Contents lists available at SciVerse ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

Lagrangian multipliers and split Bregman methods for minimization problems constrained on $\mathcal{S}^{n-1}$

Fang Li [a], Tieyong Zeng [b], Guixu Zhang [c,*]

[a] Department of Mathematics, East China Normal University, Shanghai, China
[b] Centre for Mathematical Imaging and Vision, and Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong
[c] Department of Computer Science, East China Normal University, Shanghai, China

ARTICLE INFO

Article history:
Received 10 February 2012
Accepted 2 July 2012
Available online 11 July 2012

Keywords:
Lagrangian method
Split Bregman method
Total variation

ABSTRACT

The numerical methods of total variation (TV) model for image denoising, especially Rudin–Osher–Fatemi (ROF) model, is widely studied in the literature. However, the $\mathcal{S}^{n-1}$ constrained counterpart is less addressed. The classical gradient descent method for the constrained problem is limited in two aspects: one is the small time step size to ensure stability; the other is that the data must be projected onto $\mathcal{S}^{n-1}$ during evolution since the unit norm constraint is poorly satisfied. In order to avoid these drawbacks, in this paper, we propose two alternative numerical methods based on the Lagrangian multipliers and split Bregman methods. Both algorithms are efficient and easy to implement. A number of experiments demonstrate that the proposed algorithms are quite effective in denoising of data constrained on S^1 or S^2, including general direction data diffusion and chromaticity denoising.

1. Introduction

Variational denoising methods have become popular in recent years, for instance, the well known Rudin–Osher–Fatemi (ROF) model [26] and its various extensions [11,16,19]. The scalar ROF model for gray-scale image is:

$$\min_u \int_\Omega |\nabla u| dx + \frac{\lambda}{2}\int_\Omega (f-u)^2 dx, \tag{1}$$

where f is the observed noisy image and λ is a positive balance parameter. Here, the first term is called total variation (TV) which is widely used as a regularization term in variational image processing approaches [1]. In the past decades, a large amount of fast numerical schemes instead of the gradient descent methods are proposed to handle the TV based minimization models. For instance, the Chambolle's fast dual method [7], the alternating split Bregman method [17], the operator splitting method [12,20,22], the alternating direction method of multipliers (ADMM) [15,24], the primal–dual method [8,13] and some other methods [2,3,25,23,33].

Let us now write down the n-dimensional ROF model constrained on $\mathcal{S}^{n-1}$. Assume $\Omega \subset \mathbb{R}^2$ is an open bounded domain, and $\mathbf{f} : \Omega \to \mathcal{S}^{n-1} \subset \mathbb{R}^n$ is the observed noisy data and $\mathbf{u} : \Omega \to \mathbb{R}^n$ is a vectorial function. The general problem can be formulated as:

$$\min_{\mathbf{u}} E(\mathbf{u}) = \int_\Omega \|\nabla \mathbf{u}\| dx + \frac{\lambda}{2}\int_\Omega |\mathbf{u}-\mathbf{f}|^2 dx \quad s.t.\ |\mathbf{u}| = 1, \tag{2}$$

where

$$\nabla \mathbf{u} = \begin{pmatrix} \nabla u_1 \\ \nabla u_2 \\ \vdots \\ \nabla u_n \end{pmatrix} = \begin{pmatrix} \frac{\partial u_1}{\partial x_1} & \frac{\partial u_1}{\partial x_2} \\ \frac{\partial u_2}{\partial x_1} & \frac{\partial u_2}{\partial x_2} \\ \vdots & \vdots \\ \frac{\partial u_n}{\partial x_1} & \frac{\partial u_n}{\partial x_2} \end{pmatrix},$$

with $x = (x_1, x_2)$ denotes the coordinates in image domain Ω,

$$|\mathbf{u}| = \sqrt{u_1^2 + u_2^2 + \cdots + u_n^2}$$

and

$$\|\nabla \mathbf{u}\| = \sqrt{|\nabla u_1|^2 + |\nabla u_2|^2 + \cdots + |\nabla u_n|^2}.$$

In fact, $\int_\Omega \|\nabla \mathbf{u}\| dx$ is a generalization of color TV [4]. Remark that the problem is nonconvex since the constraint $|\mathbf{u}| = 1$ is not convex.

The above model can be used for direction data diffusion where the direction data has unit norm. An example in image processing field is chromaticity denoising. Although most of the variational denoising models use Right-Green-Blur (RGB) color model, there are some methods use other color models especially Chromaticity-Brightness (CB) color model. In the CB color model, the chromaticity component $\mathbf{u}$ and the brightness component B can be calculated as follows:

$$B = |\mathbf{u}| := \sqrt{u_1^2 + u_2^2 + u_3^2}, \quad \mathbf{C} = \frac{\mathbf{u}}{|\mathbf{u}|}.$$

[*] Corresponding author.
E-mail address: gxzhang@cs.ecnu.edu.cn (G. Zhang).

1047-3203/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jvcir.2012.07.002

Note that the chromaticity $\mathbf{C}$ is a vector lives on the unit sphere in $\mathbb{R}^3 : \mathcal{S}^2 = \{\xi \in \mathbb{R}^3 : |\xi| = 1\}$. Therefore chromaticity is belonging to non-flat image feature differs from other features defined in Euclidean space. The CB model is known to be closer to human perception which is widely used in color image representation and modeling. In [9], it is shown that using CB color model gives better color control and detail recovery for color image denoising compared with other color models.

In literatures, some methods are introduced to handle the minimization problems on $\mathcal{S}^{n-1}$. Tang et al. in [29,30] proposed to denoise chromaticity or general direction data via p-harmonic maps in liquid crystals. The classical gradient descent method is used to solve the corresponding Euler–Lagrange equation which is limited by small time step and converges slowly. Recall that the gradient descent method for problem (2) is the flow [9]:

$$\frac{\partial \mathbf{u}}{\partial t} = \operatorname{div}\left(\frac{\nabla \mathbf{u}}{\|\nabla \mathbf{u}\|}\right) - \mathbf{u}\|\nabla \mathbf{u}\| + \lambda(\mathbf{f} - \langle \mathbf{f}, \mathbf{u}\rangle \mathbf{u}). \tag{3}$$

More generally, Tschumperl and Deriche in [31] studied the orthonormal vector sets diffusion problem by ϕ-function regularization [1] and the related negative gradient flow. In [32], Vese and Osher changed the constrained p-harmonic problem:

$$\min_{|\mathbf{u}|=1} \int_\Omega \|\nabla \mathbf{u}\|^p dx,$$

as an unconstrained one:

$$\min_V \int_\Omega \left\|\nabla\left(\frac{V}{|V|}\right)\right\|^p dx.$$

Numerically, the gradient descent method with implicit scheme is applied to evolve V based on polar coordinates. With similar idea, Cecil et al. in [6] proposed numerical methods for minimization problems constrained on $\mathcal{S}^1$ and $\mathcal{S}^2$ by technique based on the angle formulation, and numerically gradient descent method is used. In [10], Chan and Shen used vectorial ROF model to denoise non-flat data. Numerically, they developed fixed-point iteration. Bresson and Chan in [5] extended Chambolle's dual algorithm to vectorial ROF model, meanwhile, they generalize the algorithm to denoise the chromaticity component in color image. In [18], Haehnle and Prohl proposed discrete finite element based algorithms to approximate the L^2 gradient flow of the Mumford–Shah–Euler functional for unit vector fields and applied the algorithms in color image inpainting. In [34], Goldfarb et al. proposed new gradient descent algorithms for the p-harmonic flow problem on spheres, which searches the step along a curve that lies on the sphere and can preserve the pointwise sphere constraints. The method is generalized by Wenand Yin in [35] to handle the general orthogonal constraints.

In this paper, we consider two alternative numerical algorithms to solve problem (2) constrained on $\mathbf{S}^{n-1}$. Our main idea is to split the original problem into easier subproblems by introducing auxiliary variables. In Algorithm 1, we first use the standard Lagrangian method to handle the pointwise unit norm constraint, and then relax the energy by adding an auxiliary variable. In Algorithm 2, we first derive an equivalent problem with two auxiliary variables and three constraints, and then use the split Bregman method to handle the constraints. In both methods, all the involved subproblems are easy to solve.

The outline of this paper is as follows. In Section 2, we propose our Algorithm 1 based on Lagrangian multipliers method. In Section 3, we develop our Algorithm 2 based on the so called split Bregman method. The numerical results including direction data diffusion on S^1 and chromaticity denoising on S^2 are reported in Section 4. Finally, we conclude the paper in Section 5.

