INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 7, Number 4, Pages 700–717

A FAST ALGORITHM FOR VECTORIAL TV-BASED IMAGE RESTORATION

FANGFANG DONG, JIANBIN YANG, AND CHUNXIAO LIU

Abstract. In this paper, we first extend a simple algorithm proposed by Jia et al. [16] to color/vectorial images, and then apply the vectorial algorithm to some variational models for image restoration problems including color image denoisings with the red-green-blue (RGB) and chromaticity-brightness (CB) color representations, CB based colorization and image inpainting. The variational models are all total variation (TV)-based. The proposed vectorial algorithm is simple and straightforward to implement. Some numerical experiments show that it is fast and efficient.

Key Words. Image restoration, vectorial TV model, split Bregman iteration, color denoising, CB based colorization, TV-based inpainting.

1. Introduction

Image restoration is an important research field in image processing. It is often considered as a pre-processing step for other image tasks such as image segmentation, image registration and so on. Image restoration includes many aspects, for example denoising, deblurring, inpainting, colorization, etc.

Over the past twenty years, total variation(TV)-based models proposed firstly by Rudin, Osher, and Fatemi in [23] for gray image denoising have become very popular. They have had very good applications in image denoising [1, 9, 8, 21], deblurring [12, 15], inpainting [10, 19, 11, 24], colorization [18], and so forth. There have been a lot of methods to solve these TV-based models like standard regularized approach [23, 1], primal-dual method [7], duality based method [5], split Bregman method [14], recent augmented Lagrangian method [25, 26, 27], etc. The classical algorithms (standard regularized approach or explicit gradient descent flow) often need to solve discrete Euler-Lagrange equations [23, 1, 2, 13], whose computational speed is very slow due to the regularization process of the TV-norm. Later, Chambolle [5] proposed a fast algorithm based on the dual formulation of TV-norm, which avoided the regularization of TV-norm and hence speeded up the computation dramatically. Recently, Goldstein and Osher [14] gave a novel algorithm called "split Bregman" method to solve these TV-based models. The key of their method is that they de-coupled the ℓ_1 and ℓ_2 portions of TV model and transformed the ℓ_1 regularized term to compressed sensing (CS) problems, which can be fast solved by the Bregman iteration and shrinkage. The convergence of the split Bregman iteration was shown in [14, 17] under the assumption that the resulting subproblem is solved exactly. Cai et al. [4] had also proven that the alternating split Bregman iterations are convergent when the number of inner iterations is fixed to be one.

Received by the editors August 8, 2009 and, in revised form, April 23, 2010.

2000 *Mathematics Subject Classification.* 65K10, 68U10, 49M30.

This research was partially supported by NNSF of China under Grant number 10971190 and 10771190.

However, these iteration schemes [14, 17, 4] still require solving a partial differential equation in each iteration step. The augmented Lagrangian method [25, 26, 27] was presented to solve TV models via a splitting technique and the Lagrange multiplier method. Some subproblems can be efficiently solved by shrinkage and fast Fourier transformation (FFT) implementation, where the FFT technique is used for solving a differential equation, and thus cuts down the computational time. But it is still less efficient than the direct closed form solution. More recently, Jia and Zhao [16] proposed a fast and simple algorithm to solve the Rudin-Osher-Fatemi (ROF) model/ TV denoising model. Their algorithm did not include any partial differential equations and had very simple iteration steps, which saved more computational time. What's more, they also gave a rigorous proof of the convergence of their algorithm.

In this paper, we extend Jia and Zhao's algorithm [16] to vectorial TV model, and then apply it to vector-valued image restoration problems such as a color image restoration. Here, we mainly focus on three restoration problems: color denoisings by vectorial TV-based denoising models in the red-green-blue(RGB) and chromaticity-brightness(CB) representations of color images [1, 9, 8]; image colorization based on CB color model [18]; and image inpainting [10, 19] by TV-based model for gray and color images. The proposed algorithm has several advantages. First, it has a very simple form, which will be favorable to making code. Then, the number of iterations to reach the solution is low, which gives a fast algorithm. Finally, the algorithm converges to the solution of the original vectorial TV minimization problem if appropriate parameters are chosen.

This paper is organized as follows. In Section 2, we introduce some notations and extend Jia and Zhao's algorithm to vector-valued functions so that the speed of the vectorial image processing is faster. The applications of the algorithm to image restoration including color image denoisings based on RGB and CB color representations, CB-based colorization and inpainting for gray and color images are shown in Section 3. At last, in Section 4, we present a brief conclusion.

2. Proposed algorithm for vectorial TV minimization

2.1. Notations. As in [17], we adopt the discrete form of the vectorial TV model. Let us consider a q-dimensional/channel image $\mathbf{u}$ defined on a rectangular domain Ω as follows:

$$\begin{aligned} \mathbf{u}:\Omega &\rightarrow \mathbb{R}^q, \\ (x,y) &\rightarrow \mathbf{u}(x,y)=\big(u_1(x,y),u_2(x,y),\cdots,u_q(x,y)\big). \end{aligned}$$

Discretizing the image domain Ω to some grid points, then

$$\begin{aligned} \mathbf{u}:\{1,\cdots,M\}\times\{1,\cdots,N\} &\rightarrow \mathbb{R}^q, \\ (m,n) &\rightarrow \big(u_1(m,n),u_2(m,n),\cdots,u_q(m,n)\big), \end{aligned}$$

where $M,N\geq 2$ and $q\geq 1$.

When $q=1$, the image is scalar; otherwise, the image is vector-valued.

We shall use the following norm and inner product notations:

$$\|\mathbf{u}\|_p := \Bigg(\sum_{\substack{1\leq m\leq M\\ 1\leq n\leq N}} |\mathbf{u}(m,n)|^p\Bigg)^{1/p}, \quad \text{for} \quad 1\leq p<\infty,$$

$$\big\langle \mathbf{u}(m,n),\mathbf{v}(m,n)\big\rangle := \sum_{i=1}^{q} u_i(m,n)\cdot v_i(m,n),$$

$$|\mathbf{u}(m,n)| := \sqrt{\big\langle \mathbf{u}(m,n), \mathbf{u}(m,n)\big\rangle} = \sqrt{\sum_{i=1}^{q} \big(u_i(m,n)\big)^2}.$$

The TV norm of a vector-valued function $\mathbf{u}$ is represented by:

$$\begin{aligned} \|\nabla \mathbf{u}\|_1 : &= \sum_{\substack{1\le m\le M\\ 1\le n\le N}} |\nabla \mathbf{u}(m,n)| = \sum_{\substack{1\le m\le M\\ 1\le n\le N}} \sqrt{\sum_{i=1}^{q} \big\langle \nabla u_i(m,n), \nabla u_i(m,n)\big\rangle} \\ &= \sum_{\substack{1\le m\le M\\ 1\le n\le N}} \sqrt{\sum_{i=1}^{q} \big(\nabla_x u_i(m,n)\big)^2 + \big(\nabla_y u_i(m,n)\big)^2}, \end{aligned}$$

where ∇_x and ∇_y denote the difference operators in the x-direction and y-direction, which are given by $\nabla_x u_i(1,n) = 0$, for $n = 1, \cdots, N$,

$$\nabla_x u_i(m,n) = u_i(m,n) - u_i(m-1,n), \quad m = 2, \cdots, M; n = 1, \cdots, N,$$

and $\nabla_y u_i(m,1) = 0$, for $m = 1, \cdots, M$,

$$\nabla_y u_i(m,n) = u_i(m,n) - u_i(m,n-1), \quad n = 2, \cdots, N; m = 1, \cdots, M.$$

For the operators ∇_x and ∇_y, we define their conjugate operators as follows:

$$\nabla_x^T \omega(m,n) := \begin{cases} -\omega(2,n) & \text{if } \ m = 1, \\ \omega(m,n) - \omega(m+1,n) & \text{if } \ m = 2, \cdots, M-1, \\ \omega(M,n) & \text{if } \ m = M. \end{cases}$$

$$\nabla_y^T \omega(m,n) := \begin{cases} -\omega(m,2) & \text{if } \ n = 1, \\ \omega(m,n) - \omega(m,n+1) & \text{if } \ n = 2, \cdots, N-1, \\ \omega(m,N) & \text{if } \ n = N. \end{cases}$$

Then, the discrete Laplace operator can be given by:

$$\Delta := -\nabla_x^T \nabla_x - \nabla_y^T \nabla_y.$$

Finally, the vectorial TV denoising model can be represented as the following minimization problem:

$$\min_{\mathbf{u}} \|\nabla \mathbf{u}\|_1 + \frac{\lambda}{2}\|\mathbf{u} - \mathbf{f}\|_2^2, \tag{1}$$

where the vectorial gradient of $\mathbf{u}$ is defined as $\nabla \mathbf{u} = (\nabla u_1, \cdots, \nabla u_q)$, $\mathbf{f}$ is the observed data with noise, and $\lambda > 0$ is a weighted parameter balancing the regularization term and the fidelity term.

