x} - \mathbf{y}^k) = 0 \tag{2.7}$$

with the periodic boundary condition. Since we have

$$\nabla H(\mathbf{x}) = \begin{pmatrix} \nabla H_u(\mathbf{x}) \\ \nabla H_{\mathbf{g}}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \alpha(u + \nabla\cdot\mathbf{g} - f) \\ \alpha\nabla(f - u - \nabla\cdot\mathbf{g}) \end{pmatrix}, \tag{2.8}$$

the equation (2.7) can be written explicitly as

$$\begin{pmatrix} \alpha(u + \nabla\cdot\mathbf{g} - f) \\ \alpha\nabla(f - u - \nabla\cdot\mathbf{g}) \end{pmatrix} - \begin{pmatrix} \nabla^* & 0 \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \lambda_{\mathbf{p}}^k \\ \lambda_{\mathbf{q}}^k \end{pmatrix} + \gamma \begin{pmatrix} \nabla^* & 0 \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \nabla u - \mathbf{p}^k \\ \mathbf{g} - \mathbf{q}^k \end{pmatrix} = 0.$$

Note that $\Delta = \nabla^2 = \nabla\cdot\nabla = -\nabla^*\nabla$, we have

$$\begin{cases} \alpha(u + \nabla\cdot\mathbf{g} - f) + \nabla\cdot\lambda_{\mathbf{p}}^k + \gamma\nabla\cdot\mathbf{p}^k - \gamma\Delta u = 0, \\ \alpha\partial_x(f - u - \partial_x g_1 - \partial_y g_2) - \lambda_{q1}^k + \gamma(g_1 - q_1^k) = 0, \\ \alpha\partial_y(f - u - \partial_x g_1 - \partial_y g_2) - \lambda_{q2}^k + \gamma(g_2 - q_2^k) = 0. \end{cases} \tag{2.9}$$

In the linear system (2.9), taking the Fourier transform of both sides, we will have a Fourier linear system. With an FFT implementation, the solutions of the Fourier linear

system can be obtained via block Gaussian elimination in discrete setting, which will be showed in section 3.2.

For the $\mathbf{y}-$sub problem (2.5), it has a closed form solution. We can separate the minimization problem (2.5) into two minimization problems independently, that is,

$$\min_{\mathbf{p}} \left\{ \int_\Omega |\mathbf{p}| + \frac{\gamma}{2} \int_\Omega \left| \mathbf{p} - \nabla u^{k+1} + \frac{\lambda_{\mathbf{p}}^k}{\gamma} \right|^2 \right\} \tag{2.10}$$

and

$$\min_{\mathbf{q}} \left\{ \beta \left(\int_\Omega |\mathbf{q}|^\rho \right)^{\frac{1}{\rho}} + \frac{\gamma}{2} \int_\Omega \left| \mathbf{q} - \mathbf{g}^{k+1} + \frac{\lambda_{\mathbf{q}}^k}{\gamma} \right|^2 \right\}. \tag{2.11}$$

The minimizers of (2.10) and (2.11) ($\rho = 1$) can be obtained as follows:

$$\mathbf{p} = \max\left(0, 1 - \frac{1}{\gamma|\zeta|}\right)\zeta \text{ and } \mathbf{q} = \max\left(0, 1 - \frac{\beta}{\gamma|\eta|}\right)\eta,$$

where $\zeta = \nabla u^{k+1} - \dfrac{\lambda_{\mathbf{p}}^k}{\gamma}$ and $\eta = \mathbf{g}^{k+1} - \dfrac{\lambda_{\mathbf{q}}^k}{\gamma}$.

2.2. Proximal ADMM for TV based Vese-Osher's decomposition model

The Algorithm 2.1 is costly in practice due to the linear system (2.9). Thus we modify the (2.4) by taking a proximal step and propose the proximal ADMM scheme to solve (2.2). We will show that the solutions of each subproblem of proximal ADMM scheme can be obtained inexpensively.

Let

$$\mathcal{S} = \begin{pmatrix} \tau\mathcal{I} & \alpha\nabla^* \\ \alpha\nabla & \tau\mathcal{I} - \alpha\nabla\nabla^* \end{pmatrix}, \tag{2.12}$$

where $\tau > \alpha(\lambda_{\max}(\nabla\nabla^*) + \lambda_{\max}(\nabla^*) + \lambda_{\max}(\nabla))$. According to the definition (2.12), $\mathcal{S}$ is a self-adjoint positive semidefinite operator. We define the induced norm $\|\mathbf{x}\|_{\mathcal{S}} = \langle \mathcal{S}\mathbf{x}, \mathbf{x}\rangle^{1/2}$. Our proximal ADMM for solving (2.2) is described in Algorithm 2.2.

In fact, the difference between ADMM scheme (2.4)-(2.6) and proximal ADMM scheme (2.13)-(2.15) is only the $\mathbf{x}-$sub problem, where (2.13) can be solved more efficiently in computation owing to the proximal term $\frac{1}{2}\|\mathbf{x} - \mathbf{x}^k\|_{\mathcal{S}}^2$.

The $\mathbf{x}-$sub problem (2.13) is the following quadratic optimization problem,

$$\begin{aligned} \mathbf{x}^{k+1} &= \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}^k; \boldsymbol{\lambda}^k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}^k\|_{\mathcal{S}}^2 \\ &= \arg\min_{\mathbf{x}} \left\{ H(\mathbf{x}) - \langle \boldsymbol{\lambda}, \mathcal{A}\mathbf{x}\rangle + \frac{\gamma}{2}\|\mathbf{y}^k - \mathcal{A}\mathbf{x}\|^2 + \frac{1}{2}\|\mathbf{x} - \mathbf{x}^k\|_{\mathcal{S}}^2 \right\}, \end{aligned}$$

whose optimality condition gives a linear equation

$$\nabla H(\mathbf{x}) - \mathcal{A}^*\boldsymbol{\lambda} + \gamma\mathcal{A}^*(\mathcal{A}\mathbf{x} - \mathbf{y}^k) + \mathcal{S}(\mathbf{x} - \mathbf{x}^k) = 0, \tag{2.16}$$

Algorithm 2.2: Proximal ADMM for TV based decomposition model

Initialization: $\mathbf{x}^0, \mathbf{y}^0, \boldsymbol{\lambda}^0$;

while *stopping criteria not satisfied* **do**

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{y}^k; \boldsymbol{\lambda}^k) + \frac{1}{2}\|\mathbf{x} - \mathbf{x}^k\|_{\mathcal{S}}^2, \tag{2.13}$$

$$\mathbf{y}^{k+1} \in \arg\min_{\mathbf{y}} \mathcal{L}(\mathbf{x}^{k+1}, \mathbf{y}; \boldsymbol{\lambda}^k), \tag{2.14}$$

$$\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \gamma(\mathbf{y}^{k+1} - \mathcal{A}\mathbf{x}^{k+1}). \tag{2.15}$$

end

by the periodic boundary condition. Taking (2.8) and (2.12) into consideration, we obtain the explicit form of (2.16) as follow

$$\begin{pmatrix} \alpha(u + \operatorname{div}\mathbf{g} - f) \\ \alpha\nabla(f - u - \operatorname{div}\mathbf{g}) \end{pmatrix} - \begin{pmatrix} \nabla^* & 0 \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \lambda_{\mathbf{p}}^k \\ \lambda_{\mathbf{q}}^k \end{pmatrix} + \gamma \begin{pmatrix} \nabla^* & 0 \\ 0 & \mathcal{I} \end{pmatrix} \begin{pmatrix} \nabla u - \mathbf{p}^k \\ \mathbf{g} - \mathbf{q}^k \end{pmatrix} + \begin{pmatrix} \tau\mathcal{I} & \alpha\nabla^* \\ \alpha\nabla & \tau\mathcal{I} - \alpha\nabla\nabla^* \end{pmatrix} \begin{pmatrix} u - u^k \\ \mathbf{g} - \mathbf{g}^k \end{pmatrix} = 0,$$

i.e.

$$\begin{cases} \alpha(u + \operatorname{div}\mathbf{g}^k - f) + \operatorname{div}\lambda_{\mathbf{p}}^k + \gamma\operatorname{div}\mathbf{p}^k - \gamma\Delta u + \tau(u - u^k) = 0, \\ \alpha\nabla(f - u^k - \operatorname{div}\mathbf{g}^k) - \lambda_{\mathbf{q}}^k + \gamma(\mathbf{g} - \mathbf{q}^k) + \tau(\mathbf{g} - \mathbf{g}^k) = 0. \end{cases} \tag{2.17}$$

Because of the variables u and $\mathbf{g}$ are independent in the linear system (2.17), we can easily obtain the solution with less cost in discrete setting, which will be showed in section 3.3.

2.3. Proximal ADMM for Euler's elastica based Vese-Osher's decomposition model

The Euler's elastica energy based minimization problem (1.2) is composed of its total variation semi-norm and a curvature term. In fact, elastica is a combination of total variation suppressing oscillation in the gradient direction, and a curvature regularization term that penalizes non-smooth level set curves. Due to high non-linearity in elastica model, it is difficult to minimize and require high computational cost. To tackle the non linear constraint arising in the model, a proximal based approach is proposed so that all subproblems either is linear or has a closed-form solution. In this subsection, we propose a proximal ADMM for Euler's elastica based Vese-Osher's decomposition model. We first introduce four new variables $\mathbf{p}, \mathbf{q}, h$ and $\mathbf{n}$, then reformulate the problem

(1.2) to be the following equality constrained problem

$$\begin{aligned}&\min_{u,\mathbf{g},\mathbf{p},\mathbf{q},h,\mathbf{n}}\left\{\int_\Omega(a+bh^2)|\mathbf{p}|+\frac{\alpha}{2}\int_\Omega|f-u-\nabla\cdot\mathbf{g}|^2+\beta\left(\int_\Omega|\mathbf{q}|^\rho\right)^{\frac{1}{\rho}}\right\}\\&\text{s.t.}\quad\mathbf{p}=\nabla u,\mathbf{p}=|\mathbf{p}|\mathbf{n},h=\nabla\cdot\mathbf{n},\mathbf{q}=\mathbf{g}.\end{aligned}\tag{2.18}$$

For simplify of notations, we denote $\hat{\mathbf{x}}=\begin{pmatrix}u\\\mathbf{g}\\\mathbf{n}\end{pmatrix}$, $\hat{\mathbf{y}}=\begin{pmatrix}\mathbf{p}\\\mathbf{q}\\h\end{pmatrix}$ and $\hat{\mathcal{A}}=\begin{pmatrix}\nabla&0&0\\0&\mathcal{I}&0\\0&0&-\nabla^*\end{pmatrix}$. It follows that $\hat{\mathcal{A}}\hat{\mathbf{x}}=\begin{pmatrix}\nabla u\\\mathbf{g}\\\nabla\cdot\mathbf{n}\end{pmatrix}$ and $\hat{\mathcal{A}}^*=\begin{pmatrix}\nabla^*&0&0\\0&\mathcal{I}&0\\0&0&-\nabla\end{pmatrix}$. Then we rewrite (2.18) as

$$\begin{aligned}&\min_{\mathbf{x},\mathbf{y}}\left\{\hat{H}(\hat{\mathbf{x}})+\hat{J}(\hat{\mathbf{y}})\right\}\\&\text{s.t.}\quad\hat{\mathbf{y}}=\hat{\mathcal{A}}\hat{\mathbf{x}},\mathbf{p}=|\mathbf{p}|\mathbf{n}.\end{aligned}\tag{2.19}$$

where

$$\begin{aligned}\hat{H}(\hat{\mathbf{x}})&=\frac{\alpha}{2}\int_\Omega|f-u-\nabla\cdot\mathbf{g}|^2,\\\hat{J}(\hat{\mathbf{y}})&=\int_\Omega(a+bh^2)|\mathbf{p}|+\beta\left(\int_\Omega|\mathbf{q}|^\rho\right)^{\frac{1}{\rho}}.\end{aligned}$$

The augmented Lagrangian functional of (2.19) is defined as

$$\begin{aligned}\hat{\mathcal{L}}(\hat{\mathbf{x}},\hat{\mathbf{y}};\hat{\boldsymbol{\lambda}},\lambda_{\mathbf{n}})=\hat{H}(\hat{\mathbf{x}})+\hat{J}(\hat{\mathbf{y}})+\langle\hat{\boldsymbol{\lambda}},\hat{\mathbf{y}}-\hat{\mathcal{A}}\hat{\mathbf{x}}\rangle+\frac{\hat{\gamma}}{2}\|\hat{\mathbf{y}}-\hat{\mathcal{A}}\hat{\mathbf{x}}\|^2\\+\langle\lambda_{\mathbf{n}},\mathbf{p}-|\mathbf{p}|\mathbf{n}\rangle+\frac{\gamma_{\mathbf{n}}}{2}\|\mathbf{p}-|\mathbf{p}|\mathbf{n}\|^2,\end{aligned}\tag{2.20}$$

where $\hat{\boldsymbol{\lambda}}=\begin{pmatrix}\lambda_{\mathbf{p}}\\\lambda_{\mathbf{q}}\\\lambda_h\end{pmatrix}$, $\lambda_{\mathbf{n}}$ are the Lagrange multipliers and $\hat{\gamma}$, $\gamma_{\mathbf{n}}$ are the positive constants. Our proximal ADMM for solving (1.2) is described in Algorithm 2.3.