2. Algorithm 1 – Lagrangian multipliers method

In this section, we propose the Algorithm 1 to solve problem (2). Since the pointwise constraint $\mathbf{u}(x) = 1$ is equivalent to $|\mathbf{u}(x)|^2 - 1 = 0$, by using Lagrange multipliers method on the constraints we get an equivalent unconstrained problem:

$$\min_{\mathbf{u},\mu} \left\{ \begin{array}{l} E_1(\mathbf{u},\mu) = \int_\Omega \|\nabla \mathbf{u}\| dx + \frac{\lambda}{2}\int_\Omega |\mathbf{u} - \mathbf{f}|^2 dx \\ + \frac{1}{2}\int_\Omega \mu(x)(|\mathbf{u}(x)|^2 - 1)dx \end{array} \right\}, \tag{4}$$

where $\mu(x)$ is the Lagrange multiplier at point $x \in \Omega$. The problem is not easy to solve since TV term is nonsmooth. In order to find an efficient algorithm, we consider an approximate problem by adding new variables such that the new problem is easy to solve. We add a new variable $\mathbf{v}$ to approximate $\mathbf{u}$ and obtain an approximate problem:

$$\min_{\mathbf{u},\mathbf{v},\mu} \left\{ \begin{array}{l} E_2(\mathbf{u},\mathbf{v},\mu) = \int_\Omega \|\nabla \mathbf{v}\| dx + \frac{1}{2\theta}\int_\Omega |\mathbf{v} - \mathbf{u}|^2 dx \\ + \frac{\lambda}{2}\int_\Omega |\mathbf{u} - \mathbf{f}|^2 dx + \frac{1}{2}\int_\Omega \mu(x)(|\mathbf{u}|^2 - 1)dx \end{array} \right\}, \tag{5}$$

where θ is small enough to ensure that $\mathbf{u}$ almost equals $\mathbf{v}$. In the following subsections, we will derive the formulas for updating $\mathbf{u}$, μ and $\mathbf{v}$ in problem (5), respectively with alternating minimization method.

2.1. Solving u

Fixing μ and $\mathbf{u}$, the subproblem for $\mathbf{u}$ is:

$$\min_{\mathbf{u}} \left\{ \begin{array}{l} \frac{1}{2\theta}\int_\Omega |\mathbf{v} - \mathbf{u}|^2 dx \\ + \frac{\lambda}{2}\int_\Omega |\mathbf{u} - \mathbf{f}|^2 dx + \frac{1}{2}\int_\Omega \mu(x)(|\mathbf{u}|^2 - 1)dx \end{array} \right\}. \tag{6}$$

The corresponding Euler–Lagrange equation about $\mathbf{u}$ is:

$$\frac{1}{\theta}(\mathbf{u} - \mathbf{v}) + \lambda(\mathbf{u} - \mathbf{f}) + \mu \mathbf{u} = 0. \tag{7}$$

Then we derive the closed-form solution of $\mathbf{u}$:

$$\mathbf{u} = \frac{\mathbf{v} + \lambda\theta \mathbf{f}}{1 + \lambda\theta + \mu\theta}. \tag{8}$$

2.2. Solving the Lagrange multipliers μ

Taking derivative of E_2 with respect to μ and setting it to zero, we get:

$$|\mathbf{u}|^2 = \langle \mathbf{u}, \mathbf{u}\rangle = 1, \tag{9}$$

for each $x \in \Omega$, where $\langle\cdot\rangle$ denotes the inner product in $\mathbb{R}^3$. Taking the inner product of (7) with $\mathbf{u}$ and using (9), we obtain the closed-form solution of μ:

$$\mu = \frac{1}{\theta}\langle \mathbf{u}, \mathbf{v}\rangle + \lambda\langle \mathbf{u}, \mathbf{f}\rangle - \frac{1}{\theta} - \lambda. \tag{10}$$

Remark that the above formula (10) was also derived in [21] and successfully used in colorization problems.

2.3. Solving auxiliary variable v

Fixing $\mathbf{u}$, the subproblem for $\mathbf{v}$ is:

$$\min_{\mathbf{v}} \int_\Omega \|\nabla \mathbf{v}\| dx + \frac{1}{2\theta}\int_\Omega |\mathbf{v} - \mathbf{u}|^2 dx, \tag{11}$$

which is a standard vectorial ROF model. Recall that many fast numerical algorithm have been designed to solve the scalar ROF model, see Section 1. These fast algorithms can be directly used to solve vectorial ROF model when a channel by channel TV is used. That is because in every channel the problem becomes a scalar

ROF model. However, in color TV model the channels are combined together, and only Chambolle's algorithm has been generalized [5]. Here we generalize the operator splitting method used in [20] to solve problem (11) which is easy to understand.

Let $L = \nabla$, $f = \|\cdot\|_1$ be the L^1 norm of matrix vector module, then $f(\nabla \mathbf{v}) = \int_\Omega \|\nabla \mathbf{v}\| dx$, and $L^* = \nabla^T$ is the adjoint operator of ∇. Using these notations, (11) can be rewritten as:

$$\min_{\mathbf{u}} f(L(\mathbf{v})) + \frac{1}{2\theta}\int_\Omega |\mathbf{v}-\mathbf{u}|^2 dx.$$

The corresponding Euler–Lagrange equation is:

$$0 \in \partial(f \circ L)(\mathbf{v}) + \frac{1}{\theta}(\mathbf{v}-\mathbf{u}). \tag{12}$$

By property of subgradient, $\partial(f \circ L)(\mathbf{v}) = L^*\partial f(L\mathbf{v})$. Define $L^*\mathbf{y} \in L^*\partial f(L\mathbf{v})$ where $\mathbf{y}$ is a $n \times 2$ matrix data, then $\mathbf{y} \in \partial f(L\mathbf{v})$ which is equivalent to $L\mathbf{v} \in \partial f^*(\mathbf{y})$. Hence $\mathbf{v}$ satisfies (12) if and only if there exists an auxiliary variable $\mathbf{y}$, such that:

$$0 \in L^*\mathbf{y} + \frac{1}{\theta}(\mathbf{v}-\mathbf{u}), \tag{13}$$

$$0 \in \partial f^*(\mathbf{y}) - L\mathbf{v}. \tag{14}$$

We can now apply the operator splitting method with scalar $\tau > 0$ to (14) and obtain two equations:

$$0 \in \tau\partial f^*(\mathbf{y}) + \mathbf{y} - \mathbf{t}, \tag{15}$$

$$\mathbf{t} = \mathbf{y} + \tau L\mathbf{v}. \tag{16}$$

Note that (15) is also equivalent to:

$$0 \in \tau\mathbf{y} + \partial f(\mathbf{y}-\mathbf{t}),$$

which is the optimality condition of:

$$\min_{\mathbf{y}} \frac{\tau}{2}\int_\Omega \|\mathbf{y}\|^2 dx + f(\mathbf{y}-\mathbf{t}). \tag{17}$$

Problem (18) has closed-form solution:

$$\mathbf{y} = \min\left\{\|\mathbf{t}\|, \frac{1}{\tau}\right\}\frac{\mathbf{t}}{\|\mathbf{t}\|}. \tag{18}$$

Sum up (13), (16) and (18), the minimization problem (11) can be solved by the following alternating iteration:

$$\mathbf{v} = \mathbf{u} - \theta\nabla^T\mathbf{y}, \tag{19}$$

$$\mathbf{y} = \min\left\{\frac{1}{\tau}, \|\mathbf{y}+\tau\nabla\mathbf{v}\|\right\}\frac{\mathbf{y}+\tau\nabla\mathbf{v}}{\|\mathbf{y}+\tau\nabla\mathbf{v}\|}. \tag{20}$$

We summarize the above steps in the following as Algorithm 1:

Algorithm 1:

- Initialization: $\mathbf{u}^0 = \mathbf{f}$, $\mathbf{v}^0 = \mathbf{f}$, $\mathbf{y}^0 = \mathbf{0}$.
- For $k = 0, 1, 2, \ldots,$ repeat until a stopping criterion is reached

$$\mu^{k+1} = \frac{1}{\theta}\langle \mathbf{u}^k, \mathbf{v}^k\rangle + \lambda\langle \mathbf{u}^k, \mathbf{f}\rangle - \frac{1}{\theta} - \lambda,$$

$$\mathbf{u}^{k+1} = \frac{\mathbf{v}^k + \lambda\theta\mathbf{f}}{1+\lambda\theta+\mu^k\theta},$$

$$\mathbf{v}^{k+1} = \mathbf{u}^k - \theta\nabla^T\mathbf{y}^k,$$

$$\mathbf{y}^{k+1} = \min\left\{\frac{1}{\tau}, \|\mathbf{y}^k+\tau\nabla\mathbf{v}^k\|\right\}\frac{\mathbf{y}^k+\tau\nabla\mathbf{v}^k}{\|\mathbf{y}^k+\tau\nabla\mathbf{v}^k\|}.$$

- Output: $\mathbf{u}^{k+1}$.

