SIAM J. IMAGING SCIENCES
Vol. 3, No. 3, pp. 300–339

Augmented Lagrangian Method, Dual Methods, and Split Bregman Iteration for ROF, Vectorial TV, and High Order Models*

Chunlin Wu† and Xue-Cheng Tai†‡

Abstract. In image processing, the Rudin–Osher–Fatemi (ROF) model [L. Rudin, S. Osher, and E. Fatemi, *Phys. D*, 60 (1992), pp. 259–268] based on total variation (TV) minimization has proven to be very useful. So far many researchers have contributed to designing fast numerical schemes and overcoming the nondifferentiability of the model. Methods considered to be particularly efficient for the ROF model include the Chan–Golub–Mulet (CGM) primal-dual method [T.F. Chan, G.H. Golub, and P. Mulet, *SIAM J. Sci. Comput.*, 20 (1999), pp. 1964–1977], Chambolle's dual method [A. Chambolle, *J. Math. Imaging Vis.*, 20 (2004), pp. 89–97], the splitting and quadratic penalty-based method [Y. Wang, J. Yang, W. Yin, and Y. Zhang, *SIAM J. Imaging Sci.*, 1 (2008), pp. 248–272], and the split Bregman iteration [T. Goldstein and S. Osher, *SIAM J. Imaging Sci.*, 2 (2009), pp. 323–343], as well as the augmented Lagrangian method [X.C. Tai and C. Wu, *Lecture Notes in Comput. Sci.* 5567, Springer-Verlag, Berlin, 2009, pp. 502–513]. In this paper, we first review the augmented Lagrangian method for the ROF model and then provide some convergence analysis and extensions to vectorial TV and high order models. All the algorithms and analysis will be presented in the discrete setting, which is much clearer for practical implementation than the continuous setting as in Tai and Wu, above. We also present, in the discrete setting, the connections between the augmented Lagrangian method, the dual methods, and the split Bregman iteration. Using our extensions and observations, we can easily figure out CGM and the split Bregman iteration for vectorial TV and high order models, which, to the best of our knowledge, have not been presented in the literature. Numerical examples demonstrate the efficiency and accuracy of our method, especially in the image deblurring case.

Key words. augmented Lagrangian method, dual method, split Bregman iteration, ROF model, total variation

AMS subject classifications. 90C90, 68U10, 65K05

DOI. 10.1137/090767558

1. Introduction.

Image restoration such as denoising and deblurring are the most fundamental tasks in image processing. It is important but difficult to preserve image structures (such as edges) in image restoration. Recently, the Rudin–Osher–Fatemi (ROF) model [34] has been demonstrated to be very successful. It immediately attracted much attention and has been extended to high order models [18, 46, 29, 30, 26, 36] and vectorial models for color image restoration [35, 4, 6, 17]; see [15] for an overview.

However, the numerical computation of the ROF model suffers from difficulties related to its nonlinearity and nondifferentiability. In [34], the authors proposed a time marching

*Received by the editors August 10, 2009; accepted for publication (in revised form) May 3, 2010; published electronically July 22, 2010. This research was supported by MOE (Ministry of Education) Tier II project T207N2202 and IDM project NRF2007IDM-IDM002-010, as well as by SUG 20/07.
http://www.siam.org/journals/siims/3-3/76755.html
†Division of Mathematical Sciences, School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore (clwu@ntu.edu.sg, xctai@ntu.edu.sg).
‡Department of Mathematics, University of Bergen, N-5020 Bergen, Norway.

strategy for the associated Euler–Lagrange equation. This method is slow due to the constraint on stability conditions about the time step size. To find fast algorithms has been an active research area.

There are several methods that have proven to be particularly efficient for image restoration problems based on the ROF model. One class of approaches is dual methods and primal-dual methods [16, 11, 12, 49], which are based on the dual or primal-dual formulations of the ROF model. The other is based on variable-splitting and equality constrained optimization, e.g., the approach proposed in [41, 40, 43], which uses an alternative minimization of the quadratic-penalized cost functional for the equality constrained problem, and the method in [27], where splitting is applied to the data fidelity term, the split Bregman iteration [45, 24], as well as the augmented Lagrangian method [39]. In this paper, we extend the results in [39]. We will first review the augmented Lagrangian method to solve the ROF model, and then provide some convergence analysis, followed by extensions to vectorial total variation (TV) and high order models. All the algorithms and analysis will be presented in the discrete setting, which is much clearer for practical implementation than the description in the continuous setting as in [39]. We also present, in this discrete setting, the discrete analogue of the result in [39] that the augmented Lagrangian method, the dual methods, and the split Bregman iteration are just different iterative schemes for solving the same system based on the augmented Lagrangian saddle function. Some connections between Chan–Golub–Mulet method (CGM) and Chambolle's dual method for the ROF model were noted in [49]. The equivalence between the augmented Lagrangian method and Bregman (split Bregman) method was also noted in [45, 37, 22]. For the sake of completeness, here we give the whole discrete analogue of the result in [39], including how CGM and Chambolle's dual method are connected to the augmented Lagrangian method, and an explanation of the equivalence between the augmented Lagrangian method and the split Bregman iteration. Using our extensions and observations, we can easily figure out CGM, Chambolle's dual method, and the split Bregman iteration for vectorial TV and high order models. To the best of our knowledge, CGM and the split Bregman iteration for vectorial TV and high order models are still missing in the literature.

The paper is organized as follows. In the next section, we give basic notation. In section 3, we present the ROF model and some existing solvers, in the discrete setting. The augmented Lagrangian method will be given in section 4 with some convergence analysis. In section 5, we present connections between the proposed method and CGM and Chambolle's dual method as well as the split Bregman iteration. Our approach and observations are then extended to vectorial TV in section 6 and high order models in section 7. After providing some numerical experiments and comparisons in section 8, we conclude the paper in section 9.

2. Basic notation. Without loss of generality, we represent a grayscale image as an $N \times N$ matrix. The Euclidean space $\mathbb{R}^{N\times N}$ is denoted as V. The discrete gradient operator is a mapping $\nabla : V \to Q$, where $Q = V \times V$. For $u \in V$, ∇u is given by

$$(\nabla u)_{i,j} = ((\mathring{D}_x^+ u)_{i,j}, (\mathring{D}_y^+ u)_{i,j}),$$

with

$$(\mathring{D}_x^+ u)_{i,j} = \begin{cases} u_{i,j+1} - u_{i,j}, & 1 \le j \le N-1, \\ u_{i,1} - u_{i,N}, & j = N, \end{cases}$$

$$(\mathring{D}_y^+ u)_{i,j} = \begin{cases} u_{i+1,j} - u_{i,j}, & 1 \le i \le N-1, \\ u_{1,j} - u_{N,j}, & i = N, \end{cases}$$

where $i, j = 1, \ldots, N$. Here we use $\mathring{D}_x^+$ and $\mathring{D}_y^+$ to denote forward difference operators with periodic boundary condition (u is periodically extended). Consequently the FFT can be adopted in our algorithm. It should be pointed out that some other boundary conditions and corresponding definitions of ∇ can be used. For the Neumann boundary condition, we can adopt fast DCT (discrete cosine transform) in place of the FFT.

We denote the usual inner product and Euclidean norm of V as $(\cdot,\cdot)_V$ and $\|\cdot\|_V$, respectively. We also equip the space Q with inner product $(\cdot,\cdot)_Q$ and norm $\|\cdot\|_Q$, which are defined as follows. For $p = (p^1, p^2) \in Q$ and $q = (q^1, q^2) \in Q$,

$$(p, q)_Q = (p^1, q^1)_V + (p^2, q^2)_V$$

and

$$\|p\|_Q = \sqrt{(p, p)_Q}.$$

In addition, we mention that at each pixel (i, j),

$$|p_{i,j}| = |(p^1_{i,j}, p^2_{i,j})| = \sqrt{(p^1_{i,j})^2 + (p^2_{i,j})^2},$$

the usual Euclidean norm in $\mathbb{R}^2$. From the subscript i, j, one may regard $|p_{i,j}|$ as the pixel-by-pixel norm of p.

Using the inner products of V and Q, we can find the adjoint operator of $-\nabla$, i.e., the discrete divergence operator $\mathrm{div} : Q \to V$. Given $p = (p^1, p^2) \in Q$, we have

$$(\mathrm{div} p)_{i,j} = p^1_{i,j} - p^1_{i,j-1} + p^2_{i,j} - p^2_{i-1,j} = (\mathring{D}_x^- p^1)_{i,j} + (\mathring{D}_y^- p^2)_{i,j},$$

where $\mathring{D}_x^-$ and $\mathring{D}_y^-$ are backward difference operators with periodic boundary conditions $p^1_{i,0} = p^1_{i,N}$ and $p^2_{0,j} = p^2_{N,j}$.

3. The ROF model and some existing solvers.

Assume that $f \in V$ is an observed image and is degraded from the true image, $u \in V$, as follows:

$$f = Ku + n, \tag{3.1}$$

where $K : V \to V$ is a convolution operator and $n \in V$ is random Gaussian noise (the most typical noise model). Image restoration aims at recovering u from f. Since the problem is usually ill-posed, we cannot directly solve u from (3.1). Regularization of the solution should be considered. One of the most basic and successful image regularization models is the ROF model [34], which reads

$$\min_{u \in V} \left\{ F_{\mathrm{rof}}(u) = R_{\mathrm{rof}}(\nabla u) + \frac{\alpha}{2} \|Ku - f\|_V^2 \right\}, \tag{3.2}$$

where

$$R_{\mathrm{rof}}(\nabla u) = \mathrm{TV}(u) = \sum_{1 \le i,j \le N} |(\nabla u)_{i,j}| \tag{3.3}$$

is the total variation of u. Note here that $R_{\text{rof}}(\cdot)$ is regarded as a functional of ∇u. In [34], the authors considered the image denoising problem ($K = I$) and presented a gradient descent method to solve (3.2). Here the method is described for general K. The artificial time marching is introduced into the associated Euler–Lagrange equation (it is actually a system of ordinary differential equations, since we are in the discrete setting) as follows:

$$\begin{aligned} u_t &= \operatorname{div}\left(\frac{\nabla u}{\sqrt{|\nabla u|^2+\beta}}\right) + \alpha K^*(f - Ku), \\ u(0) &= f, \end{aligned} \tag{3.4}$$

where β is a small positive number to avoid zero division and K^* is the L_2 adjoint of K.