When $q = 1$, the minimization problem (1) is the ROF model for gray image denoising proposed by Rudin, Osher, and Fatemi in [23], which is one of the most influential variational and PDE-based image denoising models in image processing. This denoising model removes noise in gray-scale images while preserving main features such as edges. Chambolle and Lions [6] had proven that the ROF model is well-posed. In [1], Blomgren and Chan extended the ROF model to vector-valued functions and presented the vectorial TV model. Bresson and Chan [3] had also given the existence and uniqueness of the solution of the vectorial TV model. In this paper, we mainly focus on the case of $q \ge 2$.

2.2. The proposed algorithm for vectorial TV model. In this section, we propose a fast algorithm to solve the minimization problem (1) for $q \geq 2$, which is considered as the vectorial extension of Jia and Zhao's algorithm [16]. In fact, the new algorithm is based on the split Bregman iteration [14]. So before proposing our algorithm, we give a brief overview of the split Bregman iteration. The split Bregman method [14] makes use of a splitting technique and the Bregman iteration to solve the TV minimization problem. Goldstein and Osher transformed the TV-minimization to constrained ℓ_1-problem or CS problem by adding an auxiliary variable, which avoids the regularization of TV-norm and can be fast solved by the Bregman iteration and shrinkage. Hence, it saves much computational time. Here, the Bregman iteration, which was first used in image processing by Osher et al. [22] for ROF/ TV denoising model, is a very popular approach to solve constrained optimization problems such as CS problems and constrained TV-based models. Its solution satisfies the constraint condition to a high degree of accuracy. In the following, we present the split Bregman iteration for vector-valued function $\mathbf{u}$.

Let $\mathbf{v} := (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_q) : \Omega \to \mathbb{R}^{q\times 2}$ be an auxiliary variable such that $\mathbf{v} = \nabla \mathbf{u}$, where $\mathbf{v}_i := (v_{ix}, v_{iy}) : \Omega \to \mathbb{R}^2$,

$$|\mathbf{v}(m,n)| = \sqrt{\langle \mathbf{v}(m,n), \mathbf{v}(m,n)\rangle} = \sqrt{\sum_{i=1}^{q} \big(v_{ix}(m,n)\big)^2 + \big(v_{iy}(m,n)\big)^2}$$

and $\mathbf{b} := (\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_q) : \Omega \to \mathbb{R}^{q\times 2}$, $\mathbf{b}_i := (b_{ix}, b_{iy}) : \Omega \to \mathbb{R}^2$.

The split Bregman iteration for vector-valued function: set $b_{ix}^0 = b_{iy}^0 = v_{ix}^0 = v_{iy}^0 = 0$ $(i = 1, \cdots, q)$, and $\mathbf{u}^0 = \mathbf{f}$. For $k = 0, 1, \cdots$

$$\big(\mathbf{u}^{k+1}, \mathbf{v}^{k+1}\big) = \arg\min_{\mathbf{u},\mathbf{v}} \left\{ \|\mathbf{v}\|_1 + \frac{\mu}{2}\|\mathbf{v} - \nabla\mathbf{u} - \mathbf{b}^k\|_2^2 + \frac{\lambda}{2}\|\mathbf{u} - \mathbf{f}\|_2^2 \right\}, \tag{2}$$
$$b_{ix}^{k+1} = b_{ix}^k + \nabla_x u_i^{k+1} - v_{ix}^{k+1}, \quad \text{for} \quad i = 1, \cdots, q,$$
$$b_{iy}^{k+1} = b_{iy}^k + \nabla_y u_i^{k+1} - v_{iy}^{k+1}, \quad \text{for} \quad i = 1, \cdots, q,$$

where

$$\|\mathbf{v}\|_1 = \sum_{\substack{1\leq m\leq M\\ 1\leq n\leq N}} |\mathbf{v}(m,n)|,$$

$$\|\mathbf{v} - \nabla\mathbf{u} - \mathbf{b}^k\|_2^2 = \sum_{i=1}^{q} \|v_{ix} - \nabla_x u_i - b_{ix}^k\|_2^2 + \sum_{i=1}^{q} \|v_{iy} - \nabla_y u_i - b_{iy}^k\|_2^2.$$

To solve the minimization problem (2), one commonly needs to solve two subproblems about $\mathbf{u}$ and $\mathbf{v}$ iteratively:

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} \left\{ \frac{\mu}{2}\|\mathbf{v}^k - \nabla\mathbf{u} - \mathbf{b}^k\|_2^2 + \frac{\lambda}{2}\|\mathbf{u} - \mathbf{f}\|_2^2 \right\}, \tag{3}$$

$$\mathbf{v}^{k+1} = \arg\min_{\mathbf{v}} \left\{ \|\mathbf{v}\|_1 + \frac{\mu}{2}\|\mathbf{v} - \nabla\mathbf{u}^{k+1} - \mathbf{b}^k\|_2^2 \right\}. \tag{4}$$

Here, the solution of (3) is given by the following optimality condition:

$$(\lambda I - \mu\Delta)\mathbf{u} = \lambda\mathbf{f} + \mu\nabla_x^T(\mathbf{v}_x^k - \mathbf{b}_x^k) + \mu\nabla_y^T(\mathbf{v}_y^k - \mathbf{b}_y^k), \tag{5}$$

where $\Delta\mathbf{u} = (\Delta u_1, \cdots, \Delta u_q)$, $\nabla_x^T\mathbf{w} = (\nabla_x^T w_1, \cdots, \nabla_x^T w_q)$ and $\nabla_y^T\mathbf{w} = (\nabla_y^T w_1, \cdots, \nabla_y^T w_q)$.

Note that the vectorial split Bregman iterative scheme still requires solving differential equations (5) in each iteration, thus it may slow the computation. In order to improve the computation speed, we propose to extend Jia and Zhao's algorithm [16] to vector-valued functions.

The proposed algorithm: set $b_{ix}^0 = 0$, $b_{iy}^0 = 0$ $(i = 1, \cdots, q)$, and $\mathbf{u}^1 = \mathbf{f}$. For $k = 1, 2, \cdots$, let

$$w_{ix}^k = \nabla_x u_i^k + b_{ix}^{k-1},\ w_{iy}^k = \nabla_y u_i^k + b_{iy}^{k-1}, \quad \text{for} \quad i = 1, \cdots, q, \tag{6}$$

$$t^k = \sqrt{\sum_{i=1}^{q} (w_{ix}^k)^2 + (w_{iy}^k)^2},\ s^k = \max(\mu t^k, 1), \tag{7}$$

$$b_{ix}^k = w_{ix}^k / s^k,\ b_{iy}^k = w_{iy}^k / s^k, \quad \text{for} \quad i = 1, \cdots, q, \tag{8}$$

$$u_i^{k+1} = f_i - \frac{\mu}{\lambda}(\nabla_x^T b_{ix}^k + \nabla_y^T b_{iy}^k), \quad \text{for} \quad i = 1, \cdots, q. \tag{9}$$

The proposed algorithm has very simple iteration steps, which will be favorable to making code. Furthermore, in terms of the later experiments, the number of iterations to reach the solution is low, which gives a fast algorithm. In the following, we give a simple deduction of the proposed algorithm.