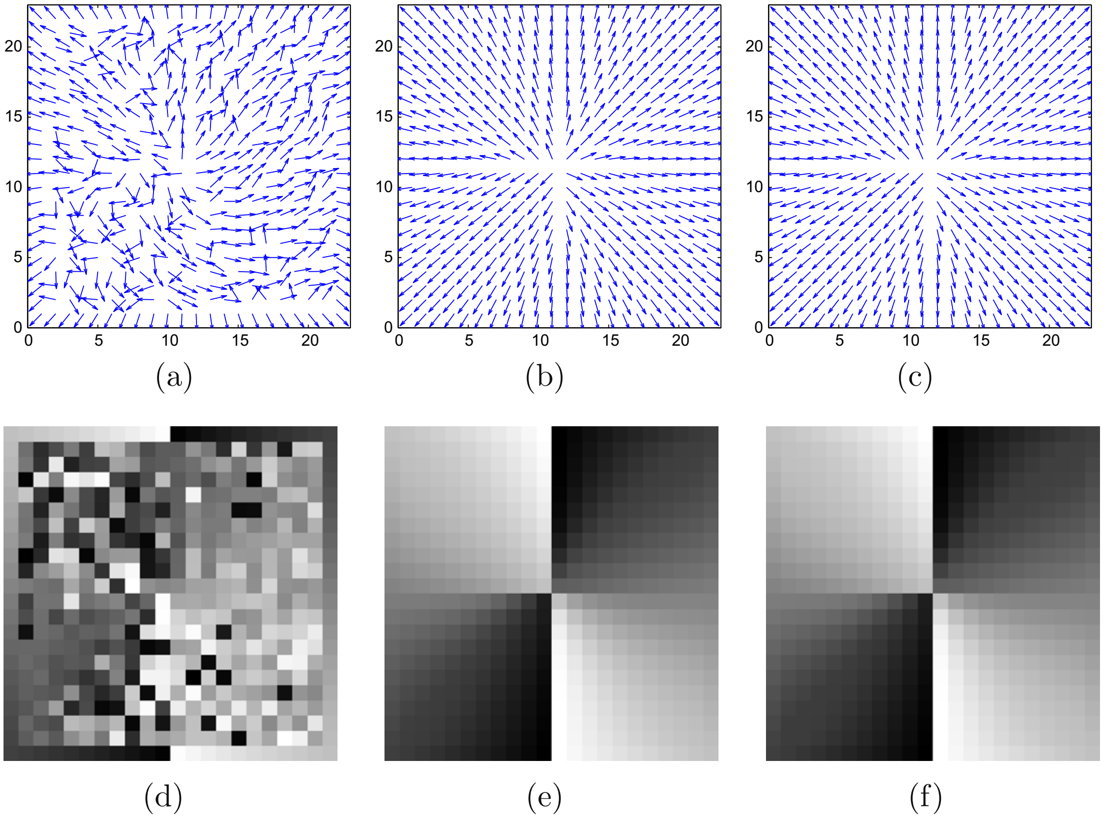
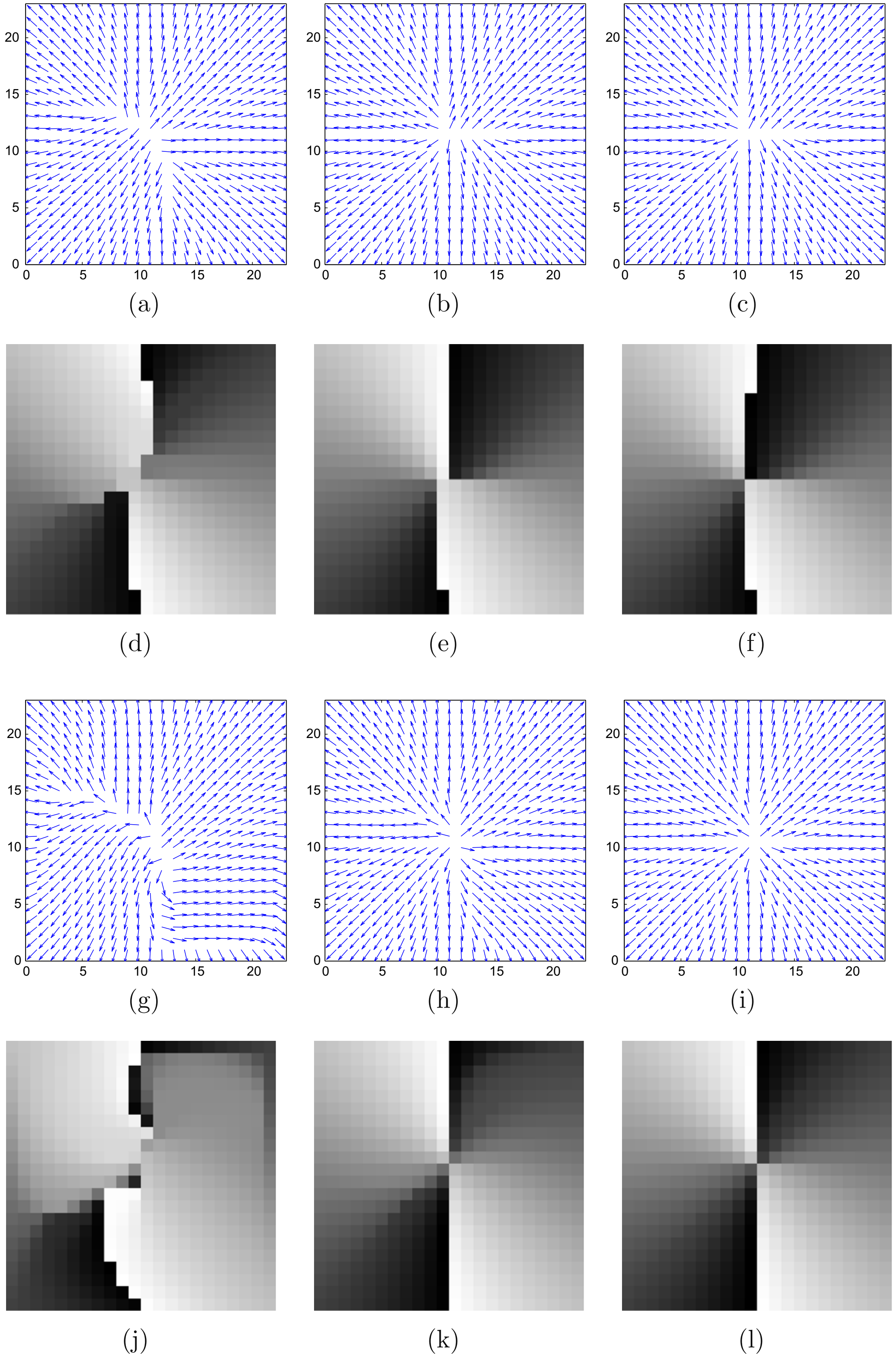


Fig. 1. S^1 data diffusion with Dirichlet boundary condition. (a) the initial data 1; (b) result by Algorithm 1, computational time = 0.49 s (parameters: $\lambda = 0$, $\theta = 1/30$, $\tau = 1$); (c) result by Algorithm 2, computational time = 1.05 s with FFT to solve $\mathbf{u}$, 0.68 s with GS iteration (parameters: $\lambda = 0$, $\delta_1 = 1$, $\delta_2 = 20$); (d) angle of initial data and (e) angle of result by Algorithm 1 and (f) angle of result by Algorithm 2.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Fig. 2. The results of gradient descent method and method in [32] on initial data 1, $\lambda = 0$, time step = 0.001. The first two rows are results of gradient descent method: (a) iteration = 10,000; (b) iteration = 40,000; (c) iteration = 50,000, computational time = 13.62 s and (d)–(f) angles corresponding to (a)–(c). The last two rows are results of method in [32]: (g) iteration = 1000; (h) iteration = 3000; (i) iteration = 5000, computational time = 0.74 s and (j)–(l) angles corresponding to (g)–(i).

3. Algorithm 2 – split Bregman method

In this section, we propose another algorithm to solve the constrained problem (2) based on split Bregman method. Firstly we give a brief introduction on split Bregman method. Assume $H(\cdot)$ and $|\Phi(\cdot)|$ are convex functionals and $|\Phi(\cdot)|$ is differentiable. Let us consider the problem:

$$\min_{u,d}\{|d| + H(u)\} \quad s.t.\ \Phi(u) = d.$$

The split Bregman method for this problem is given by:

$$(u^{k+1}, d^{k+1}) = \min_{u,d}\left\{|d| + H(u) + \frac{\delta}{2}\left\|b^k + \Phi(u) - d\right\|_2^2\right\},$$
$$b^{k+1} = b^k + \Phi(u^{k+1}) - d^{k+1}.$$

For convex problem, it has been proved that the split Bregman algorithm converges [27]. Recently, split Bregman method is also used in nonconvex problems to build efficient algorithm by splitting the original problem into easy subproblems [14,28]. Remark that the pointwise unit norm constraint is considered in both [14,28]. Here we follow the similar idea.