In the Algorithm 2.3, we let $\hat{\mathcal{S}}$ is a self-adjoint positive semidefinite operator, which is defined as

$$\hat{\mathcal{S}}=\begin{pmatrix}\tau_1\mathcal{I}&\alpha\nabla^*&0\\\alpha\nabla&\tau_1\mathcal{I}-\alpha\nabla\nabla^*&0\\0&0&\tau_2\mathcal{I}-\gamma\nabla\nabla^*\end{pmatrix},\tag{2.24}$$

where $\tau_1>\alpha(\lambda_{\max}(\nabla\nabla^*)+\lambda_{\max}(\nabla^*)+\lambda_{\max}(\nabla))$ and $\tau_2>\gamma\lambda_{\max}(\nabla\nabla^*)$.

Algorithm 2.3: Proximal ADMM for Euler's elastica based decomposition model

Initialization: $\mathbf{x}^0, \mathbf{y}^0, \boldsymbol{\lambda}^0, \lambda_{\mathbf{n}}^0$;

while *stopping criteria not satisfied* **do**

$$\hat{\mathbf{x}}^{k+1} = \arg\min_{\hat{\mathbf{x}}} \hat{\mathcal{L}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}^k; \hat{\boldsymbol{\lambda}}^k, \lambda_{\mathbf{n}}^k) + \frac{1}{2}\|\hat{\mathbf{x}} - \hat{\mathbf{x}}^k\|_{\hat{\mathcal{S}}}^2, \tag{2.21}$$

$$\hat{\mathbf{y}}^{k+1} \in \arg\min_{\hat{\mathbf{y}}} \hat{\mathcal{L}}(\hat{\mathbf{x}}^{k+1}, \hat{\mathbf{y}}; \hat{\boldsymbol{\lambda}}^k, \lambda_{\mathbf{n}}^k), \tag{2.22}$$

$$\begin{aligned} \hat{\boldsymbol{\lambda}}^{k+1} &= \hat{\boldsymbol{\lambda}}^k + \hat{\gamma}(\hat{\mathbf{y}}^{k+1} - \hat{\mathcal{A}}\hat{\mathbf{x}}^{k+1}), \\ \lambda_{\mathbf{n}}^{k+1} &= \lambda_{\mathbf{n}}^k + \gamma_{\mathbf{n}}(\mathbf{p}^{k+1} - |\mathbf{p}^{k+1}|\mathbf{n}^{k+1}). \end{aligned} \tag{2.23}$$

end

In the following, we show how to solve the subproblems in the each iteration of Algorithm 2.3. For the $\hat{\mathbf{x}}-$sub problem (2.21), it is a quadratic optimization problem,

$$\begin{aligned} \hat{\mathbf{x}}^{k+1} &= \arg\min_{\mathbf{x}} \hat{\mathcal{L}}(\hat{\mathbf{x}}, \hat{\mathbf{y}}^k; \hat{\boldsymbol{\lambda}}^k, \lambda_{\mathbf{n}}^k) + \frac{1}{2}\|\hat{\mathbf{x}} - \hat{\mathbf{x}}^k\|_{\hat{\mathcal{S}}}^2 \\ &= \arg\min_{\mathbf{x}} \Bigg\{ \hat{H}(\hat{\mathbf{x}}) - \langle \hat{\boldsymbol{\lambda}}^k, \hat{\mathcal{A}}\hat{\mathbf{x}} \rangle + \frac{\gamma}{2}\|\hat{\mathbf{y}}^k - \hat{\mathcal{A}}\hat{\mathbf{x}}\|^2 - \langle \lambda_{\mathbf{n}}^k, |\mathbf{p}^k|\mathbf{n} \rangle \\ &\qquad + \frac{\gamma_{\mathbf{n}}}{2}\|\mathbf{p}^k - |\mathbf{p}^k|\mathbf{n}\|^2 + \frac{1}{2}\|\hat{\mathbf{x}} - \hat{\mathbf{x}}^k\|_{\hat{\mathcal{S}}}^2 \Bigg\}, \end{aligned}$$

whose optimality condition gives a linear equation

$$\nabla\hat{H}(\hat{\mathbf{x}}) - \hat{\mathcal{A}}^*\hat{\boldsymbol{\lambda}} + \gamma\hat{\mathcal{A}}^*(\hat{\mathcal{A}}\hat{\mathbf{x}} - \hat{\mathbf{y}}^k) - \begin{pmatrix} 0 \\ 0 \\ |\mathbf{p}^k|\lambda_{\mathbf{n}}^k - \gamma_{\mathbf{n}}|\mathbf{p}^k|(|\mathbf{p}^k|\mathbf{n} - \mathbf{p}^k) \end{pmatrix} + \hat{\mathcal{S}}(\hat{\mathbf{x}} - \hat{\mathbf{x}}^k) = 0 \tag{2.25}$$

by the periodic boundary condition. Combining (2.24) and using the fact that

$$\nabla\hat{H}(\mathbf{x}) = \begin{pmatrix} \nabla\hat{H}_u(\mathbf{x}) \\ \nabla\hat{H}_{\mathbf{g}}(\mathbf{x}) \\ \nabla\hat{H}_{\mathbf{n}}(\mathbf{x}) \end{pmatrix} = \begin{pmatrix} \alpha(u + \nabla\cdot\mathbf{g} - f) \\ \alpha\nabla(f - u - \nabla\cdot\mathbf{g}) \\ 0 \end{pmatrix},$$

we rewrite the equation (2.25) explicitly as

$$\begin{aligned} &\begin{pmatrix} \alpha(u + \nabla\cdot\mathbf{g} - f) \\ \alpha\nabla(f - u - \nabla\cdot\mathbf{g}) \\ 0 \end{pmatrix} + \begin{pmatrix} \nabla\cdot\lambda_{\mathbf{p}}^k \\ -\lambda_{\mathbf{q}}^k \\ \nabla\lambda_h^k \end{pmatrix} + \gamma \begin{pmatrix} -\Delta u + \nabla\cdot\mathbf{p}^k \\ \mathbf{g} - \mathbf{q}^k \\ \nabla(h^k - \nabla\cdot\mathbf{n}) \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \vec{z} \end{pmatrix} \\ &\qquad + \begin{pmatrix} \tau_1\mathcal{I} & \alpha\nabla^* & 0 \\ \alpha\nabla & \tau_1\mathcal{I} - \alpha\nabla\nabla^* & 0 \\ 0 & 0 & \tau_2\mathcal{I} - \gamma\nabla\nabla^* \end{pmatrix} \begin{pmatrix} u - u^k \\ \mathbf{g} - \mathbf{g}^k \\ \mathbf{n} - \mathbf{n}^k \end{pmatrix} = 0, \end{aligned}$$

where $\vec{z} = |\mathbf{p}^k|\lambda_{\mathbf{n}}^k - \gamma_{\mathbf{n}}|\mathbf{p}^k|(|\mathbf{p}^k|\mathbf{n} - \mathbf{p}^k)$. Thus, we have

$$\begin{cases} \alpha(u + \nabla\cdot\mathbf{g}^k - f) + \nabla\cdot\lambda_{\mathbf{p}}^k + \gamma\nabla\cdot\mathbf{p}^k - \gamma\Delta u + \tau_1(u - u^k) = 0, & (2.26a) \\ \alpha\nabla(f - u^k - \nabla\cdot\mathbf{g}^k) - \lambda_{\mathbf{q}}^k + \gamma(\mathbf{g} - \mathbf{q}^k) + \tau_1(\mathbf{g} - \mathbf{g}^k) = 0, & (2.26b) \\ \gamma\nabla(h^k - \nabla\cdot\mathbf{n}^k) + \nabla\lambda_h^k - |\mathbf{p}^k|\lambda_{\mathbf{n}}^k \\ \qquad +\gamma_{\mathbf{n}}|\mathbf{p}^k|(|\mathbf{p}^k|\mathbf{n} - \mathbf{p}^k) + \tau_2(\mathbf{n} - \mathbf{n}^k) = 0. & (2.26c) \end{cases}$$

Note that in the linear system (2.26), since the variables u, $\mathbf{g}$ and $\mathbf{n}$ are independent, we can get the solution from (2.26a), (2.26b) and (2.26c) easily.

For the $\hat{\mathbf{y}}-$sub problem (2.22), it is difficult to solve due to the non-differentiability element $|\mathbf{p}|$ in the quadratic term and the variables $\mathbf{p}$ and h are strong coupled together. Following [18, 43], we replace $\mathbf{p} = |\mathbf{p}|\mathbf{n}^{k+1}$ by $\mathbf{p} = |\mathbf{p}^k|\mathbf{n}^{k+1}$ in the quadratic penalty term in (2.22) with considering to utilize the fixed-point formulation to the nonlinear constraint $\mathbf{p} = |\mathbf{p}|\mathbf{n}$, and separate the minimization problem (2.22) into three minimization problems, i.e.

$$\min_{\mathbf{p}} \left\{ \int_\Omega c|\mathbf{p}| + \gamma \int_\Omega |\mathbf{p} - \zeta|^2 \right\}, \tag{2.27}$$

$$\min_{\mathbf{q}} \left\{ \beta \left(\int_\Omega |\mathbf{q}|^\rho \right)^{\frac{1}{\rho}} + \frac{\gamma}{2} \int_\Omega |\mathbf{q} - \eta|^2 \right\}, \tag{2.28}$$

and

$$\min_{h} \left\{ \int_\Omega (a + bh^2)|\mathbf{p}^k| + \int_\Omega \lambda_h \cdot h + \frac{\gamma}{2} \int_\Omega |h - \nabla\cdot\mathbf{n}^{k+1}|^2 \right\}, \tag{2.29}$$

where $c = a + b(h^k)^2 - \lambda_{\mathbf{p}}^k \cdot \mathbf{n}^{k+1}$, $\zeta = \dfrac{\gamma_{\mathbf{n}}|\mathbf{p}^k|\mathbf{n}^{k+1} + \gamma\nabla u^{k+1} - \lambda_{\mathbf{p}} - \lambda_{\mathbf{n}}}{\gamma + \gamma_{\mathbf{n}}}$ and $\eta = \mathbf{g}^{k+1} - \dfrac{\lambda_{\mathbf{q}}^k}{\gamma}$.

The minimization problems (2.27) and (2.28) ($\rho = 1$) have closed form solutions, which read

$$\mathbf{p} = \max\left(0, 1 - \frac{c}{(\gamma + \gamma_{\mathbf{n}})|\zeta|}\right)\zeta \text{ and } \mathbf{q} = \max\left(0, 1 - \frac{\beta}{\gamma|\eta|}\right)\eta.$$

When $\rho = \infty$, problem (2.28) can also be solved explicitly; See [23, Lemma 4.1, 4.2]. The optimality condition for the minimization problems (2.29) gives a linear equation

$$2b|\mathbf{p}^k|h + \lambda_h + \gamma(h - \operatorname{div}\mathbf{n}^{k+1}) = 0,$$

with a closed form solution

$$h = \frac{\gamma \operatorname{div}\mathbf{n}^{k+1} - \lambda_h}{2b|\mathbf{p}^k| + \gamma}.$$

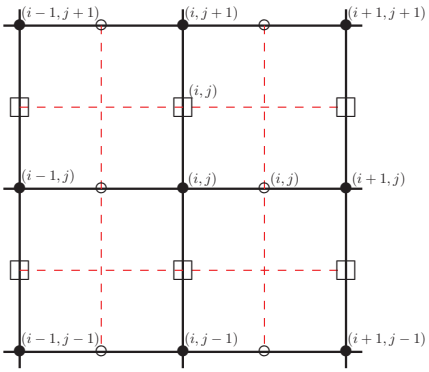


Figure 1: Grid definition. The rule of indexing variables in the augmented Lagrangian functional (2.20): u, h, λ_h are defined on $\bullet$-nodes. The first and second components of $\mathbf{g}, \mathbf{p}, \mathbf{q}, \mathbf{n}, \lambda_{\mathbf{p}}, \lambda_{\mathbf{q}}, \lambda_{\mathbf{n}}$ are defined on $\circ$-nodes and $\square$-nodes, respectively.

3. Numerical implementation and convergence analysis

In this section, we present the details of numerical algorithms for ADMM and proximal ADMM. In Section 3.1, we introduce some basic notations. The numerical implementations of Algorithm 2.1, Algorithm 2.2 and Algorithm 2.3 are described in Sect. 3.2, Sect. 3.3 and Sect. 3.4, respectively.