Following [16], we demonstrate that the iterative scheme (6)-(9) is equivalent to the following algorithm: set $b_{ix}^0 = b_{iy}^0 = 0$, $v_{ix}^0 = v_{iy}^0 = 0 (i = 1, \cdots, q)$, and $\mathbf{u}^1 = \mathbf{f}$. For $k = 1, 2, \cdots$, let

$$\mathbf{v}^k = \arg\min_{\mathbf{v}} \left\{ \|\mathbf{v}\|_1 + \frac{\mu}{2}\|\mathbf{v} - \nabla \mathbf{u}^k - \mathbf{b}^{k-1}\|_2^2 \right\}, \tag{10}$$

$$b_{ix}^k = b_{ix}^{k-1} + \nabla_x u_i^k - v_{ix}^k, \quad \text{for} \quad i = 1, \cdots, q, \tag{11}$$

$$b_{iy}^k = b_{iy}^{k-1} + \nabla_y u_i^k - v_{iy}^k, \quad \text{for} \quad i = 1, \cdots, q, \tag{12}$$

and

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} \left\{ \frac{1}{2}\|B(\mathbf{u} - \mathbf{f})\|_2^2 - \langle B^2(\mathbf{u}^k - \mathbf{f}), \mathbf{u} - \mathbf{u}^k \rangle + \frac{\mu}{2}\|\mathbf{v}^k - \nabla \mathbf{u}\|_2^2 \right\}, \tag{13}$$

where $B^2\mathbf{u} := \big((\lambda + \mu\Delta)u_1, \cdots, (\lambda + \mu\Delta)u_q\big)$, and the operator $\lambda + \mu\Delta$ is positive definite if $0 < \mu/\lambda < 1/8$.

Note that the minimization problem (10) with respect to $\mathbf{v}$ and the iterative relationships of b_{ix} and b_{iy} (11)-(12) are the same with those in the split Bregman iteration, but the proposed iterative scheme (6)-(9) derived from the equivalent algorithm does not involve any difference equations.

Firstly, we deduce the iterative expressions of b_{ix} and b_{iy}, i.e., (8).

Following [28] (Lemma 3.3), we get the solution of minimization problem (10) by vectorial shrinkage:

$$\mathbf{v}^k = \max\left(|\nabla \mathbf{u}^k + \mathbf{b}^{k-1}| - \frac{1}{\mu}, 0 \right) \frac{\nabla \mathbf{u}^k + \mathbf{b}^{k-1}}{|\nabla \mathbf{u}^k + \mathbf{b}^{k-1}|},$$

where

$$|\nabla \mathbf{u}^k + \mathbf{b}^{k-1}| = \sqrt{\sum_{i=1}^{q} (\nabla_x u_i^k + b_{ix}^{k-1})^2 + (\nabla_y u_i^k + b_{iy}^{k-1})^2}.$$

For each component of $\mathbf{v}$, we have

$$\begin{aligned} v_{ix}^k &= \max\left(|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|-\frac{1}{\mu},0\right)\frac{\nabla_x u_i^k+b_{ix}^{k-1}}{|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|},\\ v_{iy}^k &= \max\left(|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|-\frac{1}{\mu},0\right)\frac{\nabla_y u_i^k+b_{iy}^{k-1}}{|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|}. \end{aligned}$$

From (11) and (12), we get

$$\begin{aligned} b_{ix}^k &= b_{ix}^{k-1}+\nabla_x u_i^k - v_{ix}^k\\ &= b_{ix}^{k-1}+\nabla_x u_i^k - \max\left(|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|-\frac{1}{\mu},0\right)\frac{\nabla_x u_i^k+b_{ix}^{k-1}}{|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|}\\ &= \frac{b_{ix}^{k-1}+\nabla_x u_i^k}{\max(\mu|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|,1)}, \end{aligned}$$

$$\begin{aligned} b_{iy}^k &= b_{iy}^{k-1}+\nabla_y u_i^k - v_{iy}^k\\ &= b_{iy}^{k-1}+\nabla_y u_i^k - \max\left(|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|-\frac{1}{\mu},0\right)\frac{\nabla_y u_i^k+b_{iy}^{k-1}}{|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|}\\ &= \frac{b_{iy}^{k-1}+\nabla_y u_i^k}{\max(\mu|\nabla\mathbf{u}^k+\mathbf{b}^{k-1}|,1)}. \end{aligned}$$

Therefore, we obtain the iterations of b_{ix} and b_{iy} (8) for $i=1,\cdots,q$.

Secondly, we demonstrate the equality (9) is actually the solution of the minimization problem (13). Note that the minimization problem (13) is equivalent to the following ones:

$$\begin{aligned} u_i^{k+1} &= \arg\min_{u_i}\left\{\frac{1}{2}\|B_i(u_i-f_i)\|_2^2-\langle B_i^2(u_i^k-f_i),u_i-u_i^k\rangle\right.\\ &+ \left.\frac{\mu}{2}\left(\|v_{ix}^k-\nabla_x u_i\|_2^2+\|v_{iy}^k-\nabla_y u_i\|_2^2\right)\right\}, \quad \text{for} \quad i=1,\cdots,q, \end{aligned}$$

where $B_i^2=\lambda+\mu\Delta$. Each one of the above minimization problems is the same as that in [16]. Thus, from Lemma 2 in [16], we can obtain the solution of (13), i.e., (9).

Finally, we prove the following theorem.

Theorem. *The equivalent algorithm (10)-(13), and hence the proposed algorithm (6)-(9) converges to the solution of original minimization problem (1) if* $0<\mu/\lambda<1/8$.

Proof. Let $F(\mathbf{u})=\frac{\lambda}{2}\|\mathbf{u}-\mathbf{f}\|_2^2$, $G(\mathbf{v})=\|\mathbf{v}\|_1$ and $H(\mathbf{u},\mathbf{v},\mathbf{b})=\frac{\mu}{2}\|\mathbf{v}-\nabla\mathbf{u}-\mathbf{b}\|_2^2$, then the problem (10) can be rewritten as

$$\mathbf{v}^k=\arg\min_{\mathbf{v}}\left\{G(\mathbf{v})+H(\mathbf{u}^k,\mathbf{v},\mathbf{b}^{k-1})\right\},$$

Note that $-\mu(\mathbf{v}^k-\nabla\mathbf{u}^k-\mathbf{b}^{k-1})\in\partial G(\mathbf{v}^k)$, where ∂ denotes the subdifferential symbol, and from the equivalent algorithm (11) and (12), we know that

$$\mathbf{b}^k=\mathbf{b}^{k-1}+\nabla\mathbf{u}^k-\mathbf{v}^k.$$

Then,

$$\mu\mathbf{b}^k\in\partial G(\mathbf{v}^k).$$

On the other hand,

$$\partial F(\mathbf{u})=\lambda(\mathbf{u}-\mathbf{f}).$$

According to the definition of the subgradient, for any $\mathbf{w} \in \mathbb{R}^q$, we have

$$\|\mathbf{v}^k + \nabla \mathbf{w}\|_1 - \|\mathbf{v}^k\|_1 - \langle \mu \mathbf{b}^k, \nabla \mathbf{w} \rangle \geq 0, \tag{14}$$

and

$$F(\mathbf{u}^{k+1} + \mathbf{w}) - F(\mathbf{u}^{k+1}) - \langle \lambda(\mathbf{u}^{k+1} - \mathbf{f}), \mathbf{w} \rangle \geq 0.$$

From (9), the above inequality is equivalent to

$$F(\mathbf{u}^{k+1} + \mathbf{w}) - F(\mathbf{u}^{k+1}) + \langle \mu \mathbf{b}^k, \nabla \mathbf{w} \rangle \geq 0. \tag{15}$$

Adding (14) and (15), we obtain

$$\|\mathbf{v}^k\|_1 + F(\mathbf{u}^{k+1}) \leq \|\mathbf{v}^k + \nabla \mathbf{w}\|_1 + F(\mathbf{u}^{k+1} + \mathbf{w}). \tag{16}$$