There are mainly two drawbacks for the gradient descent method (3.4). First, it is an approximation of the original problem (3.2), since the regularity term $R_{\text{rof}}(\nabla u)$ is smoothed and thus approximated to get (3.4). Second, the method is slow due to strict constraints on the time step. The choice of β will affect both of these aspects. When β is larger, the scheme is more efficient but the approximation is worse. Therefore there is a trade-off between accuracy and efficiency.

Many algorithms have been proposed to improve the gradient descent method, aiming to compute the solution of the ROF model (3.2) as efficiently and exactly as possible; see, e.g., the primal-dual method [16], the dual method [12], the split Bregman iteration [45, 24], and the splitting and quadratic penalty-based method [41, 40], as well as the augmented Lagrangian method [39].

The difficulty in solving the ROF restoration model (3.2) is due to the nondifferentiability of the TV seminorm. By using an operator-splitting technique [23, 41, 40, 24, 39], we can separate the calculation of the nondifferentiable term and the squared 2-norm term. Concretely, an auxiliary variable $p \in Q$ is introduced for ∇u. The model (3.2) is thus equivalent to

$$\begin{aligned} &\min_{u\in V, p\in Q} \left\{ G_{\text{rof}}(u,p) = R_{\text{rof}}(p) + \frac{\alpha}{2}\|Ku - f\|_V^2 \right\} \\ &\text{s.t.} \qquad p = \nabla u, \end{aligned} \tag{3.5}$$

which is a constrained optimization problem.

Since the blur is essentially averaging, it is reasonable to assume the following:

- The null spaces of ∇ and K have only 0 as common elements; i.e., $\text{Null}(\nabla) \cap \text{Null}(K) = \{0\}$.

Under this assumption, the functional $F_{\text{rof}}(u)$ in (3.2) is convex, proper, coercive, and continuous. According to the generalized Weierstrass theorem and Fermat's rule [21, 23], we have the following result.

Theorem 3.1. *The problem* (3.2) *has at least one solution* u*, which satisfies*

$$0 \in \alpha K^*(Ku - f) - \operatorname{div}\partial R_{\text{rof}}(\nabla u), \tag{3.6}$$

where $\partial R_{\text{rof}}(\nabla u)$ *is the subdifferential* [21] *of* R_{rof} *at* ∇u*. Moreover, if* $\text{Null}(K) = \{0\}$*, the minimizer is unique.*

In the following, we review some typical existing solvers for the ROF model.

3.1. The CGM. In [16] Chan, Golub, and Mulet proposed a primal-dual method for solving the ROF model. They introduced a new variable $\omega \in Q$, defined by

$$\omega_{i,j} = \frac{(\nabla u)_{i,j}}{|(\nabla u)_{i,j}|}, \qquad 1 \le i,j \le N, \tag{3.7}$$

to the Euler–Lagrange equation of (3.2), to remove some of the singularity caused by the nondifferentiability of the objective functional. This yields the following primal-dual system:

$$\begin{aligned} -\mathrm{div}\omega + \alpha K^*(Ku - f) &= 0, \\ \nabla u - \omega|\nabla u| &= 0, \\ |\omega_{i,j}| \le 1, \quad 1 \le i,j &\le N, \end{aligned} \tag{3.8}$$

where u and ω are called primal and dual variables, respectively. The system is then approximated using a smoothed TV seminorm (with some small positive β which can be updated iteration-by-iteration) in numerical computation. Newton's linearization technique for both the primal and dual variables is adopted. In the CGM, the parameter β can be very close to 0. However, it is relatively slow in the family of dual and primal-dual methods because it solves exactly a system of linear equations at each iteration.

3.2. Chambolle's dual method. Another work based on the dual formulation with a different derivation is due to Chambolle [12]. In this method, the primal variable of the image data is expressed explicitly with the dual variable, and only the dual variable is computed iteratively. However, the algorithm does not consider a general operator K in (3.2). In the following we introduce Chambolle's method in our context. (Note the difference in the boundary condition used, and the slight difference between (3.2) and the model in [12] about the parameter α.)

Denoting

$$S = \mathrm{Closure}\{\mathrm{div}\xi : \xi \in Q, |\xi_{i,j}| \le 1 \ \forall\ 1 \le i,j \le N\}, \tag{3.9}$$

Chambolle [12] showed that the ROF restoration model (3.2) with $K = I$ yields

$$u = f - \frac{1}{\alpha}\pi_S(\alpha f) = f - \pi_{\frac{S}{\alpha}}(f), \tag{3.10}$$

where $\pi_S(\cdot)$ is a nonlinear projection operator to S, which reads

$$\min_{\mathrm{div}\xi}\{\|\mathrm{div}\xi - \cdot\|_V^2 : \xi \in Q, |\xi_{i,j}| \le 1 \ \forall\ 1 \le i,j \le N\}. \tag{3.11}$$

From the Karush–Kuhn–Tucker (KKT) conditions and a careful observation, it was shown that ξ in the nonlinear projection satisfies

$$-(\nabla(\mathrm{div}\xi - \alpha f))_{i,j} + \xi_{i,j}|(\nabla(\mathrm{div}\xi - \alpha f))_{i,j}| = 0, \tag{3.12}$$

from which ξ can be calculated by the following semi-implicit gradient descent algorithm:

$$\xi_{i,j}^{k+1} = \frac{\xi_{i,j}^k + \tau(\nabla(\mathrm{div}\xi^k - \alpha f))_{i,j}}{1 + \tau|(\nabla(\mathrm{div}\xi^k - \alpha f))_{i,j}|}$$

for a chosen step length τ.

3.3. Split Bregman iteration. Recently, Bregman iteration and split Bregman iteration attracted much attention in the signal and image processing community [9, 10, 24, 31, 45, 44, 48]. The basic idea is to transform a constrained optimization problem to a series of unconstrained problems. In each unconstrained problem, the objective functional is defined via the Bregman distance [5] of a convex functional.

The Bregman distance of a convex functional $J(u)$ is defined as the following (nonnegative) quantity:

$$D_J^g(u,v) \equiv J(u) - J(v) - \langle g, u - v\rangle, \tag{3.13}$$

where $g \in \partial J(v)$, i.e., one of the subgradients of J at v.

When $J(u)$ is a continuously differentiable functional, its subdifferential $\partial J(v)$ has a single element for each v, and consequently the Bregman distance is unique. In this case the distance is just the difference at the point u between $J(u)$ and its first order approximation at the point v. For those nondifferentiable functionals, the subdifferential may be empty or contain multiple values. Therefore, the Bregman distance between u and v can be ill-defined or multivalued. However, this does not matter in Bregman distance–based iterative algorithms since the algorithms automatically choose a unique subgradient in each iteration as long as the fidelity term for the constraints is differentiable. (This condition usually holds.) We also recall here that the Bregman distance of a functional is not a distance in the usual sense since, in general, $D_J^g(u,v) \neq D_J^g(v,u)$ and the triangle inequality does not hold. See [31, 45] for more details.

To find the solution of the ROF model (3.2), or equivalently the constrained problem (3.5), the split Bregman iteration [24] solves a sequence of unconstrained problems of the form

$$(u^k, p^k) = \arg\min_{u\in V, p\in Q} D_{G_{\mathrm{rof}}}^{(g_u^{k-1}, g_p^{k-1})}((u,p),(u^{k-1},p^{k-1})) + \frac{r}{2}\|p - \nabla u\|_Q^2, \tag{3.14}$$

where g_u^{k-1} and g_p^{k-1}, sometimes written together as (g_u^{k-1}, g_p^{k-1}), are the subgradients of G_{rof} at (u^{k-1}, p^{k-1}) with respect to u and p, respectively, and r is a positive constant. Taking the update of the subgradients into consideration, the iteration procedure is formulated as Algorithm 3.1. The computation of (u^k, p^k) in the algorithm is similar to that of Algorithm 4.2, where, in the case of image denoising, the authors in [24] proposed using Gauss–Seidel iteration for the u-subproblem.

Algorithm 3.1. Split Bregman iteration for the ROF model.

1. *Initialization*: $u^{-1} = 0, p^{-1} = 0, g_u^{-1} = 0, g_p^{-1} = 0$;
2. *For* $k = 0, 1, 2, \ldots$: Compute (u^k, p^k) using (3.14), and update

$$\begin{aligned} g_u^k &= g_u^{k-1} - r\,\mathrm{div}(p^k - \nabla u^k),\\ g_p^k &= g_p^{k-1} - r(p^k - \nabla u^k). \end{aligned} \tag{3.15}$$

4. Augmented Lagrangian method for the ROF model and convergence analysis. The augmented Lagrangian method [25, 32, 33] has many advantages over other methods such as penalty methods [1] and has been successfully applied to nonlinear PDEs and mechanics [23].

In this section, we present the method for the ROF model, or equivalently the constrained problem (3.5). The details of the algorithms will be presented, followed by some convergence analysis.

4.1. Augmented Lagrangian method for the ROF model. We first define the augmented Lagrangian functional for the constrained optimization problem (3.5) as follows:

$$\mathscr{L}_{\mathrm{rof}}(v,q;\mu) = R_{\mathrm{rof}}(q) + \frac{\alpha}{2}\|Kv-f\|_V^2 + (\mu, q-\nabla v)_Q + \frac{r}{2}\|q-\nabla v\|_Q^2, \tag{4.1}$$

where $\mu \in Q$ is the Lagrange multiplier and r is a positive constant. For the augmented Lagrangian method for (3.5), we consider the following saddle-point problem:

$$\begin{aligned} &\text{Find} \quad (u,p;\lambda) \in V \times Q \times Q \\ &\text{s.t.} \quad \mathscr{L}_{\mathrm{rof}}(u,p;\mu) \le \mathscr{L}_{\mathrm{rof}}(u,p;\lambda) \le \mathscr{L}_{\mathrm{rof}}(v,q;\lambda)\ \forall (v,q;\mu) \in V \times Q \times Q. \end{aligned} \tag{4.2}$$

The relation between the saddle-point of problem (4.2) and the solution of (3.2) is stated in the following theorem.