To implement the split Bregman method on problem (2), we first add two auxiliary variables $\mathbf{v}$, $\mathbf{w}$ and rewrite the problem (2) as:

$$\min_{\mathbf{u},\mathbf{v},\mathbf{w}}\left\{\int_\Omega \|\mathbf{v}\|dx + \frac{\lambda}{2}\int_\Omega |\mathbf{u}-\mathbf{f}|^2 dx\right\}$$
$$s.t. \quad \nabla\mathbf{u} = \mathbf{v}, \quad \mathbf{u} = \mathbf{w}, \quad |\mathbf{w}| = 1.$$

Then using the above split Bregman technique on both of the first two constraints, we get the iteration scheme:

$$(\mathbf{u}^{k+1}, \mathbf{v}^{k+1}, \mathbf{w}^{k+1}) = \min_{|\mathbf{w}|=1,\mathbf{u},\mathbf{v}} E_3(\mathbf{u},\mathbf{v},\mathbf{w}), \tag{21}$$
$$\mathbf{b}_1^{k+1} = \mathbf{b}_1^k + \nabla\mathbf{u}^{k+1} - \mathbf{v}^{k+1}, \tag{22}$$
$$\mathbf{b}_2^{k+1} = \mathbf{b}_2^k + \mathbf{u}^{k+1} - \mathbf{w}^{k+1}, \tag{23}$$

with

$$E_3(\mathbf{u},\mathbf{v},\mathbf{w}) = \left\{\begin{array}{l}\int_\Omega \|\mathbf{v}\|dx + \frac{\lambda}{2}\int_\Omega |\mathbf{u}-\mathbf{f}|^2dx + \\ \frac{\delta_1}{2}\int_\Omega \|\mathbf{b}_1^k + \nabla\mathbf{u} - \mathbf{v}\|^2 dx + \frac{\delta_2}{2}\int_\Omega |\mathbf{b}_2^k + \mathbf{u} - \mathbf{w}|^2 dx\end{array}\right\}, \tag{24}$$

where $\delta_1 > 0$, $\delta_2 > 0$ are parameters. In the following subsections, we derive the solutions for $\mathbf{u}$, $\mathbf{v}$ and $\mathbf{w}$, respectively from (21) and (24) with alternating minimization method.

3.1. Solving u

Fixing $\mathbf{v}$ and $\mathbf{w}$, the subproblem for $\mathbf{u}$ is:

$$\min_{\mathbf{u}}\left\{\frac{\lambda}{2}\int_\Omega |\mathbf{u}-\mathbf{f}|^2dx + \frac{\delta_1}{2}\int_\Omega \|\mathbf{b}_1^k + \nabla\mathbf{u} - \mathbf{v}\|^2 dx + \frac{\delta_2}{2}\int_\Omega |\mathbf{b}_2^k + \mathbf{u} - \mathbf{w}|^2 dx\right\}.$$

The corresponding Euler–Lagrangian equation about $\mathbf{u}$ is:

$$\lambda(\mathbf{u}-\mathbf{f}) + \delta_1\nabla^T(\mathbf{b}_1^k + \nabla\mathbf{u} - \mathbf{v}) + \delta_2(\mathbf{b}_2^k + \mathbf{u} - \mathbf{w}) = 0,$$

i.e.,

$$(\lambda + \delta_1\nabla^T\nabla + \delta_2)\mathbf{u} = \lambda\mathbf{f} + \delta_1\nabla^T(\mathbf{v} - \mathbf{b}_1^k) + \delta_2(\mathbf{w} - \mathbf{b}_2^k). \tag{25}$$

Then we have:

$$\mathbf{u} = \left(\lambda + \delta_1\nabla^T\nabla + \delta_2\right)^{-1}\left(\lambda\mathbf{f} + \delta_1\nabla^T(\mathbf{v} - \mathbf{b}_1^k) + \delta_2(\mathbf{w} - \mathbf{b}_2^k)\right) \tag{26}$$

Numerically $\mathbf{u}$ can be computed efficiently by fast fourier transform (FFT):

$$\mathbf{u} = \mathfrak{F}^{-1}\left(\frac{\mathfrak{F}\left(\lambda\mathbf{f} + \delta_1\nabla^T(\mathbf{v} - \mathbf{b}_1^k) + \delta_2(\mathbf{w} - \mathbf{b}_2^k)\right)}{\mathfrak{F}\left(\lambda + \delta_1\nabla^T\nabla + \delta_2\right)}\right). \tag{27}$$

Remark that another way to calculate $\mathbf{u}$ is via Gauss–Seidel (GS) iteration. Let

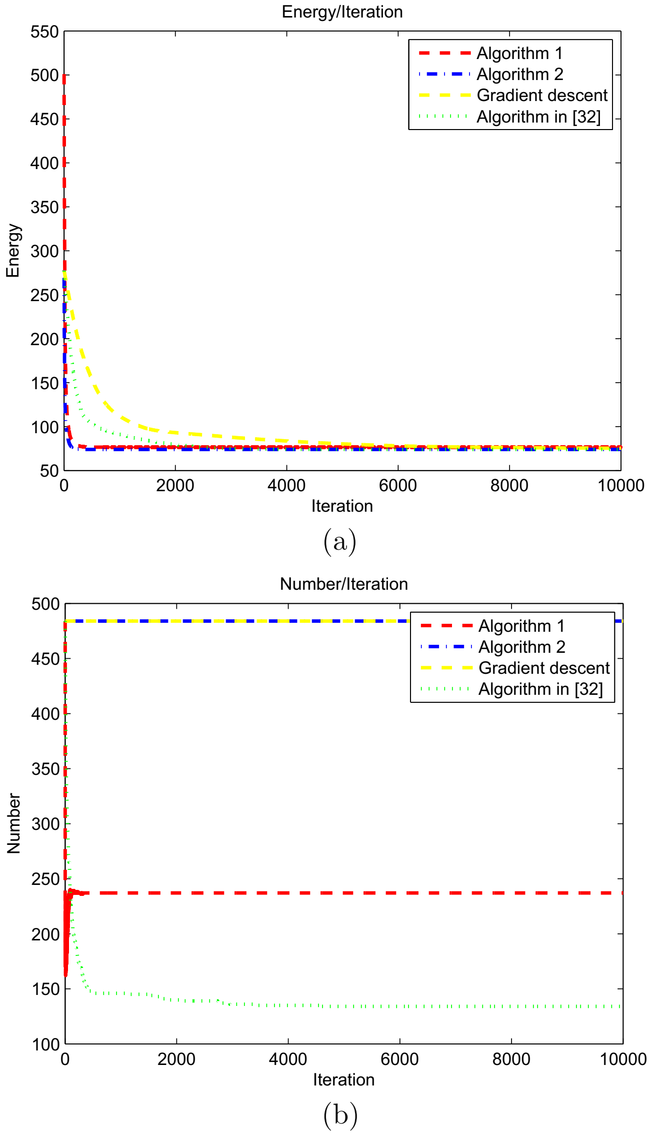
$$\text{rhs} = \lambda\mathbf{f} + \delta_1\nabla^T\left(\mathbf{v} - \mathbf{b}_1^k\right) + \delta_2\left(\mathbf{w} - \mathbf{b}_2^k\right).$$

Then the Eq. (25) can be discreted as:

$$(\lambda + \delta_2)u_{i,j}^{k+1} + \delta_1\left(4u_{i,j}^{k+1} - u_{i+1,j}^k - u_{i-1,j}^k - u_{i,j+1}^k - u_{i,j-1}^k\right) = \text{rhs}_{i,j}^k.$$

where i, j denote the grid. Then the Gauss–Seidel solution can be written component-wisely as:

$$u_{i,j}^{k+1} = \frac{\delta_1}{\lambda + 4\delta_1 + \delta_2}\left(u_{i+1,j}^k + u_{i-1,j}^k + u_{i,j+1}^k + u_{i,j-1}^k + \text{rhs}_{i,j}^k\right). \tag{28}$$



(a)

(b)

Fig. 3. Energy and number of pixels on $\mathbf{S}^1$ versus iteration applied on initial data 1. The proposed Algorithms 1, 2, the gradient descent method with renormalization and the algorithm in [32] are compared.

3.2. Solving $\mathbf{v}$

Fixing $\mathbf{u}$ and $\mathbf{w}$, the subproblem for $\mathbf{v}$ is:

$$\min_{\mathbf{v}}\left\{\int_{\Omega}\|\mathbf{v}\|dx+\frac{\delta_1}{2}\int_{\Omega}\|\mathbf{b}_1^k+\nabla\mathbf{u}-\mathbf{v}\|^2dx\right\}.$$

Let us give a simple calculation of the solution.