3.1. Notation

Without loss of generality, let $\Omega = [1, N] \times [1, M]$ be a set of $N \times M$ points in $\mathbb{R}^2$. For the convenience of description, we denote V as the Euclidean space $\mathbb{R}^{N\times M}$ and $Q = V \times V$. Then for a given $(i, j) \in [1, N] \times [1, M]$, we denote $u(i, j) \in \mathbb{R}$ and $\mathbf{p}(i, j) = (\mathbf{p}_1(i, j), \mathbf{p}_2(i, j)) \in \mathbb{R}^2$ for $u \in V$ and $\mathbf{p} \in Q$, respectively. We equip the standard Euclidean inner products as follows:

$$\langle u, v\rangle_V \equiv \sum_{i,j} u(i, j)v(i, j) \text{ and } \langle \mathbf{p}, \mathbf{q}\rangle_Q \equiv \langle p_1, q_1\rangle_V + \langle p_2, q_2\rangle_V.$$

Note that the induced norm $\|\cdot\|_V$ is the ℓ_2- norm on the vector spaces V and Q. In addition, we mention that

$$|\mathbf{p}(i, j)| = |(p_1(i, j), p_2(i, j))| = \sqrt{(p_1(i, j))^2 + (p_2(i, j))^2},$$

the usual Euclidean norm in $\mathbb{R}^2$.

The discrete backward and forward differential operators for $u \in V$ 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 gradient operator $\nabla : V \to Q$ is given as follows: For $u \in V$,

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

Considering inner products on V and Q, the discrete adjoint operator $\mathrm{div} : Q \to V$ of $-\nabla$ is given by

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

When a variable defined on $\circ$-nodes (or $\square$-nodes, see Fig. 1) needs to be evaluated at $(i,j) \in \square$-nodes (or $\circ$-nodes), for example $\mathbf{p} = (p_1, p_2)$, we use the average operators:

$$A_{i,j}^{\square}(p_1) = \frac{p_1(i,j+1) + p_1(i-1,j+1) + p_1(i,j) + p_1(i-1,j)}{4},$$

$$A_{i,j}^{\circ}(p_2) = \frac{p_2(i+1,j) + p_2(i,j) + p_2(i+1,j-1) + p_2(i,j-1)}{4},$$

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 $\mathbf{p}$ at $(i,j) \in \bullet$-nodes using the following operator:

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

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

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

3.2. Solve the subproblems in Algorithm 2.1

For the $\mathbf{x}-$sub problem (2.4) in Algorithm 2.1, we discretize (2.9) as follows:

$$\begin{cases} \alpha(u - \dfrac{\gamma}{\alpha}\Delta u + \partial_x^- g_1 + \partial_y^- g_2) = \alpha f - \mathrm{div}(\lambda_{\mathbf{p}}^k + \gamma \mathbf{p}^k), \\ \alpha(\partial_x^+ u + \partial_x^+ \partial_x^- g_1 - \dfrac{\gamma}{\alpha} g_1 + \partial_x^+ \partial_y^- g_2) = \alpha \partial_x^+ f - (\lambda_{q1}^k + \gamma q_1^k), \\ \alpha(\partial_y^+ u + \partial_y^+ \partial_x^- g_1 + \partial_y^+ \partial_y^- g_2 - \dfrac{\gamma}{\alpha} g_2) = \alpha \partial_y^+ f - (\lambda_{q2}^k + \gamma q_2^k). \end{cases} \tag{3.1}$$

Applying Fourier transform to both side of (3.1), we have

$$\mathbf{K}\mathcal{F}(\mathbf{x}) = \mathbf{F} - \mathbf{B}, \tag{3.2}$$

where

$$\mathbf{K} = \alpha \begin{pmatrix} \mathcal{I} - \frac{\gamma}{\alpha}\mathcal{F}(\Delta) & \mathcal{F}(\partial_x^-) & \mathcal{F}(\partial_y^-) \\ \mathcal{F}(\partial_x^+) & \mathcal{F}(\partial_x^+)\mathcal{F}(\partial_x^-) - \frac{\gamma}{\alpha}\mathcal{I} & \mathcal{F}(\partial_x^+)\mathcal{F}(\partial_y^-) \\ \mathcal{F}(\partial_y^+) & \mathcal{F}(\partial_y^+)\mathcal{F}(\partial_x^-) & \mathcal{F}(\partial_y^+)\mathcal{F}(\partial_y^-) - \frac{\gamma}{\alpha}\mathcal{I} \end{pmatrix},$$

$$\mathcal{F}(\mathbf{x}) = \begin{pmatrix} \mathcal{F}(u) \\ \mathcal{F}(g_1) \\ \mathcal{F}(g_2) \end{pmatrix}, \mathbf{F} = \alpha \begin{pmatrix} \mathcal{F}(f) \\ \mathcal{F}(\partial_x^+)\mathcal{F}(f) \\ \mathcal{F}(\partial_y^+)\mathcal{F}(f) \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} \mathcal{F}(\mathrm{div})\mathcal{F}(\lambda_{\mathbf{p}}^k + \gamma \mathbf{p}^k) \\ \mathcal{F}(\lambda_{q1}^k + \gamma q_1) \\ \mathcal{F}(\lambda_{q2}^k + \gamma q_2) \end{pmatrix}.$$

Clearly, the matrices $\mathcal{F}(\Delta)$, $\mathcal{F}(\partial_x^+)$, $\mathcal{F}(\partial_y^+)$, $\mathcal{F}(\partial_x^-)$, $\mathcal{F}(\partial_y^-)$, $\mathcal{F}(\mathrm{div})$, $\mathbf{K}$ and $\mathbf{F}$ only need to be computed once before iteration.

The $\mathbf{x}-$sub problem can be solved in three steps. Firstly, we apply discrete FFTs to both sides of (3.1). Secondly, we solve the resulting systems (3.2) by block Gaussian elimination for $\mathcal{F}(\mathbf{x})$. Finally, we apply $\mathcal{F}^{-1}$ to $\mathcal{F}(\mathbf{x})$ to obtain a new $\mathbf{x}$.

For the $\mathbf{y}-$sub problem (2.5), we have the following closed-form solutions:

$$\mathbf{p}(i,j) = \max\left(0, 1 - \frac{1}{\gamma|\zeta(i,j)|}\right)\zeta(i,j) \text{ and } \mathbf{q}(i,j) = \max\left(0, 1 - \frac{\beta}{\gamma|\eta(i,j)|}\right)\eta(i,j),$$

where

$$\zeta(i,j) = \nabla u^{k+1}(i,j) - \frac{\lambda_{\mathbf{p}}^k(i,j)}{\gamma},$$
$$\eta(i,j) = \mathbf{g}^{k+1}(i,j) - \frac{\lambda_{\mathbf{q}}^k(i,j)}{\gamma}.$$

At last, the update of the Lagrangian multiplier $\boldsymbol{\lambda} = (\lambda_{\mathbf{p}}, \lambda_{\mathbf{q}})$ is as follows:

$$\begin{aligned} \lambda_{p1}^{k+1} &= \lambda_{p1}^k + \gamma(p_1^{k+1} - \partial_x^+ u^{k+1}), \\ \lambda_{p2}^{k+1} &= \lambda_{p2}^k + \gamma(p_2^{k+1} - \partial_y^+ u^{k+1}), \\ \lambda_{q1}^{k+1} &= \lambda_{q1}^k + \gamma(q_1^{k+1} - g_1^{k+1}), \\ \lambda_{q2}^{k+1} &= \lambda_{q2}^k + \gamma(q_2^{k+1} - g_2^{k+1}). \end{aligned}$$

3.3. Solve the subproblems in Algorithm 2.2

For the $\mathbf{x}-$sub problem (2.13) in Algorithm 2.2, we have the following discrete form:

$$\begin{cases} (\alpha+\tau)u - \gamma\Delta u = \alpha f - \alpha \operatorname{div} \mathbf{g}^k - \operatorname{div} \lambda_{\mathbf{p}}^k - \gamma \operatorname{div} \mathbf{p}^k + \tau u^k, & \text{(3.3a)} \\ (\gamma+\tau)g_1 = \lambda_{q1}^k + \gamma q_1^k + \tau g_1^k - \alpha\partial_x^+(f - u^k - \operatorname{div} \mathbf{g}^k), & \text{(3.3b)} \\ (\gamma+\tau)g_2 = \lambda_{q2}^k + \gamma q_2^k + \tau g_2^k - \alpha\partial_y^+(f - u^k - \operatorname{div} \mathbf{g}^k). & \text{(3.3c)} \end{cases}$$

The linear system (3.3) can be efficiently solved. First, we use Fourier transform to solve the (3.3a),

$$u = \mathcal{F}^{-1}\left(\frac{\alpha\mathcal{F}(f) - \alpha\mathcal{F}(\mathrm{div})\mathcal{F}(\mathbf{g}^k) - \mathcal{F}(\mathrm{div})\mathcal{F}(\lambda_{\mathbf{p}}^k) - \gamma\mathcal{F}(\mathrm{div})\mathcal{F}(\mathbf{p}^k) + \tau\mathcal{F}(u^k)}{\alpha + \tau - \gamma\mathcal{F}(\Delta)}\right).$$

From (3.3b) and (3.3c), we have

$$g_1 = \frac{\lambda_{q1}^k + \gamma q_1^k + \tau g_1^k - \alpha \partial_x^+ (f - u^k - \operatorname{div} \mathbf{g}^k)}{\gamma + \tau},$$
$$g_2 = \frac{\lambda_{q2}^k + \gamma q_2^k + \tau g_2^k - \alpha \partial_y^+ (f - u^k - \operatorname{div} \mathbf{g}^k)}{\gamma + \tau}.$$

In Algorithm 2.2, the calculation $\mathbf{y}$- sub problem (2.14) and the update of the Lagrangian multiplier $\boldsymbol{\lambda} = (\lambda_{\mathbf{p}}, \lambda_{\mathbf{q}})$ at each pixel (i, j) are similar to Algorithm 2.1.

3.4. Solve the subproblems in Algorithm 2.3

The discretization scheme used in subsection 3.4 which has been inspired by various techniques in classical fluid dynamics, see in [40, 43]. It has a great advantage over conventional discretization scheme because this discretization scheme is more focus on locality of image, and focusing on locality is an important practice in numerical mathematics. In this computational scheme, we need to specify the half point values of the quantities $\mathbf{g}, \mathbf{p}, \mathbf{q}, \mathbf{n}, \lambda_{\mathbf{p}}, \lambda_{\mathbf{q}}, \lambda_{\mathbf{n}}$. For example for x-half-point, we need to take $(i+1/2, j)$ and for y-half-point, take $(i, j+1/2)$. Numerical experiments in Section 4 have shown the advantage of discretization scheme used for Euler's elastica based decomposition. In the next subsections, we briefly discuss about the solution of all other subproblems by using the staggered grid as shown Fig. 1.

3.4.1. The discretization for g-sub problem

From (2.26b), we have

$$\mathbf{g} = (g_1, g_2) = \frac{\lambda_{\mathbf{q}}^k + \gamma \mathbf{q}^k + \tau_1 \mathbf{g}^k - \alpha \nabla (f - u^k - \operatorname{div} \mathbf{g}^k)}{\gamma + \tau_1}.$$

Denote fixed variables $u^k, \mathbf{g}^k, \lambda_{\mathbf{q}}^k, \mathbf{q}^k$ by $u, \bar{\mathbf{g}}, \lambda_{\mathbf{q}}, \mathbf{q}$ respectively. Since for $\mathbf{g}$-sub problem, we use the proximal method to minimize the computational cost and here $\bar{\mathbf{g}}$ shows the value of $\mathbf{g}$ at k^{th} iteration. According to the rule of indexing variables in Fig. 1, the first and second component of $\mathbf{g}$ are defined on $\circ$-nodes and $\square$-nodes, respectively. The discretization of $\mathbf{g}$ at $(i, j) \in \circ$-nodes is obtained as follows:

$$g_1(i,j) = \frac{1}{\gamma + \tau_1} \Big(\lambda_{q1}(i,j) + \gamma q_1(i,j) + \tau_1 \bar{g}_1(i,j) - \alpha \partial_x^+ X(i,j) \Big),$$

where

$$X(i,j) = f(i,j) - u(i,j) - \operatorname{div}_{i,j}^{\bullet}(\bar{\mathbf{g}}).$$

Similarly, the discretization of $\mathbf{g}$ at $(i, j) \in \square$-nodes is obtained as follows:

$$g_2(i,j) = \frac{1}{\gamma + \tau_1} \Big(\lambda_{q2}(i,j) + \gamma q_2(i,j) + \tau_1 \bar{g}_2(i,j) - \alpha \partial_y^+ X(i,j) \Big).$$

3.4.2. The discretization for n-sub problem

From (2.26c), we have

$$(n_1, n_2) = \frac{|\mathbf{p}^k|\lambda_{\mathbf{n}}^k + \gamma_{\mathbf{n}}|\mathbf{p}^k|\mathbf{p}^k + \tau_2\mathbf{n}^k - \gamma\nabla(h^k - \operatorname{div}\mathbf{n}^k) - \nabla\lambda_h^k}{\gamma_{\mathbf{n}}|\mathbf{p}^k|^2 + \tau_2}.$$