Theorem 4.1. *$u \in V$ is a solution of (3.2) if and only if there exist $p \in Q$ and $\lambda \in Q$ such that $(u,p;\lambda)$ is a solution of (4.2).*

Proof. Suppose that $(u,p;\lambda)$ is a solution of (4.2). From the first inequality in (4.2) we have

$$p - \nabla u = 0. \tag{4.3}$$

The above relation, together with the second inequality in (4.2), shows

$$\begin{aligned} R_{\mathrm{rof}}(\nabla u) + \frac{\alpha}{2}\|Ku-f\|_V^2 \le R_{\mathrm{rof}}(q) + \frac{\alpha}{2}\|Kv-f\|_V^2 + (\lambda, q-\nabla v)_Q + \frac{r}{2}\|q-\nabla v\|_Q^2 \\ \forall (v,q) \in V \times Q. \end{aligned} \tag{4.4}$$

Taking $q = \nabla v$ in the above equation indicates that u is a solution of (3.2).

Conversely, we assume that $u \in V$ is a solution of (3.2). We take $p = \nabla u \in Q$. From (3.6), there exists one λ such that $-\lambda \in \partial R_{\mathrm{rof}}(\nabla u)$ and $\mathrm{div}\lambda = -\alpha K^*(Ku-f)$. We verify that $(u,p;\lambda)$ is a saddle-point of $\mathscr{L}_{\mathrm{rof}}$; i.e., $\mathscr{L}_{\mathrm{rof}}(u,p;\mu) \le \mathscr{L}_{\mathrm{rof}}(u,p;\lambda) \le \mathscr{L}_{\mathrm{rof}}(v,q;\lambda)\ \forall (v,q;\mu) \in V \times Q \times Q$. Since $p = \nabla u$, the first inequality holds. In the following we show $\mathscr{L}_{\mathrm{rof}}(u,p;\lambda) \le \mathscr{L}_{\mathrm{rof}}(v,q;\lambda)\ \forall (v,q) \in V \times Q$. Since

$$\begin{aligned} \mathscr{L}_{\mathrm{rof}}(v,q;\lambda) &= R_{\mathrm{rof}}(q) + \frac{\alpha}{2}\|Kv-f\|_V^2 + (\lambda, q-\nabla v)_Q + \frac{r}{2}\|q-\nabla v\|_Q^2 \\ &= R_{\mathrm{rof}}(q) + \frac{\alpha}{2}\|Kv-f\|_V^2 + \frac{r}{2}\left\|q-\nabla v+\frac{\lambda}{r}\right\|_Q^2 - \frac{1}{2r}\|\lambda\|_Q^2 \end{aligned}$$

is convex, proper, coercive, and continuous with respect to (v,q), $\mathscr{L}_{\mathrm{rof}}(v,q;\lambda)$ has a minimizer $(\bar{v},\bar{q})$ over $V \times Q$, which is characterized [21, 23] by

$$R_{\mathrm{rof}}(q) - R_{\mathrm{rof}}(\bar{q}) + (\lambda, q-\bar{q})_Q + r(\bar{q}-\nabla\bar{v}, q-\bar{q})_Q \ge 0 \qquad \forall q \in Q \tag{4.5}$$

and

$$\frac{\alpha}{2}\|Kv-f\|_V^2-\frac{\alpha}{2}\|K\bar{v}-f\|_V^2+(\mathrm{div}\lambda,v-\bar{v})_V+r(\mathrm{div}(\bar{q}-\nabla\bar{v}),v-\bar{v})_V\geq 0\quad\forall v\in V. \tag{4.6}$$

It is straightforward to verify that (u,p) satisfies (4.5) and (4.6). This completes the proof. ∎

Theorem 4.1, together with Theorem 3.1, shows that the problem (4.2) has at least one solution and each u in the solutions solves the original problem (3.2). We then use an iterative algorithm to solve the saddle-point problem (4.2); see Algorithm 4.1.

Algorithm 4.1. Augmented Lagrangian method for the ROF model.

1. *Initialization*: $\lambda^0=0$;
2. *For* $k=0,1,2,\ldots$: compute (u^k,p^k) as an (approximate) minimizer of the augmented Lagrangian functional with the Lagrange multiplier λ^k, i.e.,

$$(u^k,p^k)\approx\arg\min_{(v,q)\in V\times Q}\mathscr{L}_{\mathrm{rof}}(v,q;\lambda^k), \tag{4.7}$$

where $\mathscr{L}_{\mathrm{rof}}(v,q;\lambda^k)$ is defined in (4.1); update

$$\lambda^{k+1}=\lambda^k+r(p^k-\nabla u^k). \tag{4.8}$$

We are now left with the minimization problem (4.7) to address. One may notice the symbol $\approx$ in this problem. This is because, in general, it is difficult to find the minimizers u^k and p^k exactly in practical computation since v and q are coupled together. Usually, one separates the variables v and q and then uses an alternative minimization procedure [41, 40, 24] to solve (4.7), through which in practice one can obtain the minimizer only approximately. However, this does not affect the convergence of the whole Algorithm 4.1. More details are as follows.

We separate (4.7) into the following two subproblems:

$$\min_{v\in V}\frac{\alpha}{2}\|Kv-f\|_V^2-(\lambda^k,\nabla v)_Q+\frac{r}{2}\|q-\nabla v\|_Q^2 \tag{4.9}$$

for a given q, and

$$\min_{q\in Q}R_{\mathrm{rof}}(q)+(\lambda^k,q)_Q+\frac{r}{2}\|q-\nabla v\|_Q^2 \tag{4.10}$$

for a given v.

The subproblems (4.9) and (4.10) can be efficiently solved. For (4.9), the optimality condition gives a linear equation

$$\alpha K^*(Kv-f)+\mathrm{div}\lambda^k+r\mathrm{div}q-r\triangle v=0,$$

by the periodic boundary condition we are using. It allows us to use Fourier transforms and thus an FFT implementation as done in [41, 40], which, to the best of our knowledge, are the

first papers using FFT in TV minimization problems. Denoting $\mathcal{F}(v)$ as the Fourier transform of v, we write the solution as follows:

$$v = \mathcal{F}^{-1}\left(\frac{\alpha\mathcal{F}(K^*)\mathcal{F}(f) - \mathcal{F}(\mathring{D}_x^-)\mathcal{F}((\lambda^1)^k + rq^1) - \mathcal{F}(\mathring{D}_y^-)\mathcal{F}((\lambda^2)^k + rq^2)}{\alpha\mathcal{F}(K^*)\mathcal{F}(K) - r\mathcal{F}(\triangle)}\right), \tag{4.11}$$

where $\lambda^k = ((\lambda^1)^k, (\lambda^2)^k)$ and $q = (q^1, q^2)$; Fourier transforms of operators such as $K, \mathring{D}_x^-, \mathring{D}_y^-$, $\triangle = \mathring{D}_x^-\mathring{D}_x^+ + \mathring{D}_y^-\mathring{D}_y^+$ are regarded as the transforms of their corresponding convolution kernels. For (4.10), we actually have the following closed form solution [8, 40, 39]:

$$q_{i,j} = \begin{cases} \left(1 - \frac{1}{r}\frac{1}{|w_{i,j}|}\right) w_{i,j}, & |w_{i,j}| > \frac{1}{r}, \\ 0, & |w_{i,j}| \le \frac{1}{r}, \end{cases} \tag{4.12}$$

where

$$w = \nabla v - \frac{\lambda^k}{r}. \tag{4.13}$$

Some thresholding techniques similar to (4.12) have been used in [19] for image denoising in the wavelet domain. At the end of this section, we would like to remark that the two subproblems (4.9) and (4.10) also appear in the split Bregman method [24] in a different form using subgradients. The split Bregman method also uses the shrinkage formulae (4.12) to solve (4.10). However, the authors in [24] focused on the image denoising case of the ROF restoration and proposed using Gauss–Seidel iteration for (4.9) with $K = I$. This is different from our FFT-based implementation. According to our numerical test, the split Bregman iteration is faster than our method in the case of image denoising. There are two main reasons for this. First, for image denoising problems (where $K = I$), the system (4.9) is sparse in the physical domain but dense in the frequency domain. Second, it is unnecessary to solve (4.9) to full accuracy in the split Bregman iteration (and also in the augmented Lagrangian method), and one single Gauss–Seidel iteration is enough. However, we mention that, for general K (which is usually dense in physical domain), a Gauss–Seidel iteration may not be suitable for this problem. Our method can handle more general cases.

We then iteratively and alternatively compute the v and q according to (4.11) and (4.12). This has a Gauss-Seidel flavor. The procedure is shown in Algorithm 4.2.

Algorithm 4.2. Augmented Lagrangian method for the ROF model—solve the minimization problem (4.7).

- *Initialization*: $u^{k,0} = u^{k-1}, p^{k,0} = p^{k-1}$;
- *For* $l = 0, 1, 2, \ldots, L-1$: Compute $u^{k,l+1}$ from (4.11) for $q = p^{k,l}$, and then compute $p^{k,l+1}$ from (4.12) for $v = u^{k,l+1}$;
- $u^k = u^{k,L}, p^k = p^{k,L}$.

Here L can be chosen using some convergence test techniques. In this paper, we simply set $L = 1$. In our experiments we found that with larger L (> 1) the algorithm wastes the accuracy of the inner iteration and does not dramatically speed up the convergence of the whole algorithm. This was also observed in [24] for the split Bregman method.

4.2. Convergence analysis. We show some convergence results for the augmented Lagrangian method. We first give the convergence of Algorithm 4.2, and then present two convergence results for Algorithm 4.1 where the minimization problem (4.7) is computed by Algorithm 4.2 with full accuracy ($L \to \infty$) and rough accuracy ($L = 1$), respectively.

Theorem 4.2. *The sequence* $\{(u^{k,l}, p^{k,l}) : l = 0, 1, 2, \ldots\}$ *generated by Algorithm 4.2 converges to a solution of the problem* (4.7).

Proof. The proof is motivated by [40]. Here we just sketch the differences.