Then, similar to the proof given by Jia and Zhao in [16], we can prove that there exists a convergent subsequence $\{\mathbf{u}^{k_j}\}$ of $\{\mathbf{u}^k\}$ and $\tilde{\mathbf{u}}$, such that

$$\lim_{j \to \infty} \mathbf{u}^{k_j+1} = \lim_{j \to \infty} \mathbf{u}^{k_j} = \tilde{\mathbf{u}}.$$

Moreover,

$$\lim_{j \to \infty} \mathbf{v}^{k_j} = \nabla \tilde{\mathbf{u}}.$$

Replacing k by k_j in (16) and letting $j \to \infty$, we get

$$\|\nabla \tilde{\mathbf{u}}\|_1 + F(\tilde{\mathbf{u}}) \leq \|\nabla \tilde{\mathbf{u}} + \nabla \mathbf{w}\|_1 + F(\tilde{\mathbf{u}} + \mathbf{w}).$$

That is to say, $\tilde{\mathbf{u}}$ is a solution of the vectorial TV model. Since the solution of the vectorial TV model is unique [3], then we have

$$\tilde{\mathbf{u}} = \mathbf{u}^\star,$$

if we suppose $\mathbf{u}^\star$ is the unique solution. Furthermore, $\lim_{j \to \infty} \mathbf{u}^{k_j} = \mathbf{u}^\star$. □

Remark. *The proof is a little different from that in Jia and Zhao's paper* [16] *because our problem is the vectorial isotropic TV model. For scalar anisotropic TV model, the corresponding theorem and proof were given by Jia and Zhao* [16].

For $q = 3$, we can apply the algorithm to color image restoration. In the following section, we will show the applications of the algorithm to color denoisings with the RGB and CB color representations, CB-based colorization and image inpainting by vectorial TV-based models.

3. Applications to vectorial TV-based models for image restoration

In this section, we first apply the proposed algorithm and its variants to color image denoisings with the RGB and CB color representations by vectorial TV-based models in Subsection 3.1; then the application to CB-based colorization will be presented in Subsection 3.2; finally, in Subsection 3.3, we will show the algorithm for TV-based inpainting. We want to mention that all experiments in this paper are implemented by Matlab on an Intel 1.66 GHz computer with Intel Core 2 Duo processor.

3.1. Color image denoising. In this section, we apply the algorithm to the vectorial TV-based denoising models for color images in the RGB and CB representations [1, 8]. The RGB color representation is a linear color model, and the CB color representation is a nonlinear one. Each one of the two color models has its advantages. The classical processing models based on RGB color model are easy to work with, while the CB-based models show better color control. Here, we mainly focus on the applications of the proposed algorithm rather than the comparison between the different color models.

In the RGB representation, a color image is a mapping:

$$\mathbf{u} : \{1, \cdots, M\} \times \{1, \cdots, N\} \to (u_1, u_2, u_3),$$

where u_1, u_2 and u_3 represent color intensities for the three channels (red, green and blue) of the color image $\mathbf{u}$.

The vectorial TV model for this color representation, which had been introduced in [1], is exactly the minimization problem (1) if $q = 3$. Consequently, we can directly apply the iterative scheme (6)-(9) to the RGB-based TV denoising model. The results are presented in Figure 1. We compare the split Bregman method proposed by Goldstein and Osher [14] and the vectorial dual method proposed by Bresson et al. [3] with our algorithm. It shows that our algorithm is faster and the ratio of signal to noise (SNR) is higher for color TV denoising. In this paper, the SNR is defined by the following formula:

$$SNR = \frac{\|\mathbf{u} - \bar{\mathbf{u}}\|_2^2}{\|\boldsymbol{\eta} - \bar{\boldsymbol{\eta}}\|_2^2},$$

where $\mathbf{u}$ denotes the clean image, $\boldsymbol{\eta}$ denotes the noise, and $\bar{\mathbf{u}}$ and $\bar{\boldsymbol{\eta}}$ denote the means of the clean image and noise.

In the CB representation, $\mathbf{u}$ can be separated into the brightness component $B := |\mathbf{u}| = \sqrt{(u_1)^2 + (u_2)^2 + (u_3)^2}$, and the chromaticity component $\mathbf{C} := \mathbf{u}/|\mathbf{u}| = \mathbf{u}/B = (C_1, C_2, C_3)$. The brightness B can be treated as a gray image; thus any scalar denoising model can be applied, for example scalar TV model. However, the chromaticity component $\mathbf{C}$ stores the color information, and takes values on the unit sphere S^2, i.e., $|\mathbf{C}(m, n)| = 1$ (for keeping consistency with some relevant references [3, 18, 20], we simply write it as $|\mathbf{C}| = 1$); we need to apply the vectorial TV-based methods for its denoising. After denoising brightness and chromaticity components separately, we assemble the two components to get the restored image:

$$\mathbf{u} = B \times \mathbf{C}.$$

Therefore, in the CB color representation, we have the following denoising models:

$$\min_B \|\nabla B\|_1 + \frac{\lambda_1}{2}\|B - B_0\|_2^2, \tag{17}$$

$$\max_\alpha \min_{\mathbf{C}} \|\nabla \mathbf{C}\|_1 + \frac{\lambda_2}{2}\|\mathbf{C} - \mathbf{C}_0\|_2^2 + \frac{1}{2}\langle \alpha, |\mathbf{C}|^2 - 1\rangle, \tag{18}$$

where B_0 and $\mathbf{C}_0$ are the brightness and chromaticity components of a color noisy image $\mathbf{f}$, the last term of (18) is a constrained term for $|\mathbf{C}| = 1$, and α is a Lagrange multiplier.

In order to solve (18) using the proposed algorithm, we regularize it by adding an auxiliary variable $\mathbf{U}$ as follows:

$$\max_\alpha \min_{\mathbf{C},\mathbf{U}} \|\nabla \mathbf{C}\|_1 + \frac{\lambda_2}{2}\|\mathbf{C} - \mathbf{C}_0\|_2^2 + \frac{\beta}{2}\|\mathbf{U} - \mathbf{C}\|_2^2 + \frac{1}{2}\langle \alpha, |\mathbf{U}|^2 - 1\rangle. \tag{19}$$

Then solving (19) provides an approximation to the solution of (18) when β is sufficiently large. Since the above objective functional is convex, its solution can be obtained by minimizing it with respect to $\mathbf{C}$ and $\mathbf{U}$ and maximizing it with respect to α separately. Thus, we have two minimization subproblems and a maximization subproblem:

$$\begin{aligned} &\min_{\mathbf{C}} \|\nabla \mathbf{C}\|_1 + \frac{\lambda_2}{2}\|\mathbf{C}-\mathbf{C}_0\|_2^2 + \frac{\beta}{2}\|\mathbf{U}-\mathbf{C}\|_2^2 \\ \Leftrightarrow &\min_{\mathbf{C}} \|\nabla \mathbf{C}\|_1 + \frac{\lambda_2+\beta}{2}\|\mathbf{C} - \big(\frac{\lambda_2}{\lambda_2+\beta}\mathbf{C}_0 + \frac{\beta}{\lambda_2+\beta}\mathbf{U}\big)\|_2^2, \end{aligned} \tag{20}$$

$$\min_{\mathbf{U}} \frac{\beta}{2}\|\mathbf{U}-\mathbf{C}\|_2^2 + \frac{1}{2}\langle \alpha, |\mathbf{U}|^2 - 1\rangle, \tag{21}$$

and

$$\max_{\alpha} \frac{1}{2}\langle \alpha, |\mathbf{U}|^2 - 1\rangle. \tag{22}$$

Obviously, for (20), we can use the algorithm proposed in Section 2.2. The rest two subproblems have closed-form solutions. By direct computation, the solution of minimization problem (21) is given by:

$$\mathbf{U} = \frac{\beta \mathbf{C}}{\alpha+\beta}. \tag{23}$$

In order to find the saddle point of the objective functional, maximizing the functional in (22) with respect to the Lagrange multiplier α gives:

$$|\mathbf{U}|^2 - 1 = 0.$$

Then making inner product on the both side of (23) with $\mathbf{U}$, we have the solution of (22):

$$\alpha = \beta\langle \mathbf{C}, \mathbf{U}\rangle - \beta.$$

In conclusion, we have the following iterative schemes:

(i) For brightness component B: set $b_x^0 = 0$, $b_y^0 = 0$, and $B^1 = B_0$. For $k = 1, 2, \cdots$, let

$$\begin{aligned} w_x^k &= \nabla_x B^k + b_x^{k-1},\ w_y^k = \nabla_y B^k + b_y^{k-1}, \\ t^k &= \sqrt{(w_x^k)^2 + (w_y^k)^2},\ s^k = \max(\mu_1 t^k, 1), \\ b_x^k &= w_x^k/s^k,\ b_y^k = w_y^k/s^k, \\ B^{k+1} &= B_0 - \frac{\mu_1}{\lambda_1}(\nabla_x^T b_x^k + \nabla_y^T b_y^k). \end{aligned}$$

(ii) For chromaticity component $\mathbf{C}$: set $b_{ix}^0 = 0$, $b_{iy}^0 = 0$ $(i = 1, 2, 3)$, $\alpha^1 = 1$ and $\mathbf{C}^1 = \mathbf{C}_0$. For $k = 1, 2, \cdots$, let

$$
\begin{aligned}
w_{1x}^k &= \nabla_x C_1^k + b_{1x}^{k-1},\ w_{1y}^k = \nabla_y C_1^k + b_{1y}^{k-1},\\
w_{2x}^k &= \nabla_x C_2^k + b_{2x}^{k-1},\ w_{2y}^k = \nabla_y C_2^k + b_{2y}^{k-1},\\
w_{3x}^k &= \nabla_x C_3^k + b_{3x}^{k-1},\ w_{3y}^k = \nabla_y C_3^k + b_{3y}^{k-1},\\
t^k &= \sqrt{\sum_{i=1}^{3} (w_{ix}^k)^2 + (w_{iy}^k)^2},\ s^k = \max(\mu_2 t^k, 1),\\
b_{1x}^k &= w_{1x}^k / s^k,\ b_{1y}^k = w_{1y}^k / s^k,\\
b_{2x}^k &= w_{2x}^k / s^k,\ b_{2y}^k = w_{2y}^k / s^k,\\
b_{3x}^k &= w_{3x}^k / s^k,\ b_{3y}^k = w_{3y}^k / s^k,\\
\mathbf{U}^k &= \frac{\beta \mathbf{C}^k}{\alpha^k + \beta},\\
\alpha^{k+1} &= \beta \langle \mathbf{C}^k, \mathbf{U}^k \rangle - \beta,\\
\mathbf{C}^{k+1} &= \frac{\lambda_2}{\lambda_2 + \beta} \mathbf{C}_0 + \frac{\beta}{\lambda_2 + \beta} \mathbf{U}^k\\
&\quad - \frac{\mu_2}{\lambda_2 + \beta} \left(\nabla_x^T b_{1x}^k + \nabla_y^T b_{1y}^k, \nabla_x^T b_{2x}^k + \nabla_y^T b_{2y}^k, \nabla_x^T b_{3x}^k + \nabla_y^T b_{3y}^k \right).
\end{aligned}
$$

The restored image is $\mathbf{u} = B \times \mathbf{C}$. Figure 2 presents the application of the proposed iterative schemes, whose speed is faster than the dual method.

3.2. TV-based colorization based on CB color model. In this section, we present an algorithm of TV-based colorization based on CB color model [18].

Let $\Omega := \{1, \cdots, M\} \times \{1, \cdots, N\}$ be the image domain and the measurable subset D denote the inpainting domain where we wish to colorize. Let $D^c = \Omega \backslash D$ be the complement of D in Ω, where the color is given. The colorization task can be understood as inpainting the colors in D. We consider the following minimization problem depending on chromaticity:

$$
\min_{\mathbf{C}, |\mathbf{C}|=1} \|\nabla \mathbf{C}\|_1 + \frac{\hat{\lambda}}{2} \|\mathbf{C} - \mathbf{C}_0\|_2^2, \tag{24}
$$

where $\mathbf{C}_0$ is the chromaticity component of a given image, and

$$
\hat{\lambda} = \begin{cases} \lambda, & (m, n) \in D^c,\\ 0, & (m, n) \in D, \end{cases}
$$

$1 \le m \le M$ and $1 \le n \le N$.

By Lagrange multiplier method, we transform the constrained problem (24) to the following unconstrained one:

$$
\max_{\alpha} \min_{\mathbf{C}} \|\nabla \mathbf{C}\|_1 + \frac{\hat{\lambda}}{2} \|\mathbf{C} - \mathbf{C}_0\|_2^2 + \frac{1}{2} \langle \alpha, |\mathbf{C}|^2 - 1 \rangle, \tag{25}
$$

where α is a variable which can be updated automatically.

Since $\hat{\lambda} = 0$ if $(m, n) \in D$, the iterative scheme (6)-(9) cannot be applied directly. By adding an auxiliary variable $\mathbf{U}$ as Li et al. [20], the above energy can be approximated by:

$$
\max_{\alpha} \min_{\mathbf{U}, \mathbf{C}} \|\nabla \mathbf{U}\|_1 + \frac{1}{2\theta} \|\mathbf{U} - \mathbf{C}\|_2^2 + \frac{\hat{\lambda}}{2} \|\mathbf{C} - \mathbf{C}_0\|_2^2 + \frac{1}{2} \langle \alpha, |\mathbf{C}|^2 - 1 \rangle, \tag{26}
$$

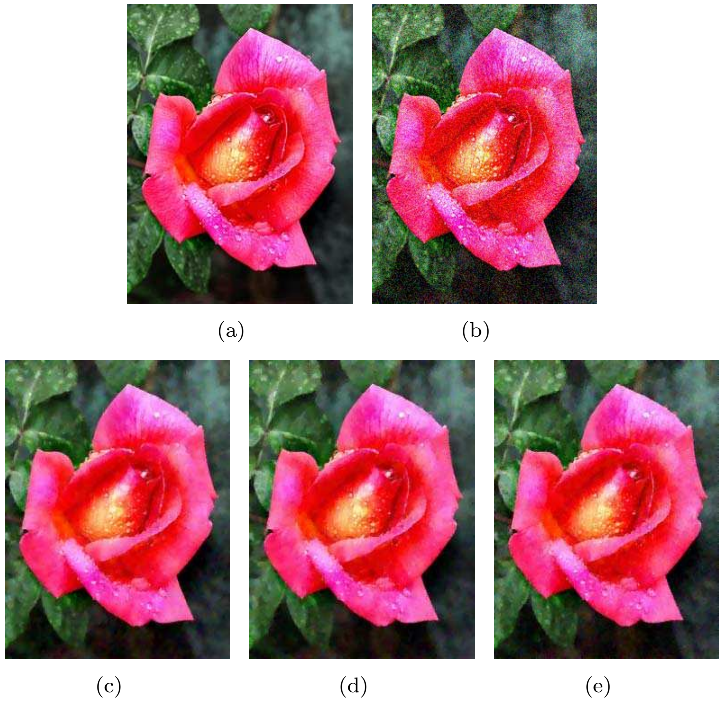

FIGURE 1. Color denoising based on RGB color model. Image size is 303 × 250. (a) Original image. (b) Noisy image (SNR=6.0751). (c) Result of proposed algorithm (SNR=41.4922), iter=10, t=0.991s; $\mu = 0.003$, $\lambda = 0.03$. (d) Result of split Bregman algorithm (SNR=41.4884), iter=20, t=1.877s. (e) Result of dual algorithm [3] (SNR=39.6822), iter=50, t=5.456s.

where θ is chosen to be small enough so that $\mathbf{U}$ is close to $\mathbf{C}$ in the sense of ℓ_2 norm.

In [20], the authors had proven that the approximate problem (26) and the original problem (24) exist minimizers respectively and the solution of the approximate problem converges to the solution of the original problem (24).