Denote fixed variables $\mathbf{p}^k, \mathbf{n}^k, h^k, \lambda_{\mathbf{n}}^k, \lambda_h^k$ by $\mathbf{p}, \bar{\mathbf{n}}, h, \lambda_{\mathbf{n}}, \lambda_h$ respectively. Similarly since for $\mathbf{n}$-sub problem, we use the proximal method to minimize the computational cost and here $\bar{\mathbf{n}}$ shows the value of $\mathbf{n}$ at k^{th} iteration. According to the rule of indexing the dicretization of $\mathbf{n}$ at $(i,j) \in \circ$-nodes as follows:

$$n_1(i,j) = \frac{|\mathbf{p}_{i,j}|^\circ\lambda_{n1}(i,j) + \gamma_{\mathbf{n}}|\mathbf{p}_{i,j}|^\circ p_1(i,j) + \tau_2\bar{n}_1(i,j) - \gamma\partial_x^+ Y(i,j) - \partial_x^+\lambda_h(i,j)}{\gamma_{\mathbf{n}}\Big(|\mathbf{p}_{i,j}|^\circ\Big)^2 + \tau_2},$$

where $Y = h(i,j) - \operatorname{div}_{i,j}^{\bullet}(\bar{\mathbf{n}})$, and modulus of $\mathbf{p}$ at $(i,j) \in \circ$-nodes is defined as

$$|\mathbf{p}_{i,j}|^\circ = \sqrt{\big(p_1(i,j)\big)^2 + \big(A_{i,j}^{\circ}(p_2)\big)^2}.$$

Similarly the dicretization of $\mathbf{n}$ at $(i,j) \in \square$-nodes as follows:

$$n_2(i,j) = \frac{|\mathbf{p}_{i,j}|^\square\lambda_{n2}(i,j) + \gamma_{\mathbf{n}}|\mathbf{p}_{i,j}|^\square p_2(i,j) + \tau_2\bar{n}_2(i,j) - \gamma\partial_y^+ Y(i,j) - \partial_y^+\lambda_h(i,j)}{\gamma_{\mathbf{n}}\big(|\mathbf{p}_{i,j}|^\square\big)^2 + \tau_2},$$

where modulus of $\mathbf{p}$ at $(i,j) \in \square$-nodes is defined as

$$|\mathbf{p}_{i,j}|^\square = \sqrt{\big(A_{i,j}^{\square}(p_1)\big)^2 + \big(p_2(i,j)\big)^2}.$$

3.4.3. The discretization for p-sub and q-sub problems

For the $\mathbf{p}$- subproblem (2.27) and $\mathbf{q}$-sub problem (2.28), we have

$$\mathbf{p}(i,j) = \max\left(0, 1 - \frac{c(i,j)}{(\gamma_{\mathbf{n}} + \gamma)|\zeta(i,j)|}\right)\zeta(i,j) \text{ and } \mathbf{q}(i,j) = \max\left(0, 1 - \frac{\beta}{\gamma|\eta(i,j)|}\right)\eta(i,j).$$

where

$$c = a + b(h^k)^2 - \lambda_{\mathbf{p}}^k \cdot \mathbf{n}^{k+1},\ \zeta = \frac{\gamma_{\mathbf{n}}|\mathbf{p}^k|\mathbf{n}^{k+1} + \gamma\nabla u^{k+1} - \lambda_{\mathbf{p}}^k - \lambda_{\mathbf{n}}^k}{\gamma_{\mathbf{n}} + \gamma} \text{ and } \eta = \mathbf{g}^{k+1} - \frac{\lambda_{\mathbf{q}}^k}{\gamma}.$$

Denote fixed variables $u^{k+1}, \mathbf{p}^k, \mathbf{g}^{k+1}, \mathbf{n}^{k+1}, h^k, \lambda_{\mathbf{p}}^k, \lambda_{\mathbf{q}}^k, \lambda_{\mathbf{n}}^k$ by $u, \bar{\mathbf{p}}, \mathbf{g}, \mathbf{n}, h, \lambda_{\mathbf{p}}, \lambda_{\mathbf{q}}, \lambda_{\mathbf{n}}$ respectively. According to the rule of indexing for c, ζ and η at $(i,j) \in \circ$-nodes is obtained as follows:

$$c^1(i,j) = a + b\left(\frac{h(i+1,j) + h(i,j)}{2}\right)^2 - \lambda_{n1}(i,j)n_1(i,j) - A_{i,j}^{\circ}(\lambda_{n2})A_{i,j}^{\circ}(n_2),$$

$$(\gamma_{\mathbf{n}}+\gamma)\zeta_1^1(i,j)=\gamma\bigg(u(i+1,j)-u(i,j)\bigg)+\gamma_{\mathbf{n}}|\bar{\mathbf{p}}_{i,j}|^{\circ}n_1(i,j)-\lambda_{n1}(i,j)-\lambda_{p1}(i,j),$$

$$(\gamma_{\mathbf{n}}+\gamma)\zeta_2^1(i,j)=\frac{\gamma}{2}\bigg(\frac{u(i+1,j+1)+u(i,j+1)}{2}-\frac{u(i+1,j-1)+u(i,j-1)}{2}\bigg)$$
$$+\gamma_{\mathbf{n}}|\bar{\mathbf{p}}_{i,j}|^{\circ}A_{i,j}^{\circ}(n_2)-A_{i,j}^{\circ}(\lambda_{n2})-A_{i,j}^{\circ}(\lambda_{p2}).$$

$$\eta_1^1(i,j)=g_1(i,j)-\frac{1}{\gamma}\lambda_{q1}(i,j),$$

$$\eta_2^1(i,j)=A_{i,j}^{\circ}(g_2)-\frac{1}{\gamma}A_{i,j}^{\circ}(\lambda_{q2}).$$

Similarly, the discretization of c, ζ and η at $(i,j)\in\square$-nodes is obtained as follows:

$$c^2(i,j)=a+b\bigg(\frac{h(i,j+1)+h(i,j)}{2}\bigg)^2-A_{i,j}^{\square}(\lambda_{n1})A_{i,j}^{\square}(n_1)-\lambda_{n2}(i,j)n_2(i,j),$$

$$(\gamma_{\mathbf{n}}+\gamma)\zeta_1^2(i,j)=\frac{\gamma}{2}\bigg(\frac{u(i+1,j+1)+u(i+1,j)}{2}-\frac{u(i-1,j+1)+u(i-1,j)}{2}\bigg)$$
$$+\gamma_{\mathbf{n}}|\bar{\mathbf{p}}_{i,j}|^{\square}A_{i,j}^{\square}(n_1)-A_{i,j}^{\square}(\lambda_{n1})-A_{i,j}^{\square}(\lambda_{p1}),$$

$$(\gamma_{\mathbf{n}}+\gamma)\zeta_2^2(i,j)=\gamma\bigg(u(i,j+1)-u(i,j)\bigg)+\gamma_{\mathbf{n}}|\bar{\mathbf{p}}_{i,j}|^{\square}n_2(i,j)-\lambda_{n2}(i,j)-\lambda_{p2}(i,j).$$

$$\eta_1^2(i,j)=A_{i,j}^{\square}(g_1)-\frac{1}{\gamma}A_{i,j}^{\square}(\lambda_{q1}),$$

$$\eta_2^2(i,j)=g_2(i,j)-\frac{1}{\gamma}\lambda_{q2}(i,j),$$

Therefore, we have the discretization of (2.27) and (2.28):

$$p_d(i,j)=\max\left(0,1-\frac{c^d(i,j)}{(\gamma_{\mathbf{n}}+\gamma)\sqrt{(\zeta_1^d(i,j))^2+(\zeta_2^d(i,j))^2}}\right)\zeta_d^d(i,j),d\in\{1,2\},$$

and

$$q_d(i,j)=\max\left(0,1-\frac{\beta}{\gamma\sqrt{(\eta_1^d(i,j))^2+(\eta_2^d(i,j))^2}}\right)\eta_d^d(i,j),d\in\{1,2\}.$$

3.4.4. The discretization for h-sub problem

The optimality condition for the minimization problems (2.29) gives a linear equation

$$2b|p^k|h+\lambda_h+\gamma(h-\operatorname{div}\mathbf{n}^{k+1})=0,$$

with a closed form solution

$$h=\frac{\gamma_h\operatorname{div}_{i,j}^{\bullet}(\mathbf{n}^{k+1})-\lambda_h^k}{2b|A|_{i,j}^{\bullet}(\mathbf{p}^k)+\gamma}.$$

3.4.5. Update Lagrange multipliers

Finally, the update of the Lagrangian multipliers $\hat{\boldsymbol{\lambda}} = (\lambda_{\mathbf{p}}, \lambda_{\mathbf{q}}, \lambda_h)$ and $\lambda_{\mathbf{n}}$ according to the algorithm. Using the staggered grid as shown in Fig. 1, the discretized form for equations (2.23) is written as

$$\begin{aligned}
\lambda_{p1}^{k+1} &= \lambda_{p1}^{k} + \gamma(p_1^{k+1} - \partial_x^+ u^{k+1}) && \text{at } \circ\text{-nodes},\\
\lambda_{p2}^{k+1} &= \lambda_{p2}^{k} + \gamma(p_2^{k+1} - \partial_y^+ u^{k+1}) && \text{at } \square\text{-nodes},\\
\lambda_{q1}^{k+1} &= \lambda_{q1}^{k} + \gamma(q_1^{k+1} - g_1^{k+1}) && \text{at } \circ\text{-nodes},\\
\lambda_{q2}^{k+1} &= \lambda_{q2}^{k} + \gamma(q_2^{k+1} - g_2^{k+1}) && \text{at } \square\text{-nodes},\\
\lambda_{n1}^{k+1} &= \lambda_{n1}^{k} + \gamma_{\mathbf{n}}(p_1^{k+1} - |\mathbf{p}_{i,j}^{k+1}|^{\circ} n_1^{k+1}) && \text{at } \circ\text{-nodes},\\
\lambda_{n2}^{k+1} &= \lambda_{n2}^{k} + \gamma_{\mathbf{n}}(p_2^{k+1} - |\mathbf{p}_{i,j}^{k+1}|^{\square} n_2^{k+1}) && \text{at } \square\text{-nodes},\\
\lambda_{h}^{k+1} &= \lambda_{h}^{k} + \gamma(h^{k+1} - \mathrm{div}_{i,j}^{\bullet}(\mathbf{n}^{k+1})) && \text{at } \bullet\text{-nodes}.
\end{aligned}$$

3.5. Convergence analysis

The TV based Vese-Osher's texture model is a convex minimization problem. The convergence analysis of the ADMM scheme (2.4)-(2.6) and proximal ADMM scheme (2.13)-(2.15) have been widely studied, one can refer to [16, 17, 22, 24, 29].

For the sake of completeness, we give the convergence theorems for the ADMM scheme (2.4)-(2.6) and proximal ADMM scheme (2.13)-(2.15) in discrete setting but without proof, one can refer in [29] and [22, Appendix B] for more details.

Theorem 3.1. *We assume that* $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*)$ *is a KKT solution of* (2.2). *Let* $\{(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k)\}$ *be generated from the ADMM scheme* (2.4)-(2.6). *Then the sequence* $\{(\mathbf{x}^k, \mathbf{y}^k)\}$ *converges to an optimal solution to* (2.2) *and* $\{\boldsymbol{\lambda}^k\}$ *converges to an optimal solution to the dual problem of* (2.2), *i.e.* $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}^*$, $\lim_{k\to\infty} \mathbf{y}^k = \mathbf{y}^*$, $\lim_{k\to\infty} \boldsymbol{\lambda}^k = \boldsymbol{\lambda}^*$.

Since $H(\mathbf{x})$ is twice differentiable convex function with a Lipschitz continuous gradient, there exists a self-adjoint positive semidefinite linear operators Σ_H, such that for any $\mathbf{x}, \mathbf{x}'$,

$$\Sigma_H \preceq \mathcal{W},\ \forall\, \mathcal{W} \in \partial^2 H(\mathbf{x}),$$

where $\partial^2 H(\mathbf{x})$ is the Clarke's generalized Hessian at $\mathbf{x}$.

$$H(\mathbf{x}) \geq H(\mathbf{x}') + \langle \nabla H(\mathbf{x}'), \mathbf{x} - \mathbf{x}' \rangle + \frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|_{\Sigma_H}^2,$$

and

$$\langle \nabla H(\mathbf{x}) - \nabla H(\mathbf{x}'), \mathbf{x} - \mathbf{x}' \rangle \geq \|\mathbf{x} - \mathbf{x}'\|_{\Sigma_H}^2.$$

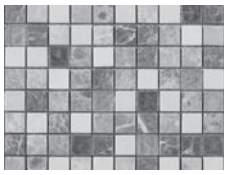
Theorem 3.2. *We assume that* $(\mathbf{x}^*, \mathbf{y}^*, \boldsymbol{\lambda}^*)$ *is a KKT solution of* (2.2) *and* $\Sigma_H + \mathcal{S} + \gamma\mathcal{A}^*\mathcal{A}$ *is positive definite. Let* $\{(\mathbf{x}^k, \mathbf{y}^k, \boldsymbol{\lambda}^k)\}$ *be generated from the proximal ADMM scheme* (2.13)-(2.15). *Then the sequence* $\{(\mathbf{x}^k, \mathbf{y}^k)\}$ *converges to an optimal solution to* (2.2) *and* $\{\boldsymbol{\lambda}^k\}$ *converges to an optimal solution to the dual problem of* (2.2), *i.e.* $\lim_{k\to\infty} \mathbf{x}^k = \mathbf{x}^*$, $\lim_{k\to\infty} \mathbf{y}^k = \mathbf{y}^*$, $\lim_{k\to\infty} \boldsymbol{\lambda}^k = \boldsymbol{\lambda}^*$.