If $\mathbf{v}\neq 0$, the Euler–Lagrange equation about $\mathbf{v}$ is given by:

$$\frac{\mathbf{v}}{\|\mathbf{v}\|}+\delta_1(\mathbf{v}-\mathbf{b}_1^k-\nabla\mathbf{u})=0. \tag{29}$$

It is equivalent to:

$$\left(\frac{1}{\|\mathbf{v}\|}+\delta_1\right)\mathbf{v}=\delta_1\left(\mathbf{b}_1^k+\nabla\mathbf{u}\right). \tag{30}$$

Then:

$$\left(\frac{1}{\|\mathbf{v}\|}+\delta_1\right)\|\mathbf{v}\|=\delta_1\|\mathbf{b}_1^k+\nabla\mathbf{u}\|,$$

and hence,

$$\|\mathbf{v}\|=\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|-1/\delta_1. \tag{31}$$

Since $\|\mathbf{v}\|>0$, we can deduce that:

$$\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|>1/\delta_1. \tag{32}$$

By Eq. (30), the direction of $\mathbf{v}$ is parallel to $b_1^k+\nabla\mathbf{u}$, together with (31), we get:

$$\mathbf{v}=\left(\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|-1/\delta_1\right)\frac{\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}}{\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|}, \tag{33}$$

under the condition (32). If the condition (32) is not satisfied, from the above deduction, we must have $\mathbf{v}=0$. As a conclusion, the closed-form solution for $\mathbf{v}$ is:

$$\mathbf{v}=\max\left\{\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|-1/\delta_1,0\right\}\frac{\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}}{\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|}. \tag{34}$$

3.3. Solving $\mathbf{w}$

Fixing $\mathbf{u}$ and $\mathbf{v}$, the subproblem for $\mathbf{w}$ is:

$$\min_{|\mathbf{w}|=1}\left\{\int_{\Omega}|\mathbf{b}_2^k+\mathbf{u}-\mathbf{w}|^2dx\right\}.$$

It is easy to get that the solution is the projection onto $\mathcal{S}^{n-1}$:

$$\mathbf{w}=\frac{\mathbf{b}_2^k+\mathbf{u}}{\|\mathbf{b}_2^k+\mathbf{u}\|}. \tag{35}$$

Finally, we summarize Algorithm 2 as follows:

Algorithm 2:

```
• Initialization: $\mathbf{u}^0=\mathbf{f},\mathbf{b}_1^0=\mathbf{0},\mathbf{b}_2^0=\mathbf{0},\mathbf{v}^0=\nabla\mathbf{f},\mathbf{w}^0=\mathbf{f}.$
• For $k=0,1,2,\ldots,$ repeat until a stopping criterion is reached
    $\mathbf{u}^{k+1}=(\lambda-\mu\Delta+\mu)^{-1}\left(\lambda\mathbf{f}+\mu\nabla^T(\mathbf{v}^k-\mathbf{b}_1^k)+\mu(\mathbf{w}^k-\mathbf{b}_2^k)\right)$,
    $\mathbf{v}^{k+1}=\max\left\{\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|-1/\delta_1,0\right\}\frac{\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}}{\|\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}\|}$,
    $\mathbf{w}^{k+1}=\frac{\mathbf{b}_2^k+\mathbf{u}^{k+1}}{\|\mathbf{b}_2^k+\mathbf{u}^{k+1}\|}$,
    $\mathbf{b}_1^{k+1}=\mathbf{b}_1^k+\nabla\mathbf{u}^{k+1}-\mathbf{v}^{k+1}$,
    $\mathbf{b}_2^{k+1}=\mathbf{b}_2^k+\mathbf{u}^{k+1}-\mathbf{w}^{k+1}$.
• Output: $\mathbf{w}^{k+1}$.
```

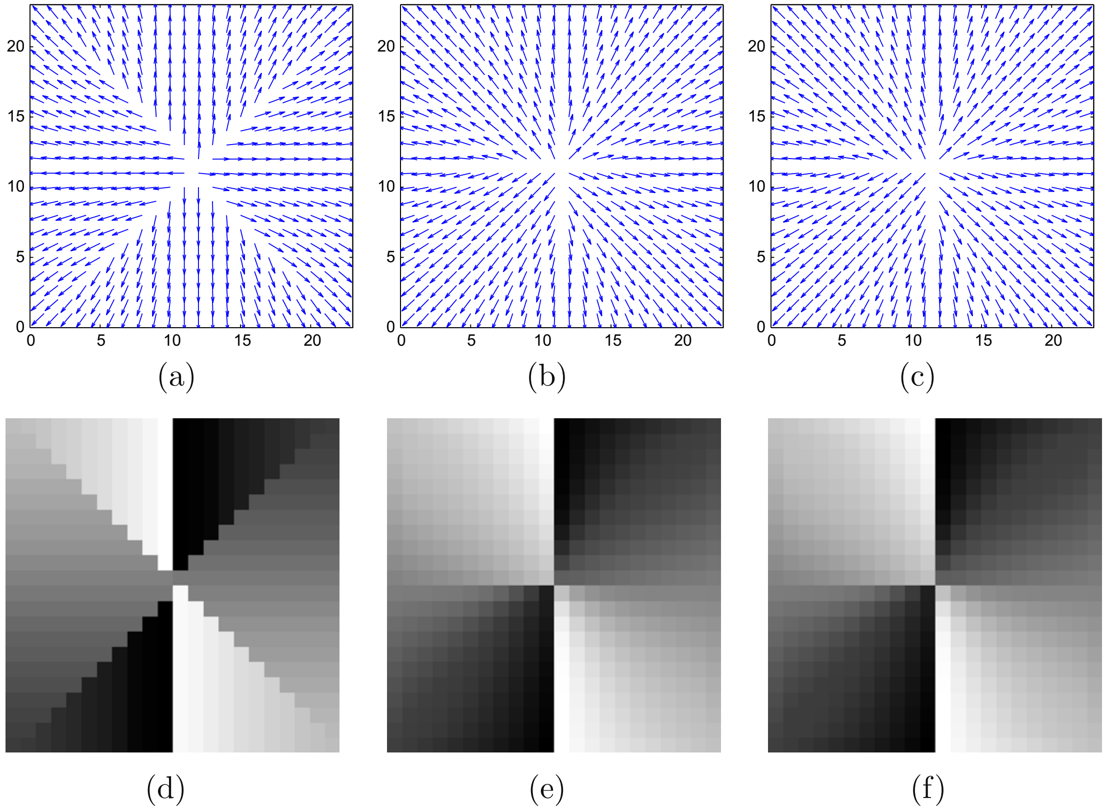


Fig. 4. S^1 data diffusion with Dirichlet boundary condition. (a) the initial data 2; (b) result by Algorithm 1 (parameters: $\lambda=0,\ \theta=1/30,\ \tau=1$); (c) result by Algorithm 2 (parameters: $\lambda=0,\ \delta_1=1,\ \delta_2=20$); (d) angle of initial data; (e) angle of result by Algorithm 1 and (f) angle of result by Algorithm 2.

4. Numerical results

In this section, we present numerical experiments in the $\mathcal{S}^1$ and $\mathcal{S}^2$ case. In the $\mathcal{S}^1$ case, we will consider Dirichlet boundary conditions and Neumann boundary conditions. We compare our algorithms with the gradient descent method and the algorithm in [32]. In the $\mathcal{S}^2$ case, we will apply our algorithms on chromaticity denoising and color image denoising problems and compare with the algorithm in [34]. Remark that in both Algorithms 1 and 2, no renormalization step is needed. The parameters are given in the figure captions. We use FFT to solve $\mathbf{u}$ in Algorithm 2 if not specified. In all the tests, the involved algorithms are stopped when max iteration is attained. In the $\mathcal{S}^1$ case, the max iteration is chosen by observing that the algorithm is converged. While in the $\mathcal{S}^2$ case, the max iteration is chosen by considering the quality of the restored image.

4.1. Results on $\mathcal{S}^1$ data

In the $\mathcal{S}^1$ case, we set the iteration to be 2000 in all figures when our algorithms are applied. The iteration is set to be large enough such that the relative error between the successive iterate of the restored image should satisfy the following inequality:

$$\frac{\|\mathbf{u}^{k+1}-\mathbf{u}^k\|_2}{\|\mathbf{u}^{k+1}\|_2} < 10^{-6}. \tag{36}$$

We consider the Dirichlet problem with boundary condition $U(x) = \frac{x-x_0}{|x-x_0|}$ on $\partial\Omega$, with $x_0 = (0.5, 0.5)$ and $\Omega = (0,1)^2$. Following [32], we consider two initial conditions. The first initial data 1 is given in Fig. 1(a) which is a noisy version of U. The random noise is added on each component of U and then renormalized to be unit vectors. Algorithms 1 and 2 give similar results in Fig. 1(b) and (c). The second row of Fig. 1 displays the angle $\alpha = \tan^{-1}(\frac{u_2}{u_1})$ corresponding to initial data, the result of Algorithms 1 and 2, respectively.