According to Lagrange multiplier method, the solution can be computed by minimizing the functional in (26) with respect to $\mathbf{U}$, $\mathbf{C}$ and maximizing it with respect to α separately, and iterating until convergence. Thus, the following three subproblems are considered:

(i) $\mathbf{C}$ and α being fixed, we search for $\mathbf{U}$ as a solution of:

$$\min_{\mathbf{U}} \|\nabla \mathbf{U}\|_1 + \frac{1}{2\theta}\|\mathbf{U} - \mathbf{C}\|_2^2, \tag{27}$$

(ii) $\mathbf{U}$ and α being fixed, we search for $\mathbf{C}$ as a solution of:

$$\min_{\mathbf{C}} \frac{1}{2\theta}\|\mathbf{U} - \mathbf{C}\|_2^2 + \frac{\hat{\lambda}}{2}\|\mathbf{C} - \mathbf{C}_0\|_2^2 + \frac{1}{2}\langle \alpha, |\mathbf{C}|^2 - 1\rangle, \tag{28}$$

(iii) $\mathbf{U}$ and $\mathbf{C}$ being fixed, we search for α as a solution of:

$$\max_{\alpha} \langle \alpha, |\mathbf{C}|^2 - 1\rangle. \tag{29}$$

FIGURE 2. Color denoising based on CB color model. Image size is 256×256. (a) Original image. (b) Noisy image (SNR=5.1574). (c) The brightness component of noisy image. (d) The chromaticity component of noisy image. (e) Denoising result of brightness component using proposed algorithm (iter=10). (f) Denoising result of chromaticity component using proposed algorithm (iter=10). (g) Denoising result of noisy image using proposed algorithm (SNR=29.8560), t=1.147s; $\mu_1 = 0.008$, $\lambda_1 = 0.065$, $\mu_2 = 2$, $\lambda_2 = 10$, $\beta = 10$. (h) Denoising result based on RGB color model using proposed algorithm (SNR=29.5145), iter=10, t=0.878s; $\mu = 0.004$, $\lambda = 0.04$. (i) Denoising result of brightness component using dual method [3] (iter=50). (j) Denoising result of chromaticity component using dual method [3] (iter=50). (k) Denoising result of noisy image using dual method [3] (SNR=28.2305), t=6.971s. (l) Denoising result based on RGB color model using dual method [3] (SNR=29.3835), iter=50, t=4.663s.

For (27), we can make use of the vectorial algorithm proposed in Section 2.2. By variational calculus of (28) with respect to $\mathbf{C}$, we have

$$\frac{1}{\theta}(\mathbf{C} - \mathbf{U}) + \hat{\lambda}(\mathbf{C} - \mathbf{C}_0) + \alpha \mathbf{C} = 0, \tag{30}$$

then, the solution of the minimization problem (28) is given by

$$\mathbf{C} = \frac{\mathbf{U} + \hat{\lambda}\theta\mathbf{C}_0}{1 + \hat{\lambda}\theta + \alpha\theta}.$$

In order to get α, variation of (29) with respect to α gives

$$|\mathbf{C}|^2 - 1 = 0.$$

Then, making inner product with $\mathbf{C}$ on the two side of (30) and using the equality $|\mathbf{C}|^2 = 1$, we get

$$\frac{1}{\theta}\big(1 - \langle \mathbf{U}, \mathbf{C}\rangle\big) + \hat{\lambda}\big(1 - \langle \mathbf{C}_0, \mathbf{C}\rangle\big) + \alpha = 0. \tag{31}$$

Hence,

$$\alpha = \frac{1}{\theta}\langle \mathbf{U}, \mathbf{C}\rangle + \hat{\lambda}\langle \mathbf{C}, \mathbf{C}_0\rangle - \frac{1}{\theta} - \hat{\lambda}.$$

To sum up, we have the following iterative algorithm: set $b_{ix}^0 = 0$, $b_{iy}^0 = 0$ $(i = 1, 2, 3)$, and $\mathbf{C}^1 = \mathbf{C}_0$, $\mathbf{U}^1 = \mathbf{C}^1$. For $k = 1, 2, \cdots$, let

$$\begin{aligned}
w_{1x}^k &= \nabla_x U_1^k + b_{1x}^{k-1},\ w_{1y}^k = \nabla_y U_1^k + b_{1y}^{k-1},\\
w_{2x}^k &= \nabla_x U_2^k + b_{2x}^{k-1},\ w_{2y}^k = \nabla_y U_2^k + b_{2y}^{k-1},\\
w_{3x}^k &= \nabla_x U_3^k + b_{3x}^{k-1},\ w_{3y}^k = \nabla_y U_3^k + b_{3y}^{k-1},\\
t^k &= \sqrt{\sum_{i=1}^{3}(w_{ix}^k)^2 + (w_{iy}^k)^2},\ s^k = \max(\mu t^k, 1),\\
b_{1x}^k &= w_{1x}^k/s^k,\ b_{1y}^k = w_{1y}^k/s^k,\\
b_{2x}^k &= w_{2x}^k/s^k,\ b_{2y}^k = w_{2y}^k/s^k,\\
b_{3x}^k &= w_{3x}^k/s^k,\ b_{3y}^k = w_{3y}^k/s^k,\\
\mathbf{U}^{k+1} &= \mathbf{C}^k - \mu\theta\big(\nabla_x^T b_{1x}^k + \nabla_y^T b_{1y}^k, \nabla_x^T b_{2x}^k + \nabla_y^T b_{2y}^k, \nabla_x^T b_{3x}^k + \nabla_y^T b_{3y}^k\big).\\
\alpha^{k+1} &= \frac{1}{\theta}\langle \mathbf{U}^{k+1}, \mathbf{C}^k\rangle + \hat{\lambda}\langle \mathbf{C}^k, \mathbf{C}_0\rangle - \frac{1}{\theta} - \hat{\lambda},\\
\mathbf{C}^{k+1} &= \frac{\mathbf{U}^{k+1} + \hat{\lambda}\theta\mathbf{C}_0}{1 + \hat{\lambda}\theta + \alpha^{k+1}\theta}.
\end{aligned}$$

Figures 3 and 4 show the results of the algorithm for TV colorization with CB color model. Figures 3(c) and 4(c) are the given images in which the color only appears in certain small region D^c and most region needs to be colorized. Here we define a mask with the same size as the image domain Ω such that the color is maintained in the given region D^c. The computational time is about 0.7 seconds for Figure 3 and 2 seconds for Figure 4, which is faster than the split Bregman method and the dual method.

3.3. TV-based inpainting. In this section, we employ the proposed algorithm for TV-based inpainting model. Assume $\Omega := \{1, \cdots, M\} \times \{1, \cdots, N\}$ be the image domain and $\mathbf{f} : \{1, \cdots, M\} \times \{1, \cdots, N\} \to \mathbb{R}^3$ be the observed color image. Let D be the inpainting domain and $D^c = \Omega \backslash D$ be the complement of D in Ω, where the intensities are given. The inpainting task is filling-in unknown data in a known region D of an image. The TV-based inpainting model is

$$\min_{\mathbf{u}} \|\nabla \mathbf{u}\|_1 + \frac{\hat{\lambda}}{2}\|\mathbf{u} - \mathbf{f}\|_2^2, \tag{32}$$