We define an operator S similarly as in [40], such that (4.12) is reformulated as $q = S(w)$, where w is as in (4.13). Therefore the iterative scheme in Algorithm 4.2 is as follows:

$$\begin{cases} u^{k,l+1} = (\nabla^*\nabla + \frac{\alpha}{r}K^*K)^{-1}\left(\nabla^* p^{k,l} + \nabla^*\frac{\lambda^k}{r} + \frac{\alpha}{r}K^* f\right), \\ p^{k,l+1} = S\left(\nabla u^{k,l+1} - \frac{\lambda^k}{r}\right), \end{cases} \tag{4.14}$$

where $\nabla^* = -\mathrm{div}$ is the adjoint operator of ∇. Here we also mention the existence of $(\nabla^*\nabla + \frac{\alpha}{r}K^*K)^{-1}$ under the assumption $\mathrm{Null}(\nabla) \cap \mathrm{Null}(K) = \{0\}$. We then define a linear operator $h : Q \to Q$ as

$$h(q) = \nabla\left(\nabla^*\nabla + \frac{\alpha}{r}K^*K\right)^{-1}\left(\nabla^* q + \nabla^*\frac{\lambda^k}{r} + \frac{\alpha}{r}K^* f\right) - \frac{\lambda^k}{r}. \tag{4.15}$$

It is straightforward to verify the nonexpansiveness of h defined above.

Rewriting the iterative scheme (4.14) as

$$\begin{cases} u^{k,l+1} = (\nabla^*\nabla + \frac{\alpha}{r}K^*K)^{-1}\left(\nabla^* p^{k,l} + \nabla^*\frac{\lambda^k}{r} + \frac{\alpha}{r}K^* f\right), \\ p^{k,l+1} = S \circ h(p^{k,l}), \end{cases} \tag{4.16}$$

one can show the convergence via an argument similar to that of [40]. ∎

In the following we give the convergence of Algorithm 4.1 where the minimization problem (4.7) is computed by Algorithm 4.2 with full accuracy ($L \to \infty$) and rough accuracy ($L = 1$), respectively. We should point out that the idea of our proofs follows the convergence proof in [23]. However, the convergence proof of (u^k, p^k) (see (4.18) and (4.36)) in [23] requires the uniform convexity of $R_{\mathrm{rof}}(p)$ (in our context) and thus cannot be directly applied to our case. In addition to modifying this part, we provide more details to make the proof clearer.

Theorem 4.3. *Assume that* $(u, p; \lambda)$ *is a saddle-point of* $\mathscr{L}_{\mathrm{rof}}(v, q; \mu)$. *Suppose that the minimization problem* (4.7) *is exactly solved in each iteration; i.e.,* $L \to \infty$ *in Algorithm 4.2. Then the sequence* $(u^k, p^k; \lambda^k)$ *generated by Algorithm 4.1 satisfies*

$$\begin{cases} \lim\limits_{k\to\infty} G_{\mathrm{rof}}(u^k, p^k) = G_{\mathrm{rof}}(u, p), \\ \lim\limits_{k\to\infty} \|p^k - \nabla u^k\|_Q = 0. \end{cases} \tag{4.17}$$

Since $R_{\mathrm{rof}}(p)$ *is continuous,* (4.17) *indicates that* u^k *is a minimizing sequence of* F_{rof}. *If we further have* $\mathrm{Null}(K) = \{0\}$, *then*

$$\begin{cases} \lim\limits_{k\to\infty} u^k = u, \\ \lim\limits_{k\to\infty} p^k = p. \end{cases} \tag{4.18}$$

Proof. Let us define $\overline{u}^k, \overline{p}^k, \overline{\lambda}^k$ as

$$\overline{u}^k = u^k - u, \quad \overline{p}^k = p^k - p, \quad \overline{\lambda}^k = \lambda^k - \lambda.$$

Since $(u, p; \lambda)$ is a saddle-point of $\mathscr{L}_{\text{rof}}(v, q; \mu)$, we have

$$\mathscr{L}_{\text{rof}}(u, p; \mu) \le \mathscr{L}_{\text{rof}}(u, p; \lambda) \le \mathscr{L}_{\text{rof}}(v, q; \lambda) \quad \forall (v, q; \mu) \in V \times Q \times Q. \tag{4.19}$$

From the first inequality of (4.19), we have $p = \nabla u$. This relationship, together with (4.8), indicates

$$\overline{\lambda}^{k+1} = \overline{\lambda}^k + r(\overline{p}^k - \nabla \overline{u}^k).$$

It then follows that

$$\|\overline{\lambda}^k\|_Q^2 - \|\overline{\lambda}^{k+1}\|_Q^2 = -2r(\overline{\lambda}^k, \overline{p}^k - \nabla \overline{u}^k)_Q - r^2 \|\overline{p}^k - \nabla \overline{u}^k\|_Q^2. \tag{4.20}$$

On the other hand, from the second inequality of (4.19), (u, p) is characterized by

$$\frac{\alpha}{2}\|Kv - f\|_V^2 - \frac{\alpha}{2}\|Ku - f\|_V^2 + (\text{div}\lambda, v - u)_V + r(\text{div}(p - \nabla u), v - u)_V \ge 0 \quad \forall v \in V, \tag{4.21}$$

$$R_{\text{rof}}(q) - R_{\text{rof}}(p) + (\lambda, q - p)_Q + r(p - \nabla u, q - p)_Q \ge 0 \quad \forall q \in Q. \tag{4.22}$$

Similarly, (u^k, p^k) is characterized by

$$\frac{\alpha}{2}\|Kv - f\|_V^2 - \frac{\alpha}{2}\|Ku^k - f\|_V^2 + (\text{div}\lambda^k, v - u^k)_V + r(\text{div}(p^k - \nabla u^k), v - u^k)_V \ge 0 \quad \forall v \in V, \tag{4.23}$$

$$R_{\text{rof}}(q) - R_{\text{rof}}(p^k) + (\lambda^k, q - p^k)_Q + r(p^k - \nabla u^k, q - p^k)_Q \ge 0 \quad \forall q \in Q, \tag{4.24}$$

since (u^k, p^k) is the solution of (4.7). Taking $v = u^k$ in (4.21) and $v = u$ in (4.23), we obtain, by addition,

$$-(\overline{\lambda}^k, \nabla \overline{u}^k)_Q - r(\overline{p}^k - \nabla \overline{u}^k, \nabla \overline{u}^k)_Q \le 0. \tag{4.25}$$

Similarly, we have

$$(\overline{\lambda}^k, \overline{p}^k)_Q + r(\overline{p}^k - \nabla \overline{u}^k, \overline{p}^k)_Q \le 0, \tag{4.26}$$

by taking $q = p^k$ in (4.22), $q = p$ in (4.24), and then adding. It then follows that

$$(\overline{\lambda}^k, \overline{p}^k - \nabla \overline{u}^k)_Q + r\|\overline{p}^k - \nabla \overline{u}^k\|_Q^2 \le 0 \tag{4.27}$$

if we add (4.25) and (4.26) together.

By (4.27) and (4.20), we have

$$\|\overline{\lambda}^k\|_Q^2 - \|\overline{\lambda}^{k+1}\|_Q^2 \ge r^2 \|\overline{p}^k - \nabla \overline{u}^k\|_Q^2 \ge 0, \tag{4.28}$$

which implies

$$\begin{cases} \{\lambda^k : \forall k\} & \text{is bounded,} \\ \lim\limits_{k\to\infty} \|p^k - \nabla u^k\|_Q = 0. \end{cases} \tag{4.29}$$

Moreover, the second inequality of (4.19) indicates

$$G_{\rm rof}(u,p) \le G_{\rm rof}(u^k,p^k) + (\lambda, p^k - \nabla u^k)_Q + \frac{r}{2}\|p^k - \nabla u^k\|_Q^2. \tag{4.30}$$

If we take $v = u$ in (4.23) and $q = p$ in (4.24), we have, by addition,

$$G_{\rm rof}(u,p) \ge G_{\rm rof}(u^k,p^k) + (\lambda^k, p^k - \nabla u^k)_Q + r\|p^k - \nabla u^k\|_Q^2. \tag{4.31}$$

Using (4.29), we have

$$\liminf G_{\rm rof}(u^k,p^k) \ge G_{\rm rof}(u,p) \ge \limsup G_{\rm rof}(u^k,p^k), \tag{4.32}$$

by taking lim inf in (4.30) and lim sup in (4.31). Hence we complete the proof of (4.17).

In the following we show (4.18) if $\mathrm{Null}(K) = \{0\}$ holds. Since $(u,p;\lambda)$ is a saddle-point of $\mathscr{L}_{\rm rof}(v,q;\mu)$, we have

$$-\lambda \in \partial R_{\rm rof}(p), \tag{4.33}$$

$$\mathrm{div}\lambda = -\alpha K^*(Ku - f), \tag{4.34}$$

where $\partial R_{\rm rof}(p)$ is the subdifferential of $R_{\rm rof}$ at p. Then, we deduce

$$\begin{aligned}
&G_{\rm rof}(u^k,p^k) + (\lambda, p^k - \nabla u^k)_Q \\
&\ge R_{\rm rof}(p) - (\lambda, p^k - p)_Q + \frac{\alpha}{2}\|Ku^k - f\|_V^2 + (\lambda, p^k - \nabla u^k)_Q \\
&= R_{\rm rof}(p) + \frac{\alpha}{2}\|Ku^k - f\|_V^2 + (\lambda, \nabla u - \nabla u^k)_Q \\
&\ge R_{\rm rof}(p) + \frac{\alpha}{2}\left\|K\frac{u^k+u}{2} - f\right\|_V^2 + \alpha\left(K^*\left(K\frac{u^k+u}{2} - f\right), \frac{u^k-u}{2}\right)_V + (\lambda, \nabla u - \nabla u^k)_Q \\
&= R_{\rm rof}(p) + \frac{\alpha}{2}\|Ku - f\|_V^2 + \frac{\alpha}{2}\left\|K\frac{u^k+u}{2} - f\right\|_V^2 - \frac{\alpha}{2}\|Ku - f\|_V^2 \\
&\quad + \alpha\left(K^*\left(K\frac{u^k+u}{2} - f\right), \frac{u^k-u}{2}\right)_V + (\lambda, \nabla u - \nabla u^k)_Q \\
&= R_{\rm rof}(p) + \frac{\alpha}{2}\|Ku - f\|_V^2 + \frac{\alpha}{2}\left\|K\frac{u^k+u}{2} - f\right\|_V^2 - \frac{\alpha}{2}\|Ku - f\|_V^2 \\
&\quad + \alpha\left(K\frac{u^k+u}{2} - f, K\frac{u^k-u}{2}\right)_V + \alpha(Ku - f, K(u - u^k))_V \\
&= G_{\rm rof}(u,p) + \frac{3}{8}\alpha\|K(u^k - u)\|_V^2,
\end{aligned}$$

from which we obtain

$$\lim_{k\to\infty} \|K(u^k - u)\|_V = 0,$$

according to (4.17). If $\mathrm{Null}(K) = \{0\}$ holds, it follows that

$$\lim_{k\to\infty} u^k = u.$$

This result, together with the second equation in (4.17), yields

$$\lim_{k\to\infty} p^k = \nabla u = p,$$

which completes the proof. ∎

Theorem 4.4. *Assume that $(u, p; \lambda)$ is a saddle-point of $\mathscr{L}_{\mathrm{rof}}(v, q; \mu)$. Suppose that the minimization problem (4.7) is roughly solved in each iteration, i.e., with $L = 1$ in Algorithm 4.2. Then the sequence $(u^k, p^k; \lambda^k)$ generated by Algorithm 4.1 satisfies*

$$\begin{cases} \lim\limits_{k\to\infty} G_{\mathrm{rof}}(u^k, p^k) = G_{\mathrm{rof}}(u, p), \\ \lim\limits_{k\to\infty} \|p^k - \nabla u^k\|_Q = 0. \end{cases} \tag{4.35}$$