In Fig. 2, we compare our algorithms with the gradient descent method and the method in [32]. For gradient descent method, we evolve Eq. (3) with explicit difference scheme and renormalize the vector in each iteration. The results of gradient descent method are shown in the first two rows in Fig. 2. The vector fields and the angles during evolution at iterations 10,000, 40,000 and 50,000 are displayed. We observe that the gradient descent algorithm converges slowly. In the last two rows in Fig. 2, we show the results of method in [32]. The algorithm converges at about 5000 iterations. Observing the twelfth row of the vector field in Fig. 2(c), we find that the result is different from results of other algorithms. We remark that based on our experiments, we find that the algorithm in [32] is more sensitive to initialization than our proposed algorithms. Meanwhile, the unit norm constraints are not exactly satisfied by $V = (v_1, v_2)$.

In terms of computational time, Algorithm 1 converges at 2000 iterations consuming 0.49 s. Algorithm 2 converges at 2000 iterations consuming 1.05 s when FFT is used to solve $\mathbf{u}$, while it consumes 0.68 s when Gauss–Seidel iteration is used. Algorithm in [32] converges at 5000 iterations consuming 0.74 s. Gradient descent method takes 13.62 s. This comparison shows that the proposed algorithms are quite efficient.

In Fig. 3, we give the comparison of energy versus iteration in Fig. 3(a) which shows that the convergence speed of the proposed algorithms are the fastest, and the gradient descent method with renormalization is the slowest. It seems that the four algorithms converge to similar energy. Despite of the energy, we should

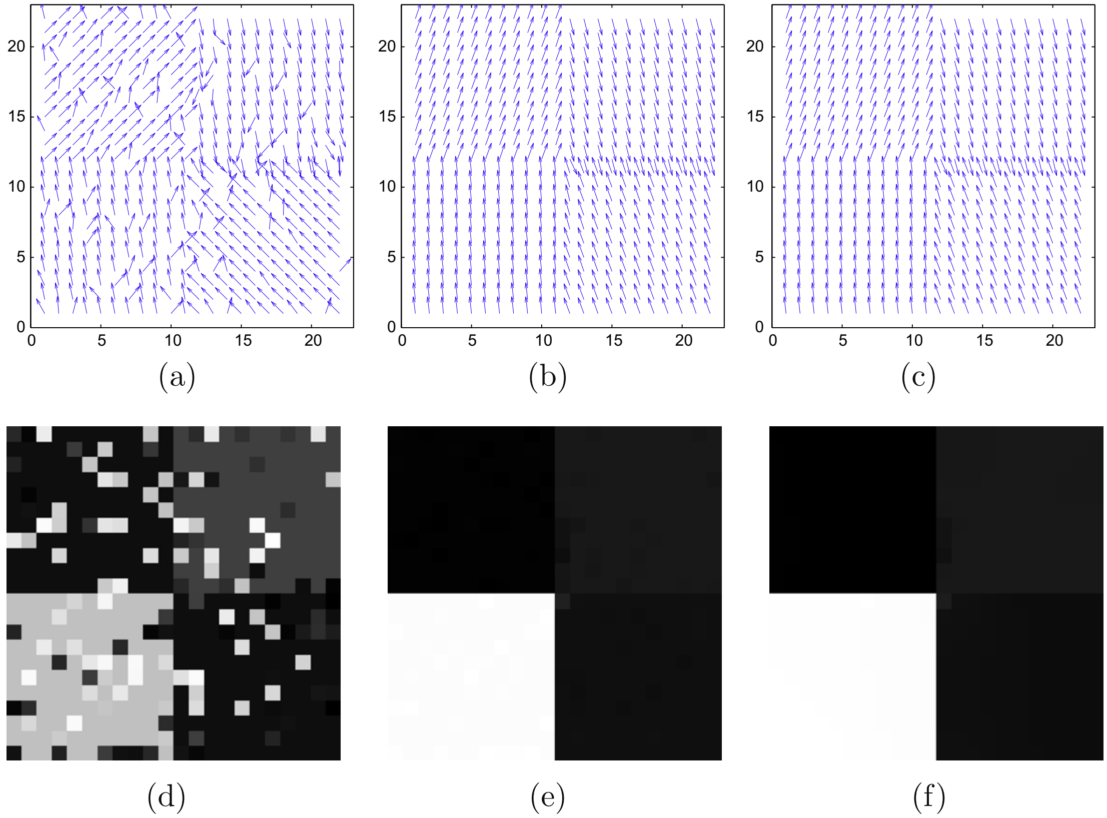


Fig. 5. S^1 data diffusion with Neumann boundary condition. (a) the initial data 3; (b) result by Algorithm 1 (parameters: $\lambda = 0.5,\ \theta = 1/20,\ \tau = 1$); (c) result by Algorithm 2 (parameters: $\lambda = 0.5,\ \delta_1 = 1,\ \delta_2 = 20$); (d) angle of initial data and (e) angle of result by Algorithm 1 and (f) angle of result by Algorithm 2.

consider that how many pixels are staying on $\mathbf{S}^1$ during evolution. For this, we calculate the norm of output vector, and find the number of vectors whose norm has an error with 1 less than 10^{-4}. Fig. 3(b) shows the number of pixels satisfy constraint versus iteration. Algorithm 2 and gradient descent method with renormalization satisfy all the constraints during iteration. Algorithm 1 can not preserve all the constraints. However, in Algorithm 1, about half of the pixels satisfy constraints which is better than algorithm in [32].

The second initial data on $\mathcal{S}^1$ is constructed like this: for $(x_1, x_2) \in \Omega$, find $(x_1^b, x_2^b) \in \partial\Omega$ as closest point to the boundary $\partial\Omega$ from (x_1, x_2). Then let $f(x_1, x_2) = U(x_1^b, x_2^b)$. This initial data is shown in Fig. 4(a). Algorithms 1 and 2 give similar results in Fig. 4(b) and Fig. 4(c). In the second row of Fig. 4, the angle $\alpha = \tan^{-1}(\frac{u_2}{u_1})$ corresponding to initial data, the results by Algorithms 1 and 2, respectively are displayed.

Remark that for initial data 1 and 2 on $\mathcal{S}^1$, for all the algorithms we set $\lambda = 0$ which means that we consider problem (2) without fidelity.

Finally we test initial data 3 with Neumann boundary condition, see Fig. 5(a). The results of Algorithms 1 and 2 are shown in Fig. 5(b) and (c). The angle corresponding to initial data, the results of Algorithms 1 and 2 are displayed in the second row of Fig. 5. The noisy vectors are effectively smoothed by both algorithms.

4.2. Results on $\mathcal{S}^2$ data

In the $\mathcal{S}^2$ case, we test three color images with different noise. The results in the following show that our proposed Algorithms 1 and 2 are efficient and effective in color image restoration.

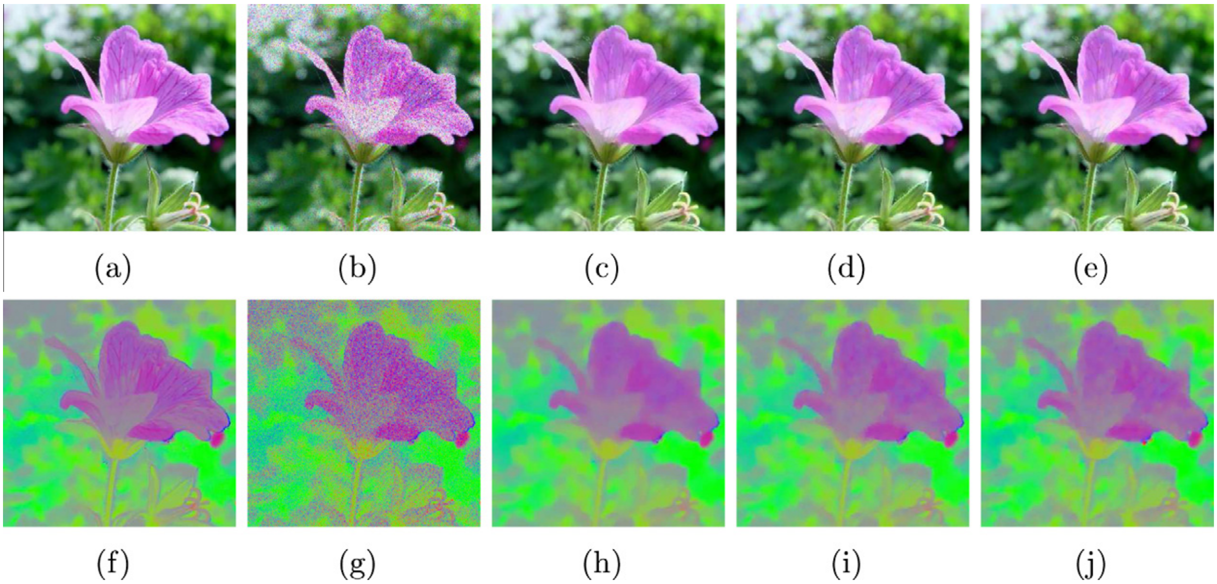


Fig. 6. $\mathcal{S}^2$ data denoising – with Gaussian noise on the chromaticity. (a) the clean color image; (b) the noisy image; (c) the recovered image by Algorithm 1, computational time = 2.9 s, iteration = 30 (parameters: $\lambda = 0.1,\ \tau = 6,\ \theta = 1/16$); (d) the recovered image by Algorithm 2, computational time = 2.1 s, iteration = 10, (parameters: $\lambda = 1,\ \delta_1 = \delta_2 = 3$); (e) the recovered image by method in [34], computational time = 3.4 s, iteration = 30; (f) the clean chromaticity of the clean image; (g) the noisy chromaticity with Gaussian noise of standard deviation 0.2, (h) the denoised chromaticity by Algorithm 1, (i) the denoised chromaticity by Algorithm 2 and (j) the denoised hromaticity by method in [34].