(a) Barbara, size: 256×256

(b) Geometry, size: 256×256

(c) Fingerprint, size: 251×244

(d) Man, size: 256×256



(e) Floor, size: 300×400

(f) Flower, size: 250×250

(g) Wall, size: 210×236

Figure 2: Test Images.

Table 1: Numerical results of correlation Corr(u, v) for natural images

	Barbara	Geometry	Fingerprint	Man
CTV-L2 [20]	0.2581	0.1134	0.1462	0.2629
BLMV [10]	0.0364	0.0117	0.0875	0.0875
AD-aBLMV-ADE [41]	0.0222	**0.0055**	0.0192	**0.0492**
TV-L1 [31]	0.2775	0.0629	0.0589	0.1882
DF [11]	0.0758	0.02293	0.0539	0.1074
ADMM-TV	0.0386	0.0146	0.023	0.1125
PADMM-TV	0.0412	0.0148	0.0249	0.1125
PADMM-EE	**0.0155**	0.0058	**0.0167**	0.0751

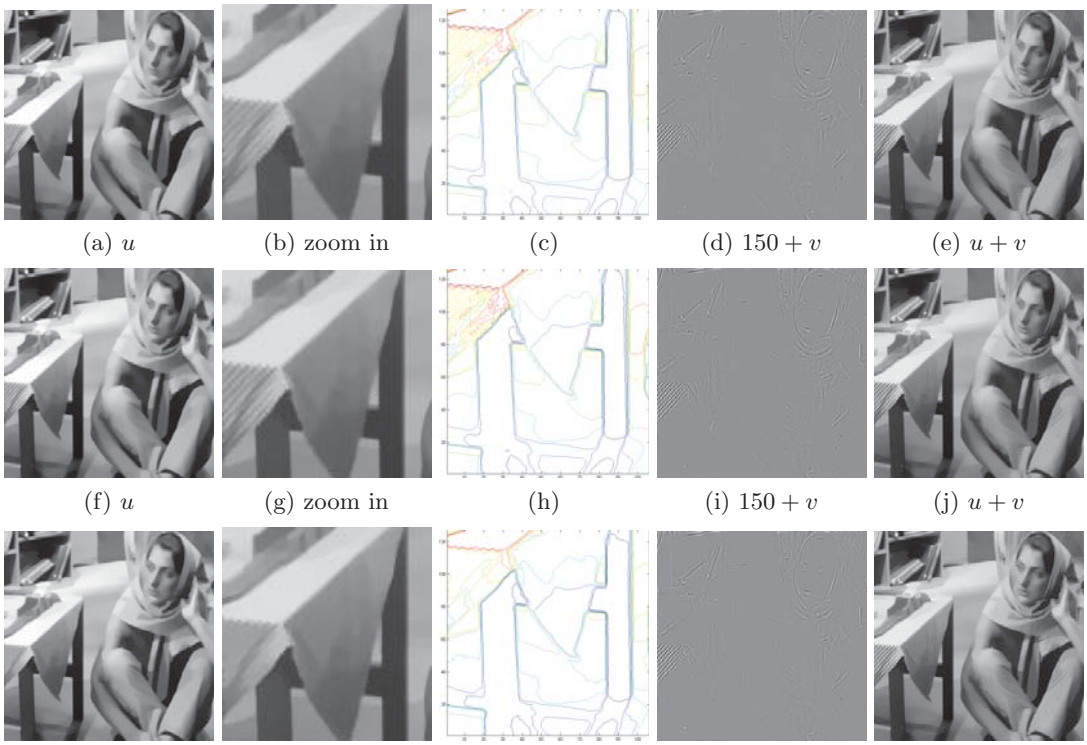
4. Numerical experiments

In this section, we provide some numerical examples. We will present some results of proposed methods for image decomposition and image denoising. All the experiments were performed using Windows 7 and MATLAB, R2013b (version 2.0) on a HP Z228 with Intel(R) Core(TM) i7-4790 CPU @3.60GHz and 8GB memory.

4.1. Test images, practical implementation and stopping condition.

In this paper, we tested lots of images, for example the Barbara, Geometry, etc. (see Fig. 2). For all of the tested algorithms, we used the following stopping condition:

$$\max\left(\frac{\|u^k - u^{k+1}\|}{\|f\|}, \frac{\|v^k - v^{k+1}\|}{\|f\|}\right) \leq \epsilon,$$



(a) u (b) zoom in (c) (d) $150 + v$ (e) $u + v$

(f) u (g) zoom in (h) (i) $150 + v$ (j) $u + v$

(k) u (l) zoom in (m) (n) $150 + v$ (o) $u + v$

Figure 3: Barbara decomposition comparison between TV and Euler's elastica based models. From top to button, the results are form ADMM-TV, PADMM-TV, PADMM-EE, respectively. We chose the parameters in the following way, for ADMM-TV we set $\alpha = 0.07, \beta = 0.8, \gamma = 0.35$. For PADMM-TV,$\alpha = 0.07, \beta = 0.8, \gamma = 0.35, \tau = 0.3$. For PADMM-EE, $\alpha = 0.09, \beta = 0.99, \gamma = 0.04, \gamma_h = 0.016, \gamma_{\mathbf{n}} = 0.01, \tau_1 = \tau_2 = 1.5, a = 1.1, b = 100$.

where $\epsilon > 0$ is stopping tolerance, in our proposed algorithms, we set $\epsilon = 1.0e - 3$. For simplicity of presentation, we denote our developed methods as follows:

- ADMM-TV: ADMM for TV based Vese-Osher's texture model;
- PADMM-TV: proximal ADMM for TV based Vese-Osher's decomposition model;
- PADMM-EE: proximal ADMM for Euler's elastica based Vese-Osher's decomposition model.

Before we show our numerical results, we give some remarks on choosing parameters and stopping criterion. There are three parameters (α, β, γ) coming from ADMM-TV, four parameters $(\alpha, \beta, \gamma, \tau)$ coming from PADMM-TV, and eight parameters $(\alpha, \beta, \gamma, \gamma_{\mathbf{n}}, \tau_1, \tau_2, a, b)$ coming from PADMM-EE. The parameter α is regularization parameter associated with the cartoon part of image. For decomposition case this

(a) $p = 1, \mathrm{Corr(u, v)} = 0.0128$

(b) $p = \infty, \mathrm{Corr(u, v)} = 0.0128$

Figure 4: Image decomposition on clean images by PADMM-EE with $p = 1$ and $p = \infty$

Table 2: Numerical results of correlation $\mathrm{Corr(u, v)}$ for synthetic images

	Fig. 9(a)	Fig. 9(b)	Fig. 9(c)	Fig. 9(d)	Fig. 9(e)
CTV-L2 [20]	0.2506	0.3724	0.2294	0.2195	0.1391
BLMV [10]	0.1086	0.2747	0.0593	0.0247	0.0148
AD-aBLMV-ADE [41]	0.0214	0.0194	0.0196	0.0086	0.0147
TV-L1 [31]	0.1469	0.18801	0.1733	0.1319	0.0697
DF [11]	0.1554	0.2901	0.0815	0.0268	0.0162
ADMM-TV	0.0278	0.0232	0.0356	0.0377	0.0260
PADMM-TV	0.0286	0.0229	0.0354	0.0273	0.0228
PADMM-EE	**0.0174**	**0.0134**	**0.0176**	**0.0053**	**0.0132**

Table 3: Computational time (in Seconds) of the different methods for Barbara image (256×256).

Computational time	
CTV-L2 [20]	3.39
BLMV [10]	0.23
AD-aBLMV-ADE [41]	5.694
TV-L1 [31]	0.44
ADMM-TV	0.7520
PADMM-TV	0.3444
PADMM-EE	1.1408

parameter is need to be small for better subtraction of cartoon from the image. For the pure cartoon-texture decomposition problem, we would like to remove the texture component without removing other key features such as edges, shading, etc. The pa-

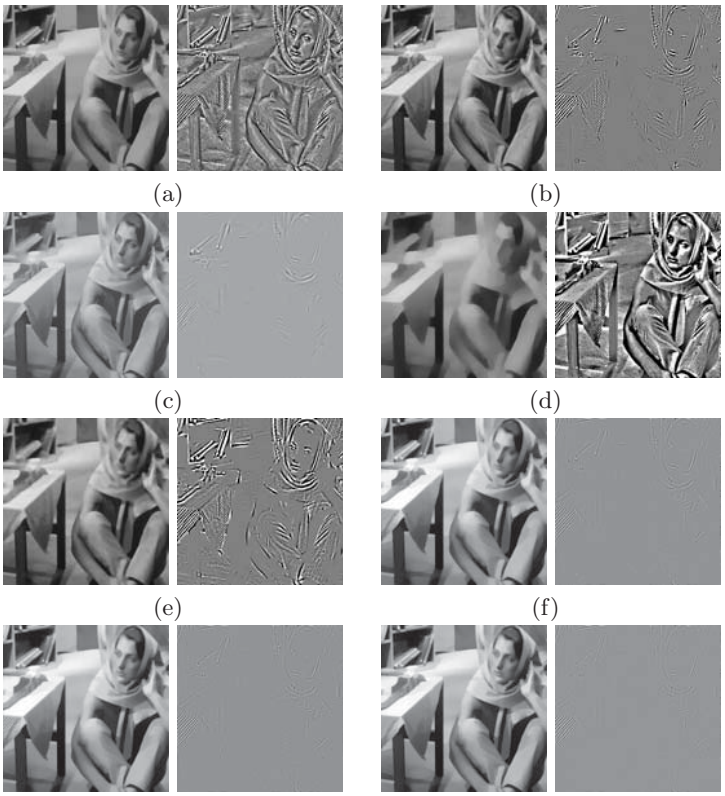


Figure 5: Separation of cartoon and texture components (Barbara). (a) CTV-L2 [20]. (b) BLMV [10]. (c) AD-aBLMV-ADE [41]. (d) TV-L1 [31]. (e) DF [11]. (f) ADMM-TV. (g) PADMM-TV. (h) Proposed PADMM-EE.

rameter β, which is the texture norm coefficient in the energy is related to the texture components of image. γ and $\gamma_{\mathbf{n}}$ are associated with Lagrange multipliers. All τ, τ_1 and τ_2 are the proximal parameters. These proximal parameters play an important role for the effectiveness of the proposed methods. a and b are Euler's elastica coefficients. The ratio a/b indicates the relative importance of the total length versus total squared curvature. For $b = 0$, the minimization problem (1.2) becomes TV problem (1.1). We used different setting for the parameters in different applications of our proposed methods. We tuned all the parameters by the above stopping criterion with tolerance.

4.2. Image decomposition

The main objective of decomposition model is to decompose an image into its cartoon and texture part. Various approaches to image decomposition have been proposed in the literature, among these approaches variational based image decomposition is one of the successful approaches and it has numerous applications in the field of image processing and computer vision. Interesting application of image decomposition are

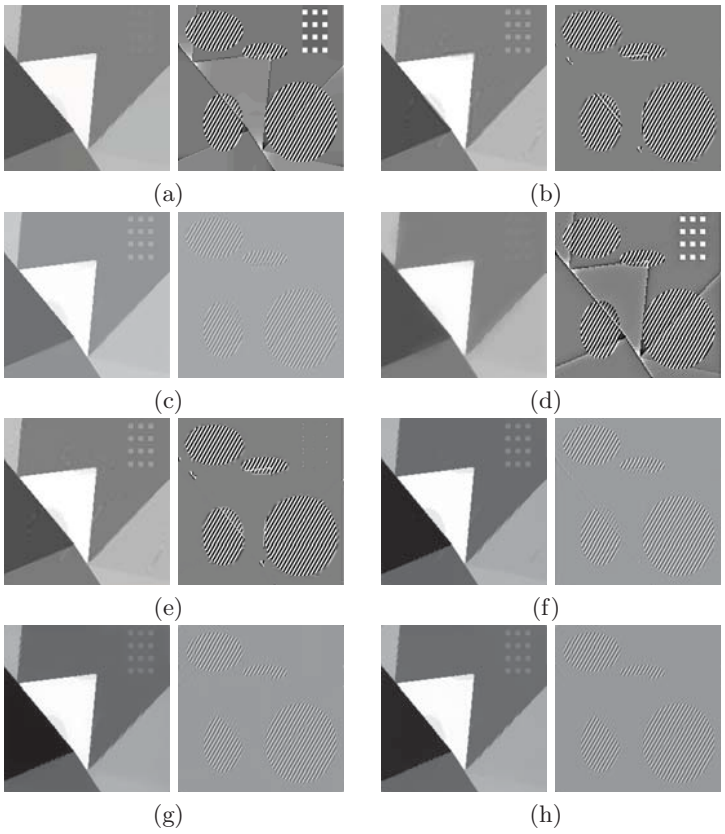


Figure 6: Separation of cartoon and texture components (Geometry). (a) CTV-L2 [20]. (b) BLMV [10]. (c) AD-aBLMV-ADE [41]. (d) TV-L1 [31]. (e) DF [11]. (f) ADMM-TV. (g) PADMM-TV. (h) Proposed PADMM-EE.

explored in [8, 45]. In this subsection, we illustrate the efficiency and effectiveness of the proposed algorithms via many examples in image decomposition.