Since $R_{\mathrm{rof}}(p)$ is continuous, (4.35) indicates that u^k is a minimizing sequence of F_{rof}. If we further have $\mathrm{Null}(K) = \{0\}$, then

$$\begin{cases} \lim\limits_{k\to\infty} u^k = u, \\ \lim\limits_{k\to\infty} p^k = p. \end{cases} \tag{4.36}$$

Proof. Again we define the following errors:

$$\overline{u}^k = u^k - u, \quad \overline{p}^k = p^k - p, \quad \overline{\lambda}^k = \lambda^k - \lambda.$$

In this case, (4.20) still holds, which is presented as follows:

$$\|\overline{\lambda}^k\|_Q^2 - \|\overline{\lambda}^{k+1}\|_Q^2 = -2r(\overline{\lambda}^k, \overline{p}^k - \nabla\overline{u}^k)_Q - r^2\|\overline{p}^k - \nabla\overline{u}^k\|_Q^2. \tag{4.37}$$

Since $(u, p; \lambda)$ is a saddle-point of $\mathscr{L}_{\mathrm{rof}}(v, q; \mu)$, (u, p) is characterized by

$$\frac{\alpha}{2}\|Kv - f\|_V^2 - \frac{\alpha}{2}\|Ku - f\|_V^2 + (\mathrm{div}\lambda, v - u)_V + r(\mathrm{div}(p - \nabla u), v - u)_V \geq 0 \quad \forall v \in V, \tag{4.38}$$

$$R_{\mathrm{rof}}(q) - R_{\mathrm{rof}}(p) + (\lambda, q - p)_Q + r(p - \nabla u, q - p)_Q \geq 0 \quad \forall q \in Q. \tag{4.39}$$

Similarly, by the construction of (u^k, p^k) (Algorithm 4.2 with $L = 1$), we have

$$\frac{\alpha}{2}\|Kv - f\|_V^2 - \frac{\alpha}{2}\|Ku^k - f\|_V^2 + (\mathrm{div}\lambda^k, v - u^k)_V + r(\mathrm{div}(p^{k-1} - \nabla u^k), v - u^k)_V \geq 0$$
$$\forall v \in V, \tag{4.40}$$

$$R_{\rm rof}(q) - R_{\rm rof}(p^k) + (\lambda^k, q - p^k)_Q + r(p^k - \nabla u^k, q - p^k)_Q \geq 0 \quad \forall q \in Q. \tag{4.41}$$

Taking $v = u^k$ in (4.38), $v = u$ in (4.40), and $q = p^k$ in (4.39), as well as $q = p$ in (4.41), we obtain, after addition,

$$(\overline{\lambda}^k, \overline{p}^k - \nabla \overline{u}^k)_Q + r\|\overline{p}^k - \nabla \overline{u}^k\|_Q^2 + r(\nabla \overline{u}^k, \overline{p}^k - \overline{p}^{k-1})_Q \leq 0. \tag{4.42}$$

It then follows from (4.37) and (4.42) that

$$\|\overline{\lambda}^k\|_Q^2 - \|\overline{\lambda}^{k+1}\|_Q^2 \geq r^2\|\overline{p}^k - \nabla \overline{u}^k\|_Q^2 + 2r^2(\nabla \overline{u}^k, \overline{p}^k - \overline{p}^{k-1})_Q. \tag{4.43}$$

In the following we estimate $(\nabla \overline{u}^k, \overline{p}^k - \overline{p}^{k-1})_Q$ in (4.43). We have

$$\begin{aligned}(\nabla \overline{u}^k, \overline{p}^k - \overline{p}^{k-1})_Q = {} & (\nabla \overline{u}^k - \nabla \overline{u}^{k-1}, \overline{p}^k - \overline{p}^{k-1})_Q \\ & + (\nabla \overline{u}^{k-1} - \overline{p}^{k-1}, \overline{p}^k - \overline{p}^{k-1})_Q + (\overline{p}^{k-1}, \overline{p}^k - \overline{p}^{k-1})_Q.\end{aligned} \tag{4.44}$$

On the other hand, by the construction of p^{k-1} (from u^{k-1}), it follows that

$$R_{\rm rof}(q) - R_{\rm rof}(p^{k-1}) + (\lambda^{k-1}, q - p^{k-1})_Q + r(p^{k-1} - \nabla u^{k-1}, q - p^{k-1})_Q \geq 0 \quad \forall q \in Q. \tag{4.45}$$

Taking $q = p^{k-1}$ in (4.41) and $q = p^k$ in (4.45), we obtain, by addition,

$$r\|\overline{p}^k - \overline{p}^{k-1}\|_Q^2 + (\overline{p}^k - \overline{p}^{k-1}, \overline{\lambda}^k - \overline{\lambda}^{k-1})_Q - r(\overline{p}^k - \overline{p}^{k-1}, \nabla \overline{u}^k - \nabla \overline{u}^{k-1})_Q \leq 0. \tag{4.46}$$

Since

$$\overline{\lambda}^k - \overline{\lambda}^{k-1} = \lambda^k - \lambda^{k-1} = r(\overline{p}^{k-1} - \nabla \overline{u}^{k-1}),$$

we have

$$(\overline{p}^k - \overline{p}^{k-1}, \nabla \overline{u}^k - \nabla \overline{u}^{k-1})_Q + (\overline{p}^k - \overline{p}^{k-1}, \nabla \overline{u}^{k-1} - \overline{p}^{k-1})_Q \geq \|\overline{p}^k - \overline{p}^{k-1}\|_Q^2 \tag{4.47}$$

according to (4.46). Equations (4.44) and (4.47), together with the identity

$$(\overline{p}^{k-1}, \overline{p}^k - \overline{p}^{k-1})_Q = \frac{1}{2}(\|\overline{p}^k\|_Q^2 - \|\overline{p}^{k-1}\|_Q^2 - \|\overline{p}^k - \overline{p}^{k-1}\|_Q^2),$$

imply

$$(\nabla \overline{u}^k, \overline{p}^k - \overline{p}^{k-1})_Q \geq \frac{1}{2}(\|\overline{p}^k\|_Q^2 - \|\overline{p}^{k-1}\|_Q^2 + \|\overline{p}^k - \overline{p}^{k-1}\|_Q^2). \tag{4.48}$$

We then obtain from (4.43) and (4.48) that

$$\|\overline{\lambda}^k\|_Q^2 + r^2\|\overline{p}^{k-1}\|_Q^2 - (\|\overline{\lambda}^{k+1}\|_Q^2 + r^2\|\overline{p}^k\|_Q^2) \geq r^2\|\overline{p}^k - \nabla \overline{u}^k\|_Q^2 + r^2\|\overline{p}^k - \overline{p}^{k-1}\|_Q^2. \tag{4.49}$$

Equation (4.49) indicates

$$\begin{cases} \{\lambda^k : \forall k\}, \{p^k : \forall k\}, \text{ and } \{\nabla u^k : \forall k\} \text{ are bounded,} \\ \lim\limits_{k\to\infty} \|p^k - \nabla u^k\|_Q = 0, \\ \lim\limits_{k\to\infty} \|p^k - p^{k-1}\|_Q = 0. \end{cases} \tag{4.50}$$

On the other hand, since $(u,p;\lambda)$ is a saddle-point of $\mathscr{L}_{\rm rof}(v,q;\mu)$, we have

$$G_{\rm rof}(u,p) \le G_{\rm rof}(u^k,p^k) + (\lambda, p^k - \nabla u^k)_Q + \frac{r}{2}\|p^k - \nabla u^k\|_Q^2. \tag{4.51}$$

If we take $v = u$ in (4.40) and $q = p$ in (4.41), we have, by addition,

$$G_{\rm rof}(u,p) \ge G_{\rm rof}(u^k,p^k) + (\lambda^k, p^k - \nabla u^k)_Q + r\|p^k - \nabla u^k\|_Q^2 + r(\nabla \overline{u}^k, \overline{p}^k - \overline{p}^{k-1})_Q. \tag{4.52}$$

Using (4.50), we have

$$\liminf G_{\rm rof}(u^k,p^k) \ge G_{\rm rof}(u,p) \ge \limsup G_{\rm rof}(u^k,p^k), \tag{4.53}$$

by taking lim inf in (4.51) and lim sup in (4.52). This completes the proof of (4.35). Starting from (4.35), one can verify (4.36) in a similar way as in the proof of Theorem 4.3. ∎

There are some other proofs for Theorem 4.3 in [2, 3] and for Theorem 4.4 in [20, 3, 10]. Results similar to Theorem 4.4 were also obtained in [37] by identifying the algorithm as a special case of Douglas–Rachford splitting on the dual problem. The proof using the alternating Douglas–Rachford splitting relies on the properties of maximal monotone operators which appear in the optimality condition of the optimization problem, while our proof relies on the convexity of the objective functionals.

We recall that the operator K is invertible in many cases, e.g., image denoising where $K = I$ and most image deblurring problems (although the condition number of the blur kernel may be very bad). In these cases, Theorems 4.3 and 4.4 imply the convergence of the sequence $\{u^k\}$ (either $L \to \infty$ or $L = 1$ in Algorithm 4.2) to the unique solution of the problem.