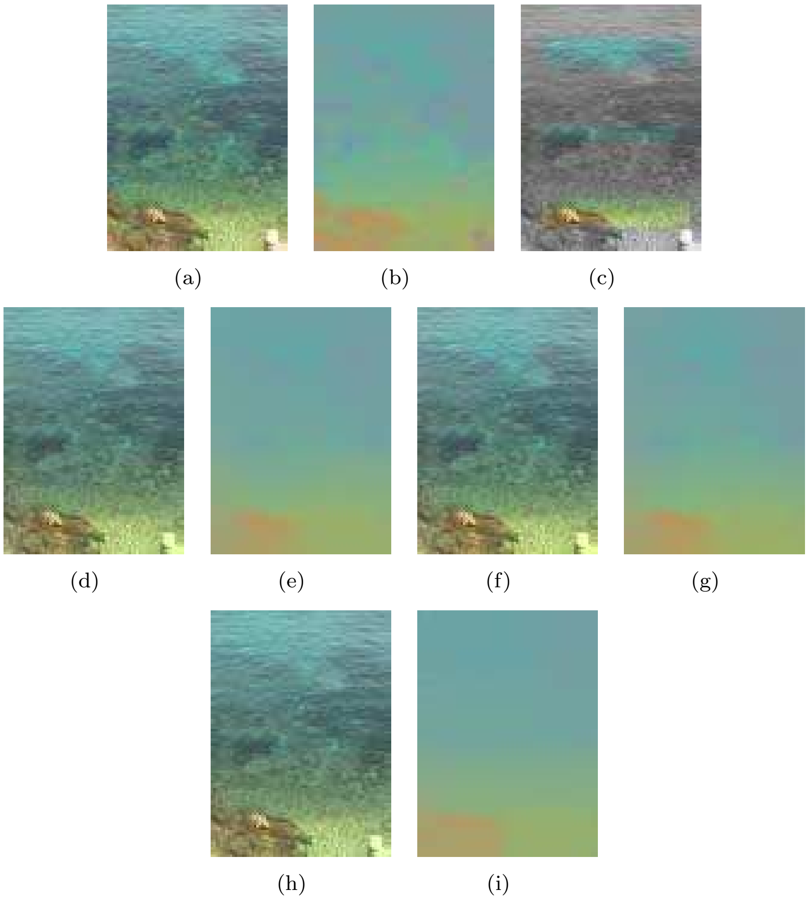

FIGURE 3. TV-based colorization results. Image size is 100×75. (a) Original image. (b) Chromaticity component. (c) Given image. (d) Result of proposed algorithm, iter=70, t=0.726s; $\theta = 0.05$, $\mu = 6$, $\lambda = 10$. (e) Result of chromaticity component using proposed algorithm. (f) Result of split Bregman algorithm, t=2.353s. (g) Result of chromaticity component using split Bregman algorithm. (h) Result of dual method in Li et al. [20], iter=300, t=2.917s. (i) Result of chromaticity component using dual method.

where

$$\hat{\lambda} = \begin{cases} \lambda, & (m,n) \in D^c, \\ 0, & (m,n) \in D, \end{cases}$$

$1 \le m \le M$ and $1 \le n \le N$.

Similar to the TV-based colorization model, we add an auxiliary variable $\mathbf{v}$, then the above energy can be approximated by

$$\min_{\mathbf{u},\mathbf{v}} \|\nabla \mathbf{v}\|_1 + \frac{1}{2\theta}\|\mathbf{v} - \mathbf{u}\|_2^2 + \frac{\hat{\lambda}}{2}\|\mathbf{u} - \mathbf{f}\|_2^2, \tag{33}$$

where θ should be chosen small enough in the numerical implementation so that $\mathbf{v}$ is close to $\mathbf{u}$ sufficiently. Thus, we solve $\mathbf{v}$ and $\mathbf{u}$ by minimizing the following

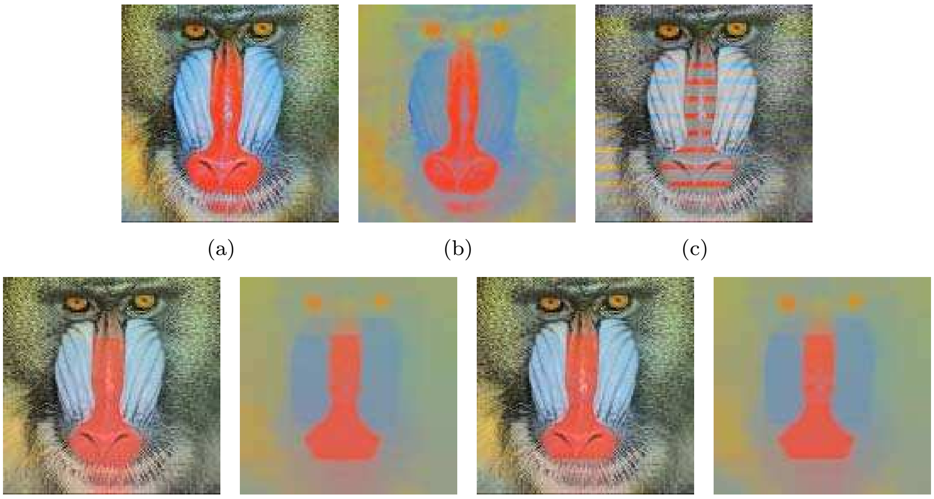

FIGURE 4. TV-based colorization results. Image size is 128×128. (a) Original image. (b) Chromaticity component. (c) Given image. (d) Result of proposed algorithm, iter=100, t=2.121s; $\theta = 0.1$, $\mu = 1$, $\lambda = 5$. (e) Result of chromaticity component using proposed algorithm. (f) Result of dual method in Li et al. [20], iter=300, t=7.160s. (g) Result of chromaticity component using dual method.

functionals respectively,

$$\min_{\mathbf{v}} \|\nabla \mathbf{v}\|_1 + \frac{1}{2\theta}\|\mathbf{v} - \mathbf{u}\|_2^2, \tag{34}$$

$$\min_{\mathbf{u}} \frac{1}{2\theta}\|\mathbf{v} - \mathbf{u}\|_2^2 + \frac{\hat{\lambda}}{2}\|\mathbf{u} - \mathbf{f}\|_2^2. \tag{35}$$

Finally, the iterative scheme is given by: set $b_{ix}^0 = b_{iy}^0 = 0$ $(i = 1, 2, 3)$, $\mathbf{u}^1 = \mathbf{f}$ and $\mathbf{v}^1 = \mathbf{u}^1$. For $k = 1, 2, \cdots$, let

$$w_{ix}^k = \nabla_x v_i^k + b_{ix}^{k-1},\ w_{iy}^k = \nabla_y v_i^k + b_{iy}^{k-1}, \quad \text{for} \quad i = 1, 2, 3,$$

$$t^k = \sqrt{\sum_{i=1}^{3} (w_{ix}^k)^2 + (w_{iy}^k)^2},\ s^k = \max(\mu t^k, 1),$$

$$b_{ix}^k = w_{ix}^k / s^k,\ b_{iy}^k = w_{iy}^k / s^k, \quad \text{for} \quad i = 1, 2, 3,$$

$$v_i^{k+1} = u_i^k - \mu\theta(\nabla_x^T b_{ix}^k + \nabla_y^T b_{iy}^k), \quad \text{for} \quad i = 1, 2, 3,$$

$$u_i^{k+1} = \frac{v_i^{k+1} + \hat{\lambda}\theta f_i}{1 + \hat{\lambda}\theta}, \quad \text{for} \quad i = 1, 2, 3.$$

When $i = 1$, we may use the algorithm for gray-scale image inpainting.

Figures 5 and 6 present the inpainting results for gray and color images. The given images, which are created by defining two appropriate masks in which the intensities in inpainting domains D are one and others are zero, are shown in Figures 5(a) and 6(a). We compare our algorithm with the augmented Lagrangian method,

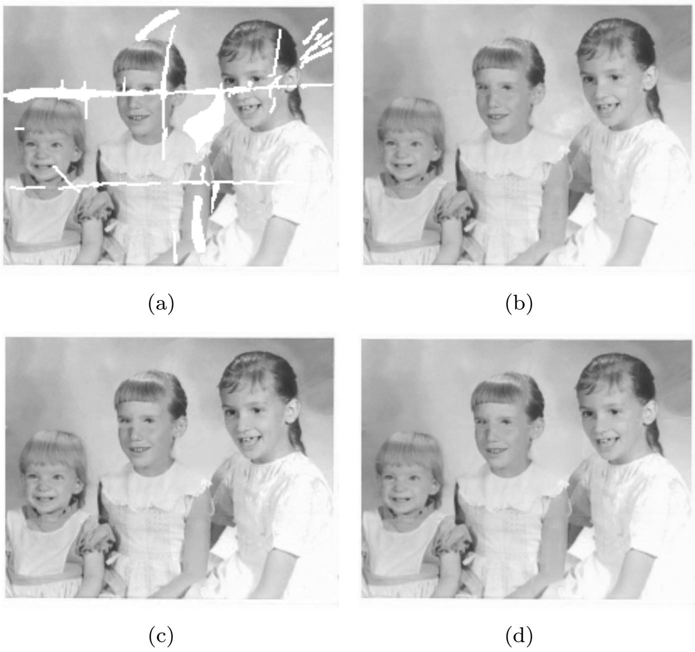

FIGURE 5. TV-based inpainting results for gray image. Image size is 202×241. (a) Image with mask. (b) Result of proposed algorithm, iter=80, t=1.701s; $\theta = 0.03$, $\mu = 4$, $\lambda = 100$. (c) Result of augmented Lagrangian method, iter=20, t=2.64s. (d) Result of dual method, iter=200, t=3.753s.

the split Bregman iteration and the dual method. The results show our algorithm is more efficient and faster, which can be seen from Figures 5(b) and 6(b).