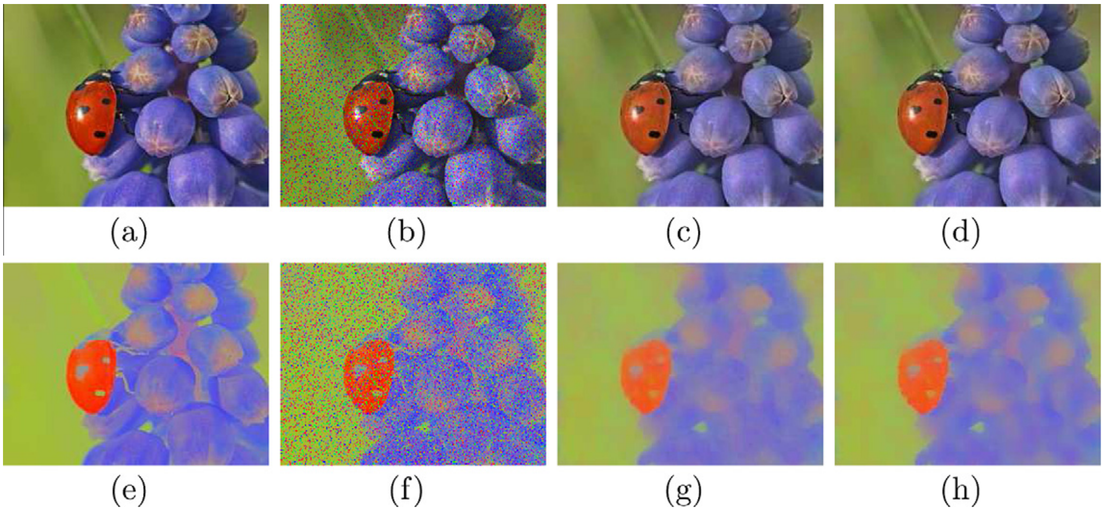


Fig. 7. $\mathcal{S}^2$ data denoising – with Salt and Pepper noise on the chromaticity. (a) the clean color image; (b) the noisy image; (c) the recovered image by Algorithm 1; (d) the recovered image by Algorithm 2; (e) the clean chromaticity of the clean image; (f) the noisy chromaticity with 20% Salt and Pepper noise; (g) the denoised chromaticity by Algorithm 1, iteration = 30 (parameters: $\lambda = 0.2,\ \tau = 3,\ \theta = 1/16$) and (g) the denoised chromaticity by Algorithm 2, iteration = 10 (parameters: $\lambda = 2.5,\ \delta_1 = \delta_2 = 3$).

In Fig. 6, we add Gaussian noise with standard deviation $\sigma = 0.2$ on the chromaticity component, meanwhile the brightness component remains unpolluted by noise. The noisy image and noisy chromaticity are displayed in Fig. 6(b) and (g), respectively. The restored images are got by multiplying the recovered chromaticity with brightness. Fig. 6(c) and (d) show the recovered images by Algorithms 1 and 2. Fig. 6(e) show the result by the curvilinear search method in [34]. The code of [34] is downloaded from <http://math.sjtu.edu.cn/faculty/zw2109/pub.html>. Fig. 6(h)–(j) show the denoised chromaticity component of Algorithms 1, 2 and method in [34], respectively. The results seems quite similar. The mean PSNR of the three methods are also similar as 31.5 ± 0.2dB. In terms of computational time, to get the displayed results, Algorithm 1 takes 2.9 s in 30 iterations, Algorithm 2 takes 2.1 s in 10 iterations, and algorithm in [34] takes 3.4 s in 30 iterations. The proposed algorithms are quite effective.

In Fig. 7, twenty percent Salt and Pepper noise is added on the chromaticity component in Fig. 7(f), meanwhile, brightness component remains unpolluted. The results in Fig. 7(c) by Algorithm 1 and Fig. 7(d) by Algorithm 2 show that our algorithms are effective to remove Salt and Pepper noise. Fig. 7(g) and (h) show the denoised chromaticity by Algorithms 1 and 2, respectively.

In Fig. 8, we add Gaussian noise on the color image, as shown in Fig. 8(b). After transformed into CB color model, both the chromaticity and the brightness components are polluted by noise, see Fig. 8(f) and (j). We use the proposed algorithms to denoise the chromaticity. For consistency, we choose the algorithm in [20] of the scalar ROF model when using Algorithm 1 on chromaticity. Meanwhile, we choose the split Bregman algorithm [17] for scalar ROF model to denoise brightness when applying Algorithm 2 on chromaticity. The recovered image is got by multiplying these two denoised components. The third column of Fig. 8 shows the results of Algorithm 1. Fig. 8(c), (g) and (k) are the recovered image, the denoised chromaticity, the denoised brightness component, respectively. The fourth column of Fig. 8 gives the results of Algorithm 2. The recovered image, the denoised chromaticity and the