5. Relations between the augmented Lagrangian method, dual methods, and split Bregman iteration for the ROF model.

In this section, we show, in the discrete setting, that CGM [16] and Chambolle's dual method [12] for the ROF model are closely connected to the augmented Lagrangian method. Also, we explain that the split Bregman iteration [24] is equivalent to Algorithm 4.1.

For the saddle-point problem (4.2), we have the following optimality conditions:

$$\partial_v \mathscr{L}_{\rm rof}(v,q;\mu)|_{(u,p;\lambda)} = \alpha K^*(Ku - f) + {\rm div}\lambda + r{\rm div}(p - \nabla u) = 0, \tag{5.1}$$

$$\partial_q \mathscr{L}_{\rm rof}(v,q;\mu)|_{(u,p;\lambda)} = \partial R_{\rm rof}(p) + \lambda + r(p - \nabla u) \ni 0, \tag{5.2}$$

$$\partial_\mu \mathscr{L}_{\rm rof}(v,q;\mu)|_{(u,p;\lambda)} = p - \nabla u = 0, \tag{5.3}$$

where $\partial R_{\rm rof}(p)$ is the subdifferential of $R_{\rm rof}$ at p.

It is definitely true that various techniques such as Newton and quasi-Newton linearizations can be applied to the above system of optimality conditions to solve the saddle-point problem. Actually CGM and Chambolle's dual method can be seen as directly solving simplified forms of the same optimality conditions (5.1), (5.2), and (5.3), as discussed in the following.

5.1. Connection to the CGM.

We show how to obtain the CGM from the augmented Lagrangian method. Using (5.3), we get $p = \nabla u$, which gives

$$\lambda_{i,j} = \begin{cases} -\frac{(\nabla u)_{i,j}}{|(\nabla u)_{i,j}|} & \text{if } |(\nabla u)_{i,j}| \neq 0, \\ g \in \mathbb{R}^2, |g| \le 1 & \text{if } |(\nabla u)_{i,j}| = 0, \end{cases} \tag{5.4}$$

from (5.2). Therefore, the multiplier λ is the dual variable in the CGM with a different sign. We then can reformulate the system of (5.1), (5.2), and (5.3) to be

$$\begin{aligned} \operatorname{div}\lambda + \alpha K^*(Ku - f) &= 0, \\ \nabla u + \lambda|\nabla u| &= 0, \\ |\lambda_{i,j}| \le 1 \quad \forall i,j, \end{aligned} \tag{5.5}$$

which is just the primal-dual system in [16] if $-\lambda$ is replaced with ω.

5.2. Connection to Chambolle's dual method. In the following we show how Chambolle's algorithm is connected to the augmented Lagrangian method. Compared to the derivation in [12], this is another way to obtain the dual method. From the system of (5.1), (5.2), and (5.3), we first eliminate the p variable to obtain (5.5), where we solve for u in the first equation as (assuming $\mathrm{Null}(K) = \{0\}$)

$$u = (\alpha K^*K)^{-1}(\alpha K^* f - \operatorname{div}\lambda), \tag{5.6}$$

and plug it into the second equation to obtain

$$\nabla((K^*K)^{-1}(\alpha K^* f - \operatorname{div}\lambda)) + \lambda|\nabla((K^*K)^{-1}(\alpha K^* f - \operatorname{div}\lambda))| = 0. \tag{5.7}$$

For image denoising problems where $K = I$, (5.7) and (5.6) are just the equations used by Chambolle in [12] to solve the dual variable and recover the primal variable u, respectively. Equation (5.7) for the dual variable in [12] was obtained through a derivation different from ours. Our derivation seems easier to follow. In addition, (5.6) and (5.7) derived here are formulations for general K. We should mention that K is sometimes compact, and thus the condition number of K^*K is very bad. In this case the algorithm is not efficient.

5.3. Connection to the split Bregman iteration. As noticed in [39, 37, 22] for the ROF model, the split Bregman iteration is equivalent to the augmented Lagrangian method. Here we briefly review the explanation by using the new notation in the discrete setting. Considering the zero initialization for the subgradients and the Lagrange multiplier and letting

$$(g_u^{k-1}, g_p^{k-1}) = -(\operatorname{div}\lambda^k, \lambda^k) \tag{5.8}$$

for $k = 0, 1, 2, \ldots$, we have

$$\begin{aligned} (u^k, p^k) &= \arg\min_{u,p} D_{G_{\mathrm{rof}}}^{(g_u^{k-1}, g_p^{k-1})}((u,p),(u^{k-1},p^{k-1})) + \frac{r}{2}\|p - \nabla u\|_Q^2 \\ &= \arg\min_{u,p} R_{\mathrm{rof}}(p) + \frac{\alpha}{2}\|Ku - f\|_V^2 + (u, \operatorname{div}\lambda^k)_V + (\lambda^k, p)_Q + \frac{r}{2}\|p - \nabla u\|_Q^2 \\ &= \arg\min_{u,p} R_{\mathrm{rof}}(p) + \frac{\alpha}{2}\|Ku - f\|_V^2 - (\lambda^k, \nabla u)_Q + (\lambda^k, p)_Q + \frac{r}{2}\|p - \nabla u\|_Q^2 \\ &= \arg\min_{u,p} \mathscr{L}_{\mathrm{rof}}(u, p; \lambda^k), \end{aligned}$$

indicating the equivalence between the solutions of the Bregman iteration and the augmented Lagrangian method and that the updates of the subgradients are also consistent with the Lagrange multiplier update if the subproblems in these two methods are solved identically.

6. Extension to the vectorial TV model. In this section, we extend our method and observations to a vectorial TV restoration model. Let us denote in general an M-channel image by $\mathbf{u} = (u_1, u_2, \dots, u_M)$, where $u_m \in V \ \forall m = 1, 2, \dots, M$. The intensity at pixel (i, j) is thus multivalued, say, $\mathbf{u}_{i,j} = ((u_1)_{i,j}, (u_2)_{i,j}, \dots, (u_M)_{i,j})$. If $M = 3$, one gets usual color models such as RGB.

For the convenience of description, we introduce the following notation:

$$\mathbf{V} = \underbrace{V \times V \times \cdots \times V}_{M},$$
$$\mathbf{Q} = \underbrace{Q \times Q \times \cdots \times Q}_{M}.$$

Hence an M-channel image $\mathbf{u}$ is an element of $\mathbf{V}$, and its gradient $\nabla\mathbf{u} = (\nabla u_1, \nabla u_2, \dots, \nabla u_M)$ is an element of $\mathbf{Q}$. The usual inner products and norms in $\mathbf{V}$ and $\mathbf{Q}$ are as follows:

$$(\mathbf{u}, \mathbf{v})_{\mathbf{V}} = \sum_{1 \le m \le M} (u_m, v_m)_V, \quad \|\mathbf{u}\|_{\mathbf{V}} = \sqrt{(\mathbf{u}, \mathbf{u})_{\mathbf{V}}};$$
$$(\mathbf{p}, \mathbf{q})_{\mathbf{Q}} = \sum_{1 \le m \le M} (p_m, q_m)_Q, \quad \|\mathbf{p}\|_{\mathbf{Q}} = \sqrt{(\mathbf{p}, \mathbf{p})_{\mathbf{Q}}}.$$

For $\mathbf{u} \in \mathbf{V}$ and $\mathbf{p} \in \mathbf{Q}$, we also define the following pixel-by-pixel norms:

$$|\mathbf{u}_{i,j}| = \sqrt{\sum_{1 \le m \le M} (u_m)_{i,j}^2}$$

and

$$|\mathbf{p}_{i,j}| = \sqrt{\sum_{1 \le m \le M} |(p_m)_{i,j}|^2}$$

at each pixel (i, j).

We consider the following vector-valued image restoration problem:

$$\min_{\mathbf{u} \in \mathbf{V}} \left\{ F_{\text{vtv}}(\mathbf{u}) = R_{\text{vtv}}(\nabla\mathbf{u}) + \frac{\alpha}{2} \|\mathbf{K}\mathbf{u} - \mathbf{f}\|_{\mathbf{V}}^2 \right\}, \tag{6.1}$$

where

$$R_{\text{vtv}}(\nabla\mathbf{u}) = \text{TV}(\mathbf{u}) = \sum_{1 \le i,j \le N} \sqrt{\sum_{1 \le m \le M} |(\nabla u_m)_{i,j}|^2} \tag{6.2}$$

is the vectorial TV seminorm [35, 6] (see [4] for some other choices), and $\mathbf{f} = (f_1, f_2, \dots, f_M) \in \mathbf{V}$ is an observed image and $\mathbf{K} = (K_{i,j})_{M \times M} : \mathbf{V} \to \mathbf{V}$ is the blur operator. In $\mathbf{K}$, each $K_{i,j}$ is a blur kernel representing a convolution. The diagonal elements of $\mathbf{K}$ denote within-channel blurs, whereas the off-diagonal elements describe cross-channel blurs. Similarly as for the ROF model, here we make the following assumption:

- $\text{Null}(\nabla) \cap \text{Null}(\mathbf{K}) = \{0\}$.

Under this assumption, the functional $F_{\text{vtv}}(\mathbf{u})$ in (6.1) is convex, proper, coercive, and continuous. Therefore we have the following result.