As in the literature [5], we assume that the cartoon and texture parts in an image are not correlated. We thus take the correlation between the cartoon u and the texture v, which is computed by

$$Corr(u, v) = \frac{Cov(u, v)}{\sigma_u . \sigma_v}$$

to measure the quality of decomposition, where $\sigma_{(.)}$ and $\mathrm{Cov}(\cdot, \cdot)$ refer to the standard deviation and covariance of given variables, respectively.

For image decomposition, we do not have ground-truth generation in case of real images as well as synthetic images. Therefore we consider the best result for smallest correlation value. To evaluate the quality of different methods, we show the decomposition result for test images (see Fig. 2(a), 2(b), 2(c) and 2(d)). But we also evaluate numerically the competing methods on synthetic images (see Fig. 9). In our numerical experiments, we observed that the most of the decomposition results for $\rho = 1$ and $\rho = \infty$ in Euler's elastica decomposition model (1.2) are look similar (see Fig.

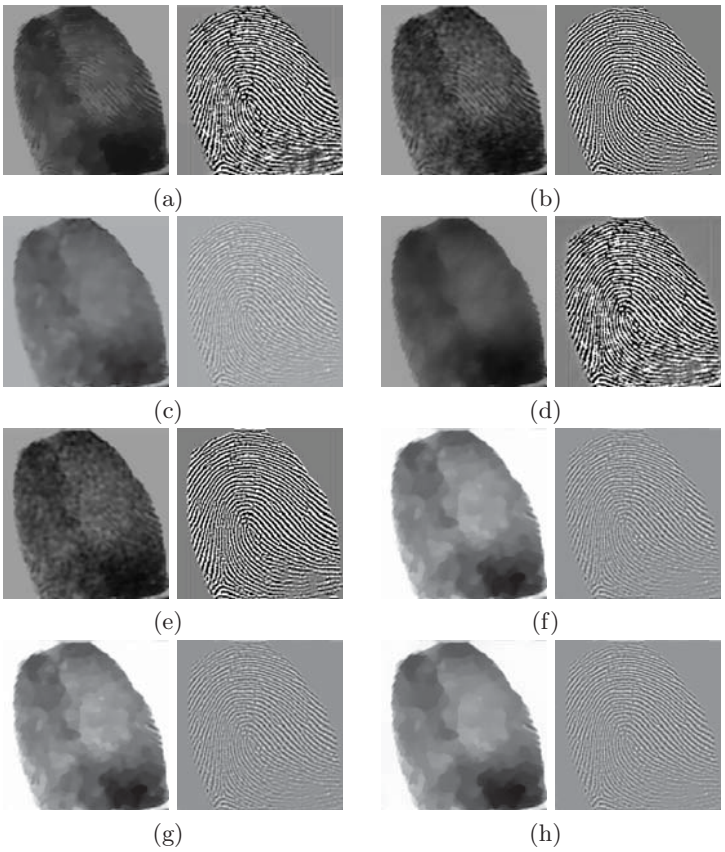


Figure 7: Separation of cartoon and texture components (Fingerprint). (a) CTV-L2 [20]. (b) BLMV [10]. (c) AD-aBLMV-ADE [41]. (d) TV-L1 [31]. (e) DF [11]. (f) ADMM-TV. (g) PADMM-TV. (h) Proposed PADMM-EE.

4). Therefore we decided to show and compare the results with $\rho = 1$ in all experiments. We compared our proposed PADMM-TV and PADMM-EE methods with the following decomposition methods: CTV-L2 [19, 20], AD-aBLMV-ADE [41], BLMV filters [10], TV-L1 [31], DF [11] and ADMM-TV. The codes for [41] was provided by author, for [20], [10], [31], [11], we used online demo provided on web by IPOL[†]. We used them with best tuned parameters in each individual test case.

The most commonly used image for texture evaluation is the Barbara image, which has precise edges and it also favors the TV-based methods, especially ADMM-TV and PADMM-TV (see Fig. 3). However, some artifacts can be seen on cartoon image of TV-based decomposition, as the cloth on the table has some texture in cartoon part (see zoom-in images in Fig. 3). PADMM-EE can completely eliminate the texture from the table cloth. Furthermore, it is obvious that we can get better contour by using cartoon part extracted by PADMM-EE algorithm (see third column Fig. 3). For numerical evaluation, we used the parameters that gave the minimum correlation

[†]Image Processing On Line. http://www.ipol.im/

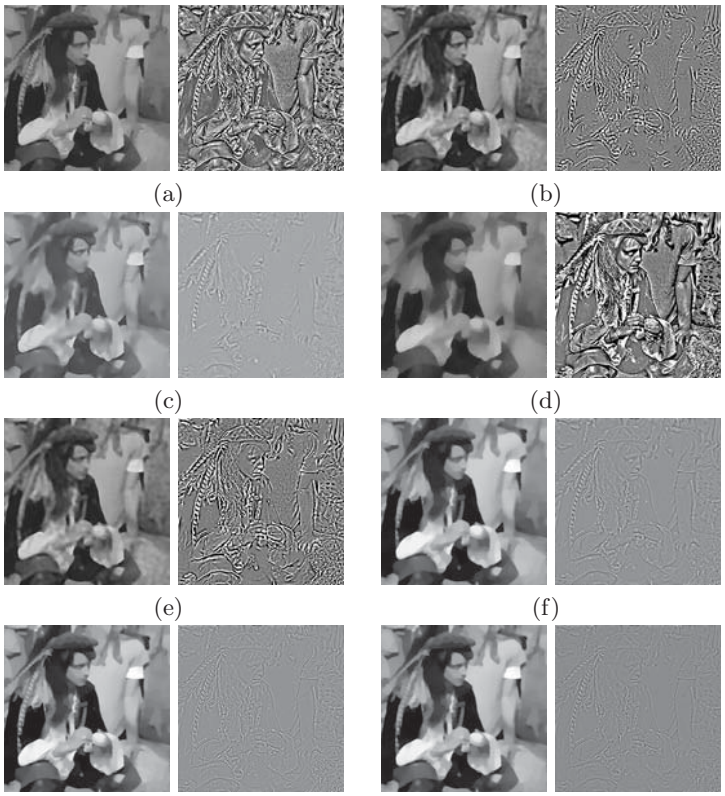


Figure 8: Separation of cartoon and texture components (Man). (a) CTV-L2 [20]. (b) BLMV [10]. (c) AD-aBLMV-ADE [41]. (d) TV-L1 [31]. (e) DF [11]. (f) ADMM-TV. (g) PADMM-TV. (h) Proposed PADMM-EE.

and also gave the best visual result. Every image has different kind of cartoon as well as texture, so the parameters need to be different, for example the parameters are simulated for the Barbara (see caption in Fig. 3).

To see how strong the cartoon edges on the texture image and precise the cartoon part, we used the Geometry image. All methods eliminate the texture part from cartoon, but methods [19, 20], [10], [31], [11], ADMM-TV and PADMM-TV bring some cartoon edges on the texture (see Fig. 6). For this image, adaptive image decomposition method [41] has best results, however Euler's elastica-based decomposition has better results among all TV-based decomposition methods and its results are very closed to adaptive decomposition method [41].

Most of decomposition methods like [10], [41] and [11] blur some parts of cartoon image (see Fig. 5,7,8), and adaptive decomposition [41] method also sightly blur the texture part. But on the other hand Euler's elastica-based decomposition has best results as compare to all TV-based methods [19], [31], ADMM-TV and PADMM-TV and as well as other methods.

We report the numerical results for real and artificial images in Table 1 and Ta-

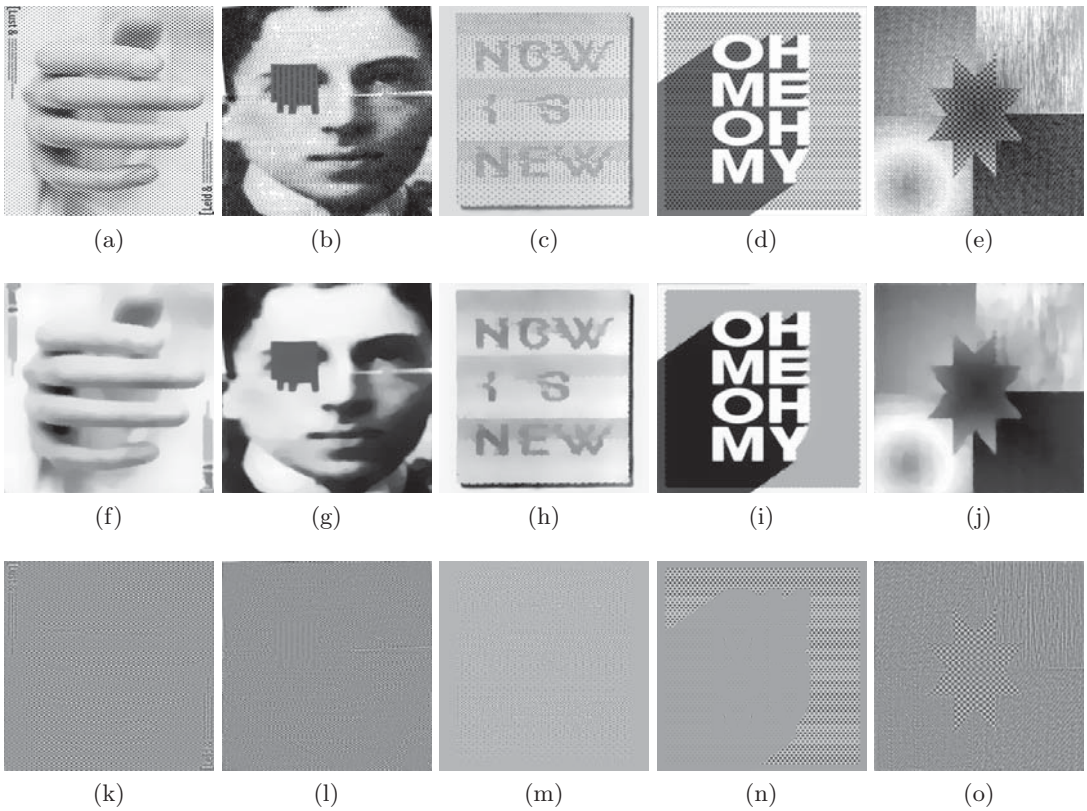


Figure 9: Synthetic images (256 ×256) used for numerical evaluation. Top row: original image. Middle row: cartoon component. Bottom row: texture component.

ble 2, where the obtained correlations (Corr) of the decomposed two parts are reported for all decomposition methods. The computational time for PADMM-TV is increased compared to the fast BLMV method, but it is faster then CTV-L2, AD-aBLMV-ADE, TV-L1, and ADMM-TV. Similarly, the computational time for PADMM-EE is increased compared to the BLMV, TV-L1, ADMM-TV and PADMM-TV methods, but it is faster than CTV-L2 and AD-aBLMV-ADE (see Table. 3).

Moreover, we also compared the decomposition results of generalized TV-stokes model [28] and our proposed Euler's elastica-based model. We used TVS-L^2 + $TV_{\mathbf{n}}$-L^2 models for image decomposition from the general TV-stokes model. The TV-stokes model is aimed to separate the jump discontinuities of an image, where u contains all the discontinuities of f and v is a continuous function (see Fig. 10). These separated parts could be used for a large number of image processing purpose. Note that the TV-stokes model do not separately obtain texture parts from an image. However, our proposed Euler's elastica-based model can better decompose an image into its texture parts (see also Fig. 10).