4. Conclusion

In this paper, a simple and efficient algorithm was presented to solve the vectorial TV-based models. We first extended Jia and Zhao's algorithm [16] to vectorial version, and then extended the proposed vectorial algorithm to some applications besides image denoising. We would like to emphasize that the proposed vectorial algorithm can be used in many problems which need the ℓ_1 regularization. In comparison with the split Bregman iteration and the augmented Lagrangian method, our algorithm does not involve any differential equations or difference equations, which will save the computation. Moreover, the algorithm is convergent to the solution of original vectorial TV minimization problem if $0 < \mu/\lambda < 1/8$. The experiments on some image restoration tasks, such as color denoisings based on RGB and CB color models, CB-based colorization and gray and color image inpainting, indicated that our algorithm is more efficient and faster. In particular, it is faster than the dual method, the split Bregman iteration and the augmented Lagrangian method, which are also very fast and efficient algorithms relatively. Of course, the proposed algorithm also outperforms the classical algorithms, for example the explicit gradient descent flow. Besides the computational speed, the restoration qualities are better by using the proposed algorithm. In a word, the proposed algorithm is efficient for vectorial TV-based models.

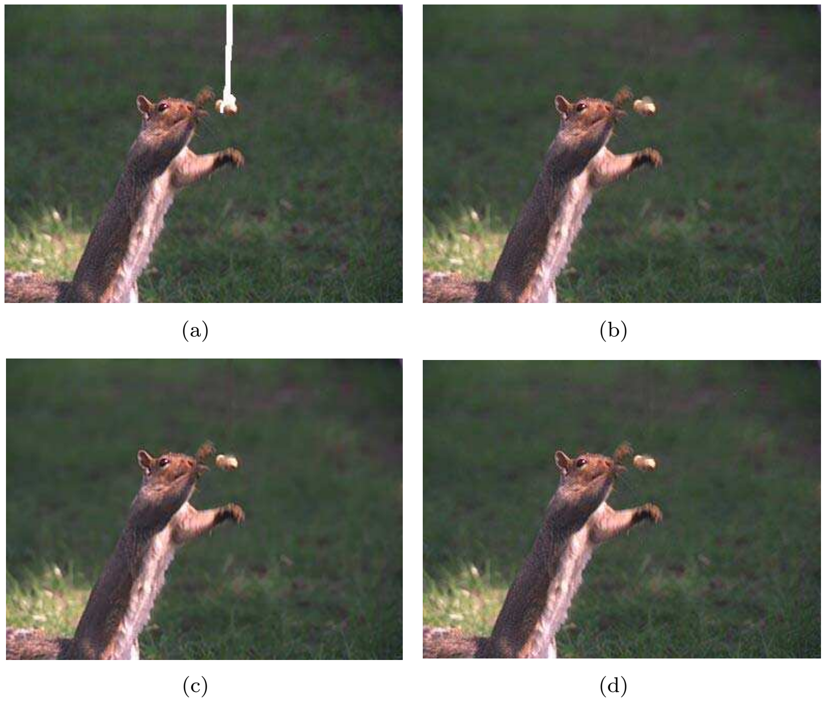

FIGURE 6. TV-based inpainting results for color image. Image size is 256×384. (a) Image with mask. (b) Result of proposed algorithm, iter=100, t=11.646s; $\theta = 0.05$, $\mu = 2$, $\lambda = 100$. (c) Result of augmented Lagrangian method, iter=20, t=13.684s. (c) Result of split Bregman algorithm, iter=100, t=18.690s.

Acknowledgments

We are grateful to Prof. Xue-Cheng Tai and anonymous referees for their useful comments and suggestions.

References

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

[2] P. Blomgren, T. F. Chan, P. Mulet, C. K. Wong, Total variation image restoration: numerical methods and extensions, International Conference on Image Processing, Santa Barbara, 3(1997), 384-387.

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

[4] J. F. Cai, S. Osher, and Z. Shen, Split Bregman methods and frame based image restoration, Multiscale Model. Simul., 8(2009), 337-369.

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

[6] A. Chambolle and P. L. Lions, Image recovery via total variation minimization and related problems, Numer. Math., 76(1997), 167-188.

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

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

[9] T. F. Chan, S. Osher and J. Shen, The digital TV filter and nonlinear denoising, IEEE Trans. Image Process., 10(2001), 231-241.
[10] T. F. Chan and J. Shen, On the role of the BV image model in image restoration, UCLA CAAM Report 02-14.
[11] T. F. Chan and J. Shen, Variational image inpainting, Comm. Pure Appl. Math., 58(2005), 579-619.
[12] T. F. Chan and C. K. Wong, Total variation blind deconvolution, IEEE Trans. Image Process., 7(1998), 370-375.
[13] T. F. Chan, H. M. Zhou and R. H. Chan, Continuation method for total variation denoising problems, UCLA CAAM Report 95-18.
[14] T. Goldstein and S. Osher, The Split Bregman method for L1 regularized problems, SIAM J. Imaging Sci., 2(2009), 323-343.
[15] L. He, A. Marquina and S. Osher, Blind deconvolution using TV regularization and Bregman iteration, Int. J. Imag. Syst. Tech., 15(2003), 74-83.
[16] R. Q. Jia and H. Q. Zhao, A fast algorithm for the total variation model of image denoising, Adv. Comput. Math., 2009, doi:10.1007/s10444-009-9128-5.
[17] R. Q. Jia, H. Q. Zhao and W. Zhao, Convergence analysis of the Bregman method for the variational model of image denosing, Appl. Comput. Harmon. Anal, 27(2009), 367-379.
[18] S. H. Kang and R. March, Variational models for image colorization via chromaticity and brightness decomposition, IEEE Trans. Image Process., 16(2007), 2251-2261.
[19] F. Lauze, Harmonic and TV inpainting, 2003.
[20] F. Li, Z. Bao, R. Liu and G. Zhang. Fast image inpainting and colorization by dual method, manuscript.
[21] M. Lysaker and X. C. Tai, Iterative image restoration combining total variation minimization and a second order functional, Int. J. Comput. Vis., 66(2006), 5-18.
[22] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, An iterative regularization method for total variation-based image restoration, Multiscale Model. Simul., 4(2005), 460-489.
[23] L. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica. D., 60(1992), 259-268.
[24] X. C. Tai, S. Osher and R. Holm, Image inpainting using a TV-Stokes equation, in: Image Processing based on partial differential equations, Springer, Heidelberg, 2006.
[25] X. C. Tai and C. Wu, Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model, UCLA CAAM Report 09-05.
[26] C. Wu and X. C. Tai, Augmented Lagrangian method, dual methods and split Bregman iteration for ROF, vectorial TV, and high order models, UCLA CAAM Report 09-76.
[27] C. Wu, J. Zhang and X. C. Tai, Augmented Lagrangian method for total variation restoration with non-quadratic fidelity, UCLA CAAM Report 09-82.
[28] J. Yang, W. Yin, Y. Zhang, and Y. Wang, A fast algorithm for edge-preserving variational multichannel image restoration, SIAM J. Imaging Sciences, 2(2009), 569-592.

Center of Mathematical Sciences, Zhejiang University, Hangzhou 310027, China
E-mail: `fangdong06@gmail.com`

Center of Mathematical Sciences, Zhejiang University, Hangzhou 310027, China
E-mail: `yjianbin@gmail.com and xxliu198431@126.com`