Theorem 6.1. *The problem* (6.1) *has at least one solution* $\mathbf{u}$*, which satisfies*

$$0 \in \alpha \mathbf{K}^*(\mathbf{K}\mathbf{u} - \mathbf{f}) - \operatorname{div} \partial R_{\text{vtv}}(\nabla \mathbf{u}), \tag{6.3}$$

where $\partial R_{\text{vtv}}(\nabla \mathbf{u})$ *is the subdifferential of* R_{vtv} *at* $\nabla \mathbf{u}$*. Moreover, if* $\text{Null}(\mathbf{K}) = \{0\}$*, the minimizer is unique.*

By introducing a new variable $\mathbf{p} = (p_1, p_2, \ldots, p_M) \in \mathbf{Q}$, the minimization problem (6.1) is equivalent to the following constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{u}\in\mathbf{V},\mathbf{p}\in\mathbf{Q}} \{G_{\text{vtv}}(\mathbf{u},\mathbf{p}) = R_{\text{vtv}}(\mathbf{p}) + \tfrac{\alpha}{2}\|\mathbf{K}\mathbf{u} - \mathbf{f}\|_{\mathbf{V}}^2\} \\ &\text{s.t.} \qquad \mathbf{p} = \nabla \mathbf{u}. \end{aligned} \tag{6.4}$$

6.1. Augmented Lagrangian method. Here we present the augmented Lagrangian method for the restoration problem (6.1) or, equivalently, (6.4). We first define the augmented Lagrangian functional as

$$\mathscr{L}_{\text{vtv}}(\mathbf{v},\mathbf{q};\mu) = R_{\text{vtv}}(\mathbf{q}) + \frac{\alpha}{2}\|\mathbf{K}\mathbf{v} - \mathbf{f}\|_{\mathbf{V}}^2 + (\mu, \mathbf{q} - \nabla \mathbf{v})_{\mathbf{Q}} + \frac{r}{2}\|\mathbf{q} - \nabla \mathbf{v}\|_{\mathbf{Q}}^2, \tag{6.5}$$

where $\mu \in \mathbf{Q}$ is the multiplier and r is a positive constant. The augmented Lagrangian method aims at solving the following saddle-point problem:

$$\begin{aligned} &\text{Find} \quad (\mathbf{u},\mathbf{p};\lambda) \in \mathbf{V}\times\mathbf{Q}\times\mathbf{Q} \\ &\text{s.t.} \quad \mathscr{L}_{\text{vtv}}(\mathbf{u},\mathbf{p};\mu) \le \mathscr{L}_{\text{vtv}}(\mathbf{u},\mathbf{p};\lambda) \le \mathscr{L}_{\text{vtv}}(\mathbf{v},\mathbf{q};\lambda)\ \forall (\mathbf{v},\mathbf{q};\mu) \in \mathbf{V}\times\mathbf{Q}\times\mathbf{Q}. \end{aligned} \tag{6.6}$$

Similarly to Theorem 4.1, we have the following result.

Theorem 6.2. $\mathbf{u} \in \mathbf{V}$ *is a solution of* (6.1) *if and only if there exist* $\mathbf{p} \in \mathbf{Q}$ *and* $\lambda \in \mathbf{Q}$ *such that* $(\mathbf{u},\mathbf{p};\lambda)$ *is a solution of* (6.6).

We use an iterative procedure as described in Algorithm 6.1 to solve the problem (6.6). Again, one may see the $\approx$ in (6.7). This is because the minimization problem (6.7) has two coupled variables and hence is difficult to solve exactly.

Algorithm 6.1. Augmented Lagrangian method for the vectorial TV model.

1. *Initialization*: $\lambda^0 = 0$;
2. *For* $k = 0, 1, 2, \ldots$: Compute $(\mathbf{u}^k, \mathbf{p}^k)$ from

$$(\mathbf{u}^k,\mathbf{p}^k) \approx \arg\min_{(\mathbf{v},\mathbf{q})\in(\mathbf{V},\mathbf{Q})} \mathscr{L}_{\text{vtv}}(\mathbf{v},\mathbf{q};\lambda^k) \tag{6.7}$$

and update

$$\lambda^{k+1} = \lambda^k + r(\mathbf{p}^k - \nabla \mathbf{u}^k). \tag{6.8}$$

As for the minimization problem (6.7), we separate it into the following two subproblems:

$$\min_{\mathbf{v}} \frac{\alpha}{2}\|\mathbf{K}\mathbf{v} - \mathbf{f}\|_{\mathbf{V}}^2 - (\lambda^k, \nabla \mathbf{v})_{\mathbf{Q}} + \frac{r}{2}\|\mathbf{q} - \nabla \mathbf{v}\|_{\mathbf{Q}}^2 \tag{6.9}$$

for a given $\mathbf{q}$, and

$$\min_{\mathbf{q}} R_{\text{vtv}}(\mathbf{q}) + (\lambda^k, \mathbf{q})_{\mathbf{Q}} + \frac{r}{2}\|\mathbf{q} - \nabla\mathbf{v}\|_{\mathbf{Q}}^2 \tag{6.10}$$

for a given $\mathbf{v}$.

Applying Fourier transforms to the optimality condition of the subproblem (6.9), we have

$$\alpha\mathcal{F}(\mathbf{K}^*)\mathcal{F}(\mathbf{K})\mathcal{F}(\mathbf{v}) - r\mathcal{F}(\triangle)\mathcal{F}(\mathbf{v}) = \alpha\mathcal{F}(\mathbf{K}^*)\mathcal{F}(\mathbf{f}) - \mathcal{F}(\text{div})\mathcal{F}(\lambda^k) - r\mathcal{F}(\text{div})\mathcal{F}(\mathbf{q}), \tag{6.11}$$

from which $\mathcal{F}(\mathbf{v})$ can be found and then $\mathbf{v}$ via an inverse Fourier transform. Here applying Fourier transforms to a matrix or a vector is regarded as applying Fourier transforms to its components, e.g., $\mathcal{F}(\mathbf{v}) = (\mathcal{F}(v_1), \mathcal{F}(v_2), \ldots, \mathcal{F}(v_M))$, $\mathcal{F}(\text{div})\mathcal{F}(\lambda^k) = (\mathcal{F}(\text{div})\mathcal{F}(\lambda_1^k)$, $\mathcal{F}(\text{div})\mathcal{F}(\lambda_2^k), \ldots, \mathcal{F}(\text{div})\mathcal{F}(\lambda_M^k))$, $\mathcal{F}(\mathbf{K}) = (\mathcal{F}(K_{i,j}))_{M\times M}$. Thus one needs to solve a system of linear algebraic equations of $\mathcal{F}(\mathbf{v}) = (\mathcal{F}(v_1), \mathcal{F}(v_2), \ldots, \mathcal{F}(v_M))$ since cross blurs exist in general. In a special case without cross blurs, the blur kernel matrix $\mathbf{K}$ is a diagonal matrix, and the $\mathcal{F}(\mathbf{v})$ can be calculated component by component. The subproblem (6.10) has the following closed form solution:

$$\mathbf{q}_{i,j} = \begin{cases} \left(1 - \frac{1}{r}\frac{1}{|\mathbf{w}_{i,j}|}\right)\mathbf{w}_{i,j}, & |\mathbf{w}_{i,j}| > \frac{1}{r}, \\ 0, & |\mathbf{w}_{i,j}| \le \frac{1}{r}, \end{cases} \tag{6.12}$$

where

$$\mathbf{w} = \nabla\mathbf{v} - \frac{\lambda^k}{r}. \tag{6.13}$$

We then have an iterative procedure to alternatively compute the $\mathbf{v}$ and $\mathbf{q}$ according to (6.11) (6.12); see Algorithm 6.2.

Algorithm 6.2. Augmented Lagrangian method for the vectorial TV model—solve the minimization problem (6.7).

- *Initialization*: $\mathbf{u}^{k,0} = \mathbf{u}^{k-1}, \mathbf{p}^{k,0} = \mathbf{p}^{k-1}$;
- *For* $l = 0, 1, 2, \ldots, L-1$: Compute $\mathbf{u}^{k,l+1}$ from (6.11) for $\mathbf{q} = \mathbf{p}^{k,l}$, and then compute $\mathbf{p}^{k,l+1}$ from (6.12) for $\mathbf{v} = \mathbf{u}^{k,l+1}$;
- $\mathbf{u}^k = \mathbf{u}^{k,L}, \mathbf{p}^k = \mathbf{p}^{k,L}$.

Here L can be chosen using some convergence test techniques and is usually simply set to be 1.

In the following we present some convergence results without giving proofs. They are straightforward generalizations of Theorems 4.2, 4.3, and 4.4.

Theorem 6.3. *The sequence* $\{(\mathbf{u}^{k,l}, \mathbf{p}^{k,l}) : l = 0, 1, 2, \ldots\}$ *generated by Algorithm 6.2 converges to a solution of the problem* (6.7).

Theorem 6.4. *Assume that* $(\mathbf{u}, \mathbf{p}; \lambda)$ *is a saddle-point of* $\mathscr{L}_{\text{vtv}}(\mathbf{v}, \mathbf{q}; \mu)$. *Suppose that the minimization problem* (6.7) *is exactly solved in each iteration; i.e.,* $L \to \infty$ *in Algorithm 6.2. Then the sequence* $(\mathbf{u}^k, \mathbf{p}^k; \lambda^k)$ *generated by Algorithm 6.1 satisfies*

$$\begin{cases} \lim\limits_{k\to\infty} G_{\text{vtv}}(\mathbf{u}^k, \mathbf{p}^k) = G_{\text{vtv}}(\mathbf{u}, \mathbf{p}), \\ \lim\limits_{k\to\infty} \|\mathbf{p}^k - \nabla\mathbf{u}^k\|_{\mathbf{Q}} = 0. \end{cases} \tag{6.14}$$

Since $R_{\mathrm{vtv}}(\mathbf{p})$ *is continuous,* (6.14) *indicates that* $\mathbf{u}^k$ *is a minimizing sequence of* F_{vtv}*. If we further have* $\mathrm{Null}(\mathbf{K}) = \{0\}$*, then*

$$\begin{cases} \lim\limits_{k\to\infty} \mathbf{u}^k = \mathbf{u}, \\ \lim\limits_{k\to\infty} \mathbf{p}^k = \mathbf{p}. \end{cases} \tag{6.15}$$

Theorem 6.5. *Assume that* $(\mathbf{u}, \mathbf{p}; \lambda)$ *is a saddle-point of* $\mathscr{L}_{\mathrm{vtv}}(\mathbf{v}, \mathbf{q}; \mu)$*. Suppose that the minimization problem* (6.7) *is roughly solved in each iteration; i.e.,* $L = 1$ *in Algorithm* 6.2*. Then the sequence* $(\mathbf{u}^k, \mathbf{p}^k; \lambda^k)$ *generated by Algorithm* 6.1 *satisfies*

$$\begin{cases} \lim\limits_{k\to\infty} G_{\mathrm{vtv}}(\mathbf{u}^k, \mathbf{p}^k) = G_{\mathrm{vtv}}(\mathbf{u}, \mathbf{p}), \\ \lim\limits_{k\to\infty} \|\mathbf{p}^k - \nabla \mathbf{u}^k\|_{\mathbf{Q}} = 0. \end{cases} \tag{6.16}$$

Since $R_{\mathrm{vtv}}(\mathbf{p})$ *is continuous,* (6.16) *indicates that* $\mathbf{u}^k$ *is a minimizing sequence of* F_{vtv}*. If we further have* $\mathrm{Null}(\mathbf{K}) = \{0\}$*, then*

$$\begin{cases} \lim\limits_{k\to\infty} \mathbf{u}^k = \mathbf{u}, \\ \lim\limits_{k\to\infty} \mathbf{p}^k = \mathbf{p}. \end{cases} \tag{6.17}$$

6.2. Dual methods for vectorial TV model. In this subsection we discuss the formulations of CGM and Chambolle's dual method for the vectorial TV restoration model.