To show the convergence of our proposed method: PADMM-EE, we plot the numerical energy, relative error in u^{k+1}, correlation, residuals and relative error in lagrange

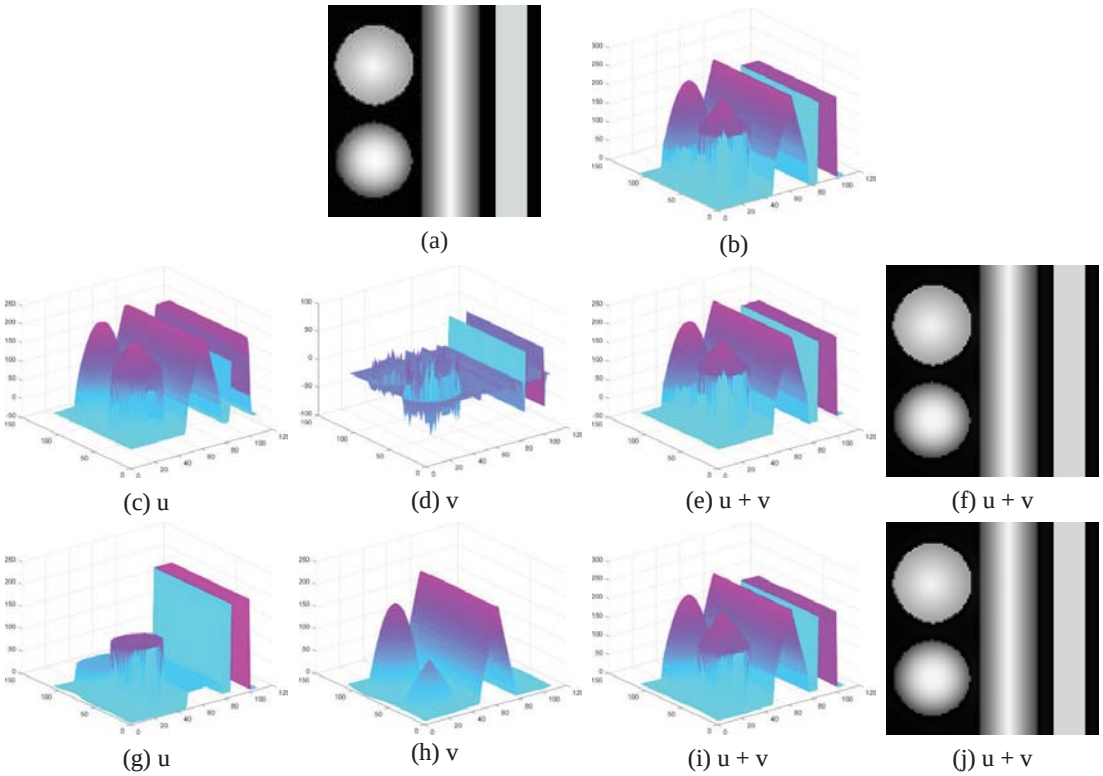


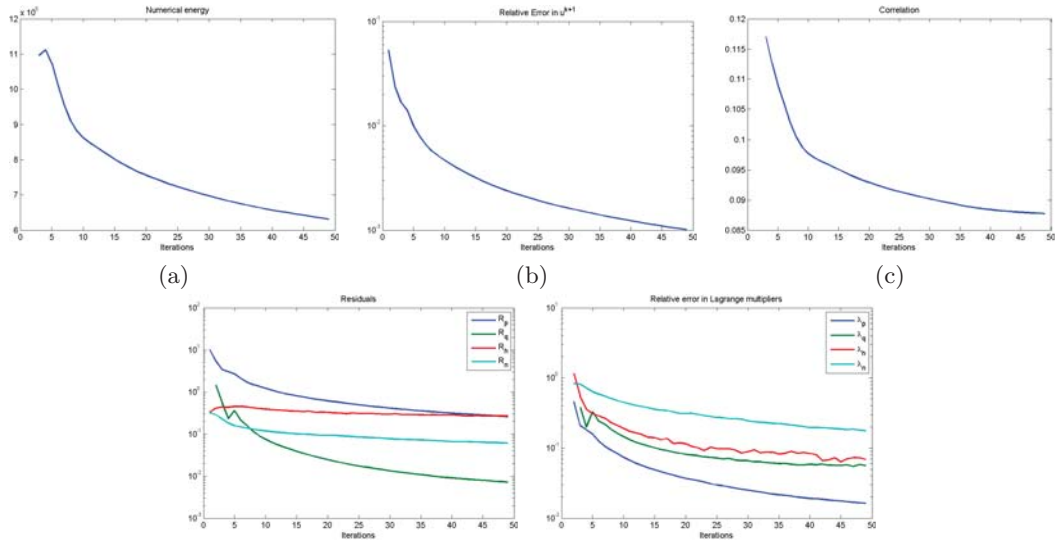
Figure 10: Synthetic images (108 ×108) compared between TV-Stokes [28] and Euler's elastica decomposition model. (a) is the original image, (b) is the surface plot of the original image. (c) to (f) are results of our proposed Euler's elastica model while (g) to (j) are results of TV-stokes model.

multipliers verses number of iterations for the image Man (see Fig. 11). From these plots, it is easy to see the proposed algorithm for Euler's elastica based decomposition has converged. The plots of residuals and relative error in lagrange multipliers also give important information about tuning of parameters and one can see in Fig. 11 that they converge to zero faster before 50 iterations.

4.3. Image denoising

Table 4: Number of iteration and CPU time for image denoising

Images	ADMM-TV		PADMM-TV		EEALM [43]		PADMM-EE	
	Nit	Time	Nit	Time	Nit	Time	Nit	Time
Floor	19	0.9507	26	0.5054	30	6.1270	28	2.0494
Flower	17	0.4484	23	0.2765	30	2.9618	26	1.1291
Wall	18	0.3712	24	0.2520	20	1.7888	29	1.0009



(a) (b) (c)

(d) (e)

Figure 11: Plots of numerical energy, relative error of u^{k+1}, correlation, residuals and relative error of Lagrange multipliers of our proposed Euler's elastica-based decomposition model for Fig. 2(d).

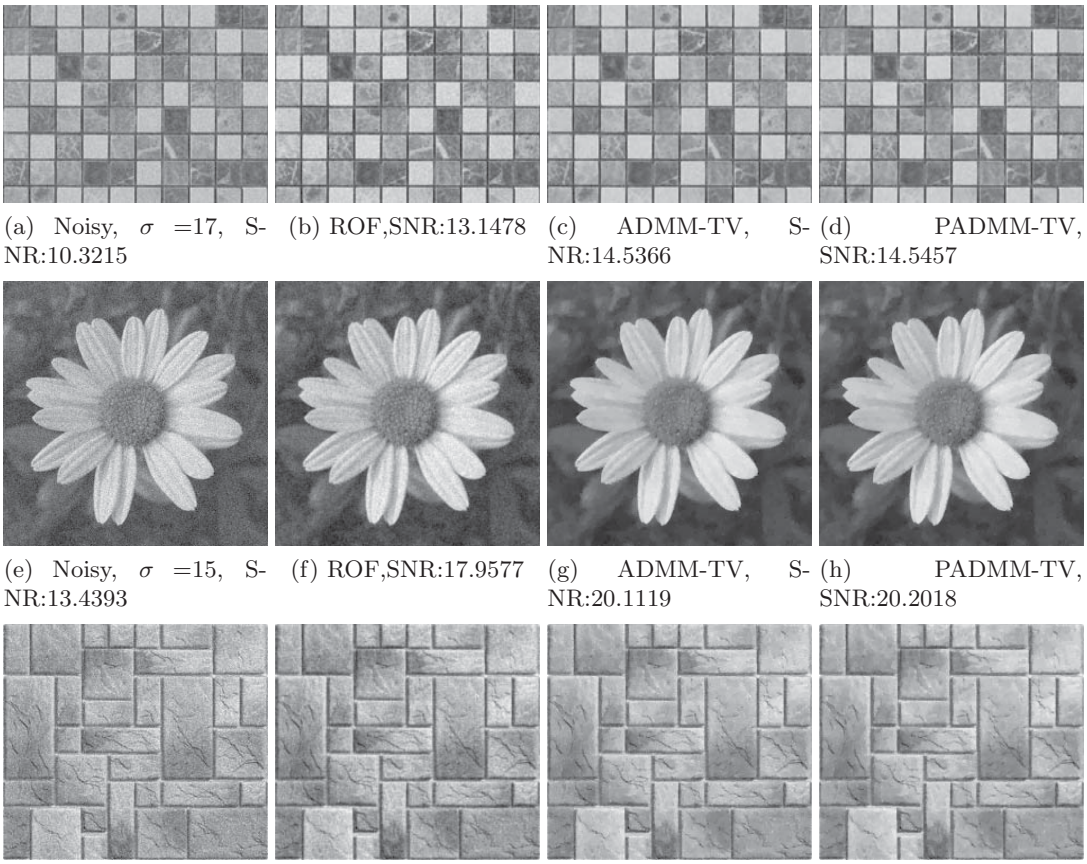
In this example firstly, we compare the very basic and pioneer ROF model [37] with Vese-Osher's decomposition model (1.1). During our experiments, we observed that although the ROF model is faster than the Vese-Osher's decomposition model, but it removes small texture and little stairs appeared in the denoised image see Fig. 12. Secondly, we compare the Euler's elastica-based model [43] (EEALM) with our proposed Euler's elastica-based decomposition model (1.2). We observed that Euler's elastica-based decomposition model has better results in the sense of SNR and also quality, see Fig. 13. In these experiments, we show the cartoon plus texture $(u + v)$ as a reconstruction or denoised image for decomposition model while in the original model the minimizer u is as a denoised image. We compare the CPU time and number of iteration of these models in Table. 4, where our proposed PADMM-TV is faster than the ADMM-TV. Similarly, PADMM-EE is more faster than EEALM.

The performance is evaluated by using the signal to noise ratio (SNR) and the SNR is defined as:

$$SNR \triangleq 10 \log_{10} \frac{\|u - \bar{u}\|^2}{\|u - \hat{u}\|^2},$$

where u is the original signal, $\bar{u}$ is the mean intensity value of u and $\hat{u}$ is recovered signal, in decomposition case the recovered signal becomes $u + v$. For each experiment, the parameters are estimated so that the best SNR is obtained.

In our experiments, Gaussian white noise with zero mean and the standard deviation 17, 15 and 10 are used for images in Fig. 2(e), Fig. 2(f) and Fig. 2(g) respectively. As we have reduced the parameters associated with lagrange multipliers in Section 2 and 3, but for the better results of image denoising and for minimizing of lagrange



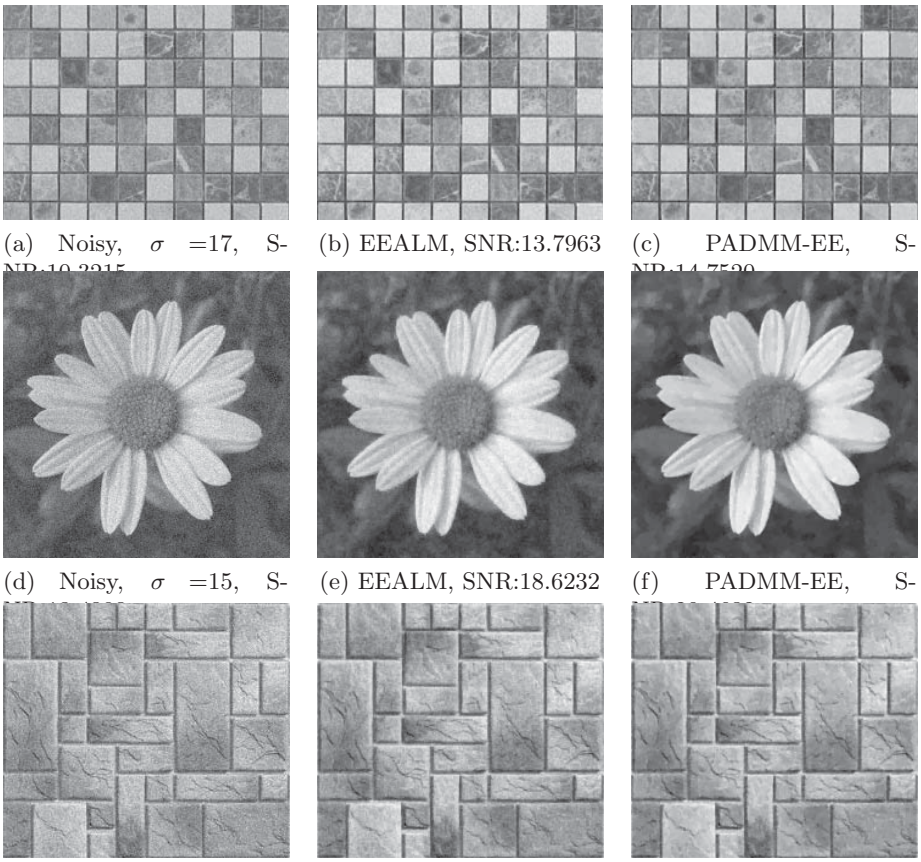
(a) Noisy, σ =17, SNR:10.3215 (b) ROF,SNR:13.1478 (c) ADMM-TV, SNR:14.5366 (d) PADMM-TV, SNR:14.5457

(e) Noisy, σ =15, SNR:13.4393 (f) ROF,SNR:17.9577 (g) ADMM-TV, SNR:20.1119 (h) PADMM-TV, SNR:20.2018

(i) Noisy, σ =10, SNR:7.9816 (j) ROF,SNR:10.8806 (k) ADMM-TV, SNR:11.3328 (l) PADMM-TV, SNR:11.3296

Figure 12: Image denoising caparison between TV based models. The results for ROF, ADMM-TV and PADMM-TV are shown from left to right respectively. Parameters for ADMM-TV, Floor ($\alpha = 0.11, \beta = 10, \gamma_{\mathbf{p}} = 0.1, \gamma_{\mathbf{q}} = 0.1$), Flower ($\alpha = 0.09, \beta = 9.5, \gamma_{\mathbf{p}} = 0.1, \gamma_{\mathbf{q}} = 0.1$) and Wall ($\alpha = 0.2, \beta = 10, \gamma_{\mathbf{p}} = 0.09, \gamma_{\mathbf{q}} = 0.1$). For PADMM-TV, Floor ($\alpha = 0.11, \beta = 10, \gamma_{\mathbf{p}} = 0.2, \gamma_{\mathbf{q}} = 0.1, \tau_1 = 1$), Flower ($\alpha = 0.09, \beta = 9.5, \gamma_{\mathbf{p}} = 0.1, \gamma_{\mathbf{q}} = 0.1, \tau_1 = 0.8$) and Wall ($\alpha = 0.17, \beta = 10, \gamma_{\mathbf{p}} = 0.12, \gamma_{\mathbf{q}} = 0.1, \tau_1 = 1$).

multipliers individually, we tuned all the parameters (see captions in Fig. 12 and Fig. 13). Finally, we plots numerical energy, relative error in u^{k+1} and SNR verses number of iterations for the image Floor (see Fig. 14). As we can observe from these plots in Fig. 14, our proposed algorithm for model (1.2) has converged.