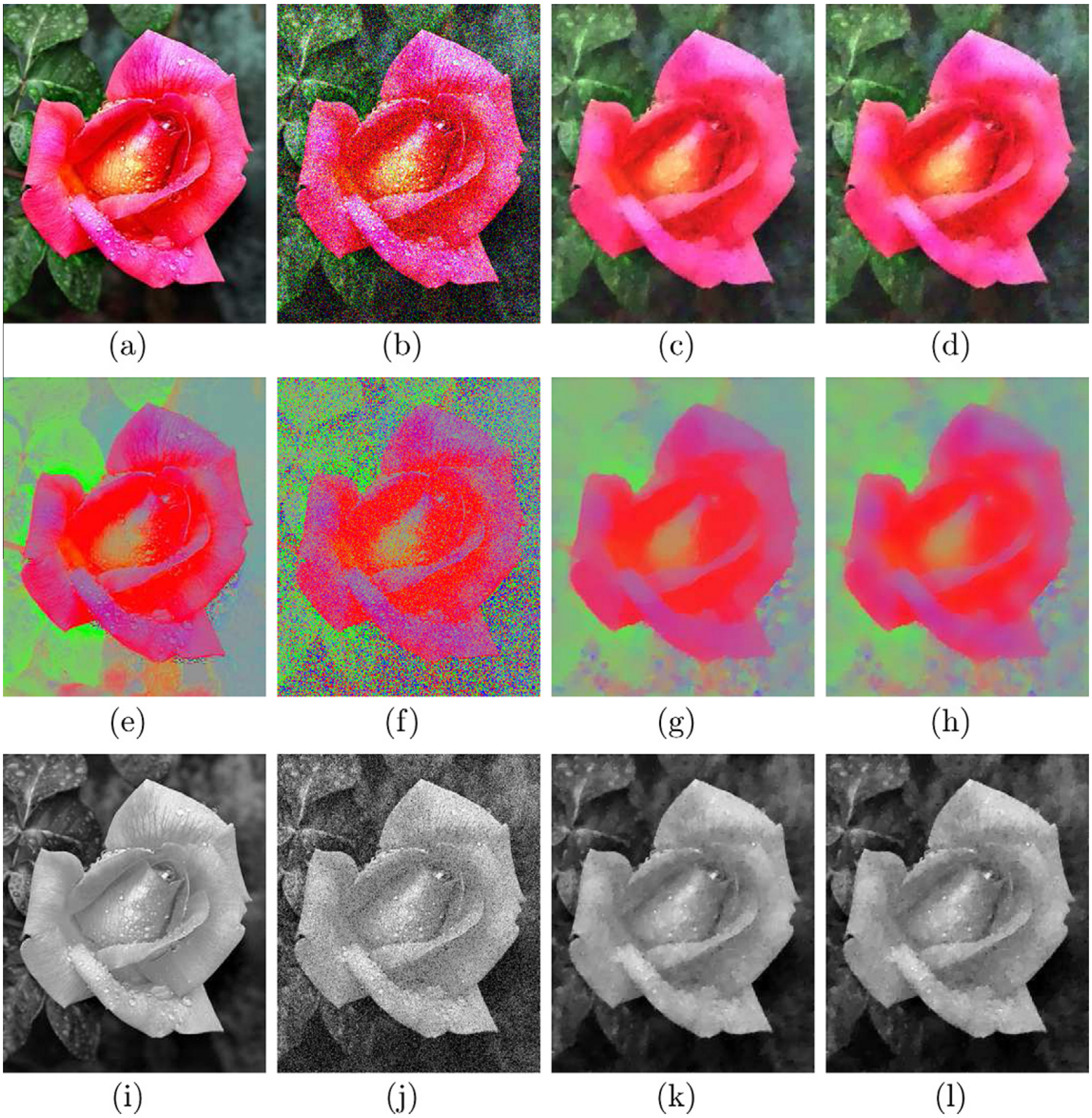


Fig. 8. $\mathcal{S}^2$ data denoising–with Gaussian noise on the color image. (a) the clean color image; (b) the noisy image with Gaussian noise of standard deviation 0.2; (c) the recovered image by Algorithm 1; (d) the recovered image by Algorithm 2; (e) the clean chromaticity of the clean image; (f) the noisy chromaticity; (g) the denoised chromaticity by Algorithm 1, iteration = 50 (parameters: $\lambda = 0.5,\ \tau = 3,\ \theta = 1/16$); (h) the denoised chromaticity by Algorithm 2, iteration = 10 (parameters: $\lambda = 2,\ \delta_1 = \delta_2 = 3$); (i) the clean brightness of the clean image; (j) the noisy brightness; (k) the denoised brightness by operator splitting algorithm of scalar ROF model in [20], iteration = 20 (parameters: $\tau = 0.007,\ \theta = 0.25$); (l) the denoised brightness by split Bregman algorithm of scalar ROF model [17], iteration = 20 (parameters: $\lambda = 0.03,\ \delta = 0.05$).

denoised brightness are displayed in Fig. 8(d), (h) and (l), respectively.

5. Conclusion

We propose two algorithms based on Lagrangian multipliers method and split Bregman method to solve ROF model constrained on $\mathcal{S}^{n-1}$. The main idea is to split the original problem into several easier subproblems by introducing auxiliary variables. The difference of the two algorithms are: Algorithm 1 satisfies the constraints approximately during iteration, while Algorithm 2 satisfies the constraints strictly. Both algorithms are easy to implement. Various experiments show that the proposed algorithms are rather effective in denoising of data constrained on S^1 or S^2, including general direction data diffusion and chromaticity denoising. Remark that the split Bregman algorithm is proved to be converge for convex problems. However, in this paper, we use the split Bregman algorithm on a nonconvex problem, such that the theoretical convergence is open. This will be our future work.

Acknowledgments

This work is supported by the 973 Program (2011CB707104), the National Science Foundation of China (11001082), and RGC 203109, 211710, 211911 and RFGs of HKBU.

References

[1] G. Aubert, P. Kornprobst, Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (Applied Mathematical Sciences), second ed., Springer-Verlag, 2006.

[2] J.-F. Aujol, Some first-order algorithms for total variation based image restoration, J. Math. Imaging Vis. 34 (3) (2009) 307–327.

[3] A. Beck, M. Teboulle, Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems, IEEE Trans. Image Process. 18 (11) (2009) 2419–2434.

[4] P.V. Blomgren, T.F. Chan, Color TV: total variation methods for restoration of vector valued images, IEEE Trans. Image Process. 7 (3) (1998) 304–309.

[5] X. Bresson, T.F. Chan, Fast dual minimization of the vectorial total variation norm and applications to color image processing, Inverse Probl. Imaging 2 (2008) 455–484.

[6] T. Cecil, L. Vese, S. Osher, Numerical methods for minimization problems constrained to S^1 and S^2, J. Comput. Phys. 198 (2004) 567–579.

[7] A. Chambolle, An algorithm for total variation minimization and applications, J. Math. Imaging Vis. 20 (1–2) (2004) 89–97.

[8] A. Chambolle, T. Pock, First-order primal–dual algorithm for convex problems with applications to imaging, J. Math. Imaging Vis. 40 (2011) 120–145.

[9] T.F. Chan, S.H. Kang, J. Shen, Total variation denoising and enhancement of color images based on the CB and HSV color models, J. Vis. Commun. Image R. 12 (4) (2001) 422–435.

[10] T. Chan, J. Shen, Variational restoration of nonflat image features: model and algorithms, SIAM J. Appl. Math. 61 (4) (2000) 1338–1361.

[11] Y.M. Chen, T. Wunderli, Adaptive total variation for image restoration in BV space, J. Math. Anal. Appl. 272 (2002) 117–137.

[12] J. Eckstein, D.P. Bertsekas, On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators, Math. Program. 55 (3) (1992) 293–318.

[13] E. Esser, X. Zhang, T.F. Chan, A general framework for a class of first order primal–dual algorithms for convex optimization in imaging science, SIAM J. Imaging Sci. 3 (2010) 1015–1046.

[14] V. Estellers, D. Zosso, R. Lai, J.-P. Thiran, S. Osher, X. Bresson, An Efficient Algorithm for Level Set Method Preserving Distance Function, UCLA CAM reports 11–58.

[15] D. Gabay, Applications of the method of multipliers to variational inequalities, in: M. Fortin, R. Glowinski (Eds.), Studies in Mathematics and its Applications, Augmented Lagrangian methods: Applications to the Numerical Solution of Boundary Value Problems, vol. 15, Amsterdam, North-Holland, 1983, pp. 299–331.

[16] G. Gilboa, N. Sochen, Y.Y. Zeevi, Variational denoising of partly textured images by spatially varying constraints, IEEE Trans. Image Process. 15 (8) (2006) 2281–2289.

[17] T. Goldstein, S. Osher, The split Bregman method for L1 regularized problems, SIAM J. Imaging Sci. 2 (2) (2009) 323–343.

[18] J. Haehnle, A. Prohl, Mumford–Shah–Euler flow with sphere constraint and applications to colour image inpainting, SIAM J. Imaging Sci. 4 (2011) 1200–1233.

[19] F. Li, C. Shen, J. Fan, C. Shen, Image restoration combining a total variational filter and a fourth-order filter, J. Vis. Commun. Image R. 18 (4) (2007) 322–330.

[20] F. Li, C. Shen, C. Li, multiphase soft segmentation with total variation and H^1 regularization, J. Math. Imaging Vis. 37 (2010) 98–111.

[21] F. Li, Z. Bao, R. Liu, G. Zhang, Fast image inpainting and colorization by Chambolle's dual method, J. Vis. Commun. Image R. 22 (6) (2011) 529–542.

[22] P.L. Lions, B. Mercier, Splitting algorithms for the sum of two nonlinear operators, SIAM J. Numer. Anal. 16 (6) (1979) 964–979.

[23] R.-Q. Jia, H. Zhao, A fast algorithm for the total variation model of image denoising, Adv. Comput. Math. 33 (2) (2010) 231–241.

[24] M. Ng, W. Fan, X. Yuan, Inexact alternating direction methods for image recovery, SIAM J. Sci. Comput. 33 (4) (2011) 1643–1668.

[25] Y. Nesterov, Smooth minimization of non-smooth functions, Math. Program. 103 (1) (2005) 127–152.

[26] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D 60 (1992) 259–268.

[27] S. Setzer, Operator splittings bregman methods and frame shrinkage in image processing, Int. J. Comput. Vis. 92 (2011) 265–280.

[28] X.-C. Tai, J. Hahn, G.J. Chung, A fast algorithm for Euler's elastica model using augmented Lagrangian method, SIAM J. Imaging Sci. 4 (1) (2011) 313–344.

[29] B. Tang, G. Sapiro, V. Caselles, Color image enhancement via chromaticity diffusion, IEEE Trans. Image Process. 10 (5) (2001) 701–707.

[30] B. Tang, G. Sapiro, V. Caselles, Diffusion of general data on non-flat manifolds via harmonic maps theory: the direction diffusion case, Int. J. Comput. Vis. 36 (2) (2000) 149–161.

[31] D. Tschumperl, R. Deriche, Orthonormal vector sets regularization with PDE's and applications, Int. J. Comput. Vis. 50 (3) (2002) 237–252.

[32] L. Vese, S. Osher, Numerical methods for p-harmonic flows and applications to image processing, SIAM J. Numer. Anal. 40 (6) (2002) 2085–2104.

[33] Y. Wang, J. Yang, W. Yin, Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, SIAM J. Imaging Sci. 1 (3) (2008) 248–272.

[34] D. Goldfarb, Z. Wen, W. Yin, A curvilinear search method for the p-Harmonic flow on sphere, SIAM J. Imaging Sci. 2 (1) (2009) 84–109.

[35] Z. Wen, W. Yin, A feasible method for optimization with orthogonality constraints, preprint: <http://math.sjtu.edu.cn/faculty/zw2109/paper/OptManifold.pdf>.