(a) Noisy, σ =17, S-NR:10.3215 (b) EEALM, SNR:13.7963 (c) PADMM-EE, S-NR:14.7520

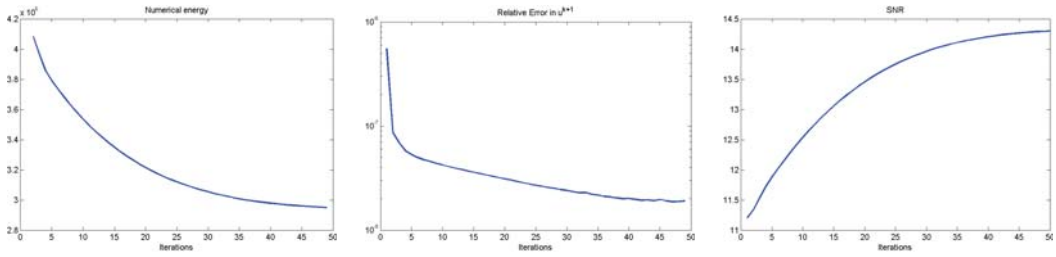
(d) Noisy, σ =15, S- (e) EEALM, SNR:18.6232 (f) PADMM-EE, S-

(g) Noisy, σ =10, S-NR:7.9816 (h) EEALM, SNR:11.2021 (i) PADMM-EE, S-NR:11.7696

Figure 13: Image denoising caparison between Euler's elastica based models. The results for Euler's elastica [43] and PADMM-EE are shown from left to right respectively. Parameters for PADMM-EE, Floor ($\alpha = 0.12, \beta = 12.5, a = 0.9, b = 10, \gamma_{\mathbf{p}} = 0.15, \gamma_{\mathbf{n}} = 0.3, \gamma_h = 0.01, \gamma_{\mathbf{q}} = 0.09, \tau_1 = \tau_2 = 1.1$), Flower($\alpha = 0.1, \beta = 10.5, a = 0.9, b = 10, \gamma_{\mathbf{p}} = 0.1, \gamma_{\mathbf{n}} = 0.45, \gamma_h = 0.01, \gamma_{\mathbf{q}} = 0.09, \tau_1 = \tau_2 = 1.2$) and Wall ($\alpha = 0.17, \beta = 9, a = 0.6, b = 10, \gamma_{\mathbf{p}} = 0.105, \gamma_{\mathbf{n}} = 0.6, \gamma_h = 0.002, \gamma_{\mathbf{q}} = 0.1, \tau_1 = \tau_2 = 1.5$)

5. Conclusion

Vese-Osher's model is used for texture extraction and image denoising in the literature of applied mathematics and image processing. Although the classical ADMM can be used to solve these problems but it is very challenging to construct efficient algorithms when minimization problem has non-linear high order derivatives. In this paper, we observed that one can solve these problems more fast by proximal ADMM in which one more subproblem has closed form solution. Firstly, we solved Vese-Osher's decomposition model by proximal ADMM in which we linearized g-subproblem to have a close form solution. Secondly, we solved Vese-Osher's model having Euler's elastica energy



(a) (b) (c)

Figure 14: Plots of numerical energy, relative error of u^{k+1} and SNR of our proposed Euler's elastica-based decomposition model for Fig. 2(e).

term instead of total variation based on the proximal ADMM. All the subproblems are easy to solve and the method is globally convergent at a linear rate for Vese-Osher's model. We also present some preliminary numerical results, which indicate that our method is not only efficient but it also can improve the quality of images in case of image denoising.

Acknowledgments

The work of Dr. H. Chang was partially supported by National Natural Science Foundation of China (Nos. 11501413, 11426165), the China Scholarship Council, 2017-Outstanding Young Innovation Team Cultivation Program 043-135202TD1703, Young Backbone of Innovative Personnel Training Program 043-135205GC37, and Innovation Project 043-135202XC1605 of Tianjin Normal University. Dr. Y. Duan was supported by National Natural Science Foundation of China (Grant No. 11701418). Dr. C. Wu was supported by National Natural Science Foundation of China (Grant No. 11301289 and 11531013). Prof. X.-C. Tai is supported by HKBU startup grant and grant RG(R)-RC/17-18/02-MATH.

References

[1] L. Ambrosio and S. Masnou. A direct variational approach to a problem arising in image reconstruction. *Interfaces Free Bound.*, 5(1):63–81, 2003.

[2] G. Aubert and J.-F. Aujol. Modeling very oscillating signals. application to image processing. *Appl. Math. Optim.*, 51(2):163–182, 2005.

[3] G. Aubert and P. Kornprobst. *Mathematical problems in image processing: partial differential equations and the calculus of variations*, volume 147. Springer Science & Business Media, 2006.

[4] J.-F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle. Image decomposition into a bounded variation component and an oscillating component. *J. Math. Imaging Vision*, 22(1):71–88, 2005.

[5] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition–modeling, algorithms, and parameter selection. *Int. J. Comput. Vis.*, 67(1):111–136, 2006.

[6] E. Bae, J. Shi, and X.-C. Tai. Graph cuts for curvature based image denoising. *IEEE Trans. Image Process.*, 20(5):1199–1210, 2011.
[7] C. Ballester, V. Caselles, and J. Verdera. Disocclusion by joint interpolation of vector fields and gray levels. *Multiscale Model. Simul.*, 2(1):80–123, 2003.
[8] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Trans. Image Process.*, 12(8):882–889, 2003.
[9] C. Brito-Loeza and K. Chen. Fast numerical algorithms for Euler's elastica inpainting model. *Int. J. Mod. Math*, 5:157–182, 2010.
[10] A. Buades, T. M. Le, J.-M. Morel, and L. A. Vese. Fast cartoon+ texture image filters. *IEEE Trans. Image Process.*, 19(8):1978–1986, 2010.
[11] A. Buades and J.-L. Lisani. Directional filters for cartoon+ texture image decomposition. *Image Processing On Line*, 6:75–88, 2016.
[12] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76(2):167–188, 1997.
[13] T. F. Chan, S. Esedoglu, and F. E. Park. Image decomposition combining staircase reduction and texture extraction. *J. Vis. Commun. Image Represent.*, 18(6):464–486, 2007.
[14] T. F. Chan and J. Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.
[15] H. Chang, X.-C. Tai, L.-L. Wang, and D. Yang. Convergence rate of overlapping domain decomposition methods for the rudin–osher–fatemi model based on a dual formulation. *SIAM J. Imaging Sci.*, 8(1):564–591, 2015.
[16] Y. Cui, X. Li, D. Sun, and K.-C. Toh. On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions. *J. Optim. Theory Appl.*, 169(3):1013–1041, 2016.
[17] W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *J. Sci. Comput.*, 66(3):889–916, 2016.
[18] Y. Duan, Y. Wang, and J. Hahn. A fast augmented Lagrangian method for Euler's elastica models. *Numer. Math. Theory Methods Appl.*, 6(01):47–71, 2013.
[19] J. Duran, M. Moeller, C. Sbert, and D. Cremers. Collaborative total variation: a general framework for vectorial TV models. *SIAM J. Imaging Sci.*, 9(1):116–151, 2016.
[20] J. Duran, M. Moeller, C. Sbert, and D. Cremers. On the implementation of collaborative TV regularization: Application to cartoon+ texture decomposition. *Image Processing On Line*, 6:27–74, 2016.
[21] S. Esedoglu and R. March. Segmentation with depth but without detecting junctions. *J. Math. Imaging Vision*, 18(1):7–15, 2003.
[22] M. Fazel, T. K. Pong, D. Sun, and P. Tseng. Hankel matrix rank minimization with applications to system identification and realization. *SIAM J. Matrix Anal. Appl.*, 34(3):946–977, 2013.
[23] M. Fornasier and H. Rauhut. Recovery algorithms for vector-valued data with joint sparsity constraints. *SIAM J. Numer. Anal.*, 46(2):577–613, 2008.
[24] M. Fortin and R. Glowinski. *Augmented Lagrangian Methods: Applications to the numerical solution of boundary-value problems*, volume 15 of *Studies in Mathematics and Its Applications*. Elsevier, 1983.
[25] J. B. Garnett, T. M. Le, Y. Meyer, and L. A. Vese. Image decompositions using bounded variation and generalized homogeneous Besov spaces. *Appl. Comput. Harmon. Anal.*, 23(1):25–56, 2007.

[26] G. Gilboa. A total variation spectral framework for scale and texture analysis. *SIAM J. Imaging Sci.*, 7(4):1937–1961, 2014.
[27] T. Goldstein and S. Osher. The split bregman method for L1-regularized problems. *SIAM J. Imaging Sci.*, 2(2):323–343, 2009.
[28] J. Hahn, C. Wu, and X.-C. Tai. Augmented lagrangian method for generalized tv-stokes model. *Journal of Scientific Computing*, 50(2):235–264, 2012.
[29] B. He, L.-Z. Liao, D. Han, and H. Yang. A new inexact alternating directions method for monotone variational inequalities. *Math. Program.*, 92(1):103–118, 2002.
[30] T. M. Le and L. A. Vese. Image decomposition using total variation and div(BMO). *Multiscale Model. Simul.*, 4(2):390–423, 2005.
[31] V. Le Guen. Cartoon+ texture image decomposition by the tv-l1 model. *Image Processing On Line*, 4:204–219, 2014.
[32] M. Lysaker, A. Lundervold, and X.-C. Tai. Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time. *IEEE Trans. Image Process.*, 12(12):1579–1590, 2003.
[33] S. Masnou and J.-M. Morel. Level lines based disocclusion. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, pages 259–263. IEEE, 1998.
[34] Y. Meyer. *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, volume 22. American Mathematical Soc., 2001.
[35] D. Mumford. Elastica and computer vision. In *Algebraic geometry and its applications*, pages 491–506. Springer, 1994.
[36] M. Nitzberg, D. Mumford, and T. Shiota. *Filtering, segmentation and depth*. 1993.
[37] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1):259–268, 1992.
[38] C. W. Samad Wali, Zhifang Liu and H. Chang. A boosting procedure for variational-based image restoration. *Numer. Math. Theory Methods Appl.*, (In Press), 2017.
[39] H. Schaeffer and S. Osher. A Low Patch-Rank Interpretation of Texture. *SIAM J. Imaging Sci.*, 6(1):226–262, Jan. 2013.
[40] J. Shen, S. H. Kang, and T. F. Chan. Euler's elastica and curvature-based inpainting. *SIAM J. Appl. Math.*, 63(2):564–592, 2003.
[41] D. Szolgay and T. Szirányi. Adaptive image decomposition into cartoon and texture parts optimized by the orthogonality criterion. *IEEE Trans. Image Process.*, 21(8):3405–3415, 2012.
[42] X.-C. Tai. Fast numerical schemes related to curvature minimization: a brief and elementary review. *Actes des rencontres du CIRM*, 3(1):17–30, 2013.
[43] 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):313–344, 2011.
[44] L. A. Vese and C. Le Guyader. *Variational methods in image processing*. CRC Press, 2015.
[45] L. A. Vese and S. J. Osher. Modeling textures with total variation minimization and oscillating patterns in image processing. *J. Sci. Comput.*, 19(1-3):553–572, 2003.
[46] L. A. Vese and S. J. Osher. Image denoising and decomposition with total variation minimization and oscillatory functions. *J. Math. Imaging Vision*, 20(1-2):7–18, 2004.
[47] 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):300–339, 2010.
[48] C. Wu, J. Zhang, and X.-C. Tai. Augmented Lagrangian method for total variation

restoration with non-quadratic fidelity. *Inverse Probl. Imaging*, 5(1):237–261, 2011.

[49] J. Zhang and K. Chen. A total fractional-order variation model for image restoration with nonhomogeneous boundary conditions and its numerical solution. *SIAM J. Imaging Sci.*, 8(4):2487–2518, 2015.

[50] J. Zhang, R. Chen, C. Deng, and S. Wang. Fast linearized augmented lagrangian method for euler's elastica model. *Numer. Math. Theory Methods Appl.*, 10(1):98–115, 2017.

[51] W. Zhu, X.-C. Tai, and T. Chan. Augmented Lagrangian method for a mean curvature based image denoising model. *Inverse Probl. Imaging*, 7(4):1409–1432, 2013.

[52] W. Zhu, X.-C. Tai, and T. Chan. Image segmentation using Euler's elastica as the regularization. *J. Sci. Comput.*, 57(2):414–438, 2013.