We start from the optimality conditions of the saddle-point problem (6.6), which read

$$\partial_{\mathbf{v}} \mathscr{L}_{\mathrm{vtv}}(\mathbf{v}, \mathbf{q}, \mu)|_{(\mathbf{u},\mathbf{p};\lambda)} = \alpha \mathbf{K}^*(\mathbf{K}\mathbf{u} - \mathbf{f}) + \mathrm{div}\lambda + r\,\mathrm{div}(\mathbf{p} - \nabla \mathbf{u}) = 0, \tag{6.18}$$

$$\partial_{\mathbf{q}} \mathscr{L}_{\mathrm{vtv}}(\mathbf{v}, \mathbf{q}, \mu)|_{(\mathbf{u},\mathbf{p};\lambda)} = \partial R_{\mathrm{vtv}}(\mathbf{p}) + \lambda + r(\mathbf{p} - \nabla \mathbf{u}) \ni 0, \tag{6.19}$$

$$\partial_{\mu} \mathscr{L}_{\mathrm{vtv}}(\mathbf{v}, \mathbf{q}, \mu)|_{(\mathbf{u},\mathbf{p};\lambda)} = \mathbf{p} - \nabla \mathbf{u} = 0, \tag{6.20}$$

where $\partial R_{\mathrm{vtv}}(\mathbf{p})$ is the subdifferential of R_{vtv} at $\mathbf{p}$, and $\mathrm{div}\lambda = (\mathrm{div}\lambda_1, \mathrm{div}\lambda_2, \ldots, \mathrm{div}\lambda_M)$ as well as $\mathrm{div}(\mathbf{p} - \nabla \mathbf{u})$ means similarly.

6.2.1. The CGM. The CGM for color image restoration is still missing in the literature, as pointed out in [6]: "the question of extension is open for the CGM's model." Here we present the method via simplifying the optimality conditions of the saddle-point problem (6.6). Using (6.20) to eliminate $\mathbf{p}$ and rearranging the result yields

$$\begin{aligned} \alpha \mathbf{K}^*(\mathbf{K}\mathbf{u} - \mathbf{f}) + \mathrm{div}\lambda &= 0, \\ \nabla \mathbf{u} + \lambda |\nabla \mathbf{u}| &= 0, \\ |\lambda_{i,j}| \le 1 \quad \forall i, j, \end{aligned} \tag{6.21}$$

which is a system similar to that in the CGM in [16]. In our implementation we use Newton's linearization techniques to simultaneously compute the primal and dual variables $\mathbf{u}$ and λ in (6.21). We also adopt a continuation scheme on the smoothing parameter β based on the duality gap [49].

6.2.2. Chambolle's dual method. If we go a step further, we will get a method similar to Chambolle's [12]. See [6]. From the first equation of (6.21) we have a relation between the primal variable $\mathbf{u}$ and the dual variable λ as (assume here $\text{Null}(\mathbf{K}) = \{0\}$)

$$\mathbf{u} = (\alpha \mathbf{K}^* \mathbf{K})^{-1}(\alpha \mathbf{K}^* \mathbf{f} - \text{div}\lambda). \tag{6.22}$$

Substituting this equation into the second equation of (6.21) gives

$$\nabla((\mathbf{K}^*\mathbf{K})^{-1}(\alpha \mathbf{K}^* \mathbf{f} - \text{div}\lambda)) + \lambda |\nabla((\mathbf{K}^*\mathbf{K})^{-1}(\alpha \mathbf{K}^* \mathbf{f} - \text{div}\lambda))| = 0, \tag{6.23}$$

which can be solved with a semi-implicit gradient descent scheme. Here we derive Chambolle's dual method for the vectorial TV model in a way different from [6].

6.3. Split Bregman iteration for vectorial TV model. Split Bregman iteration for the restoration problem (6.4) is presented in Algorithm 6.3. Therein the minimization problem (6.24) can be solved using Algorithm 6.2. For the image denoising case, Gauss–Seidel iteration can be applied to the $\mathbf{u}$-subproblem (as in our implementation), which results in a more efficient algorithm than the FFT-based implementation. To the best of our knowledge, Algorithm 6.3 has not been proposed previously.

Algorithm 6.3. Split Bregman iteration for the vectorial TV model.

1. *Initialization*: $(\mathbf{u}^{-1}, \mathbf{p}^{-1}) = (0, 0)$, $(\mathbf{g}_{\mathbf{u}}^{-1}, \mathbf{g}_{\mathbf{p}}^{-1}) = (0, 0)$;
2. *For* $k = 0, 1, 2, \ldots$: Compute $(\mathbf{u}^k, \mathbf{p}^k)$ from

$$(\mathbf{u}^k, \mathbf{p}^k) = \arg\min_{(\mathbf{u},\mathbf{p})} D_{G_{\text{vtv}}}^{(\mathbf{g}_{\mathbf{u}}^{k-1}, \mathbf{g}_{\mathbf{p}}^{k-1})}\left((\mathbf{u}, \mathbf{p}), (\mathbf{u}^{k-1}, \mathbf{p}^{k-1})\right) + \frac{r}{2}\|\mathbf{p} - \nabla \mathbf{u}\|_{\mathbf{Q}}^2, \tag{6.24}$$

and update

$$\begin{aligned} \mathbf{g}_{\mathbf{u}}^k &= \mathbf{g}_{\mathbf{u}}^{k-1} - r\,\text{div}(\mathbf{p}^k - \nabla \mathbf{u}^k), \\ \mathbf{g}_{\mathbf{p}}^k &= \mathbf{g}_{\mathbf{p}}^{k-1} - r(\mathbf{p}^k - \nabla \mathbf{u}^k). \end{aligned} \tag{6.25}$$

With an argument similar to that in the last section for the ROF model, one can show the equivalence between the augmented Lagrangian method (Algorithm 6.1) and the split Bregman iteration (Algorithm 6.3), provided the subproblems are solved identically.

7. Extension to high order models. We can also extend our method and observations to high order models. As is well known, the TV restoration models (e.g., ROF and vectorial TV) suffer from a staircase effect; see [47, 42, 18, 15, 7] and references therein. To overcome this, high order models have been proposed [13, 18, 46, 29, 30, 14, 26]. Here we take the Lysaker–Lundervold–Tai (LLT) model [29] as an example. Other high order models can be treated similarly. Moreover, we present the model (which is still denoted as LLT) and method for multivalued images, for generality.

Since the LLT model is defined using second order derivatives, we need to introduce second

order difference operators. Given $u \in V$, we define

$$\begin{aligned}(\mathring{D}_{xx}^{-+}u)_{i,j} &:= (\mathring{D}_x^-(\mathring{D}_x^+u))_{i,j},\\ (\mathring{D}_{xy}^{++}u)_{i,j} &:= (\mathring{D}_x^+(\mathring{D}_y^+u))_{i,j},\\ (\mathring{D}_{yx}^{++}u)_{i,j} &:= (\mathring{D}_y^+(\mathring{D}_x^+u))_{i,j},\\ (\mathring{D}_{yy}^{-+}u)_{i,j} &:= (\mathring{D}_y^-(\mathring{D}_y^+u))_{i,j},\end{aligned}$$

based on the first order difference operators introduced in section 2. One may verify that, by the definition above, $(\mathring{D}_{xy}^{++}u)_{i,j} = (\mathring{D}_{yx}^{++}u)_{i,j}$ holds. In this paper we also use other second order difference operators such as $\mathring{D}_{xx}^{+-}, \mathring{D}_{xy}^{+-}, \mathring{D}_{xy}^{-+}, and\ \mathring{D}_{xy}^{--}$. They can be similarly defined, and we omit the details. We now denote the discrete Hessian of u as

$$Hu = \begin{pmatrix} \mathring{D}_{xx}^{-+}u & \mathring{D}_{xy}^{++}u \\ \mathring{D}_{yx}^{++}u & \mathring{D}_{yy}^{-+}u \end{pmatrix} \in Q_2$$

with

$$(Hu)_{i,j} = \begin{pmatrix} (\mathring{D}_{xx}^{-+}u)_{i,j} & (\mathring{D}_{xy}^{++}u)_{i,j} \\ (\mathring{D}_{yx}^{++}u)_{i,j} & (\mathring{D}_{yy}^{-+}u)_{i,j} \end{pmatrix},$$

where

$$Q_2 = V \times V \times V \times V.$$

We point out that there are actually $3 \times 4 \times 3$ symmetric discrete Hessians by different combinations of all the second order difference operators. Here we use just one of those defined above. For a vector-valued image $\mathbf{u} = (u_1, u_2, \ldots, u_M) \in \mathbf{V}$, the Hessian is computed channel by channel and denoted as

$$H\mathbf{u} = (Hu_1, Hu_2, \ldots, Hu_M) \in \mathbf{Q}_2,$$

where

$$\mathbf{Q}_2 = \underbrace{Q_2 \times \cdots \times Q_2}_{M}.$$

Given

$$\mathbf{p} = \left(\begin{pmatrix} p_1^{11} & p_1^{12} \\ p_1^{21} & p_1^{22} \end{pmatrix}, \begin{pmatrix} p_2^{11} & p_2^{12} \\ p_2^{21} & p_2^{22} \end{pmatrix}, \ldots, \begin{pmatrix} p_M^{11} & p_M^{12} \\ p_M^{21} & p_M^{22} \end{pmatrix} \right) \in \mathbf{Q}_2$$

and

$$\mathbf{q} = \left(\begin{pmatrix} q_1^{11} & q_1^{12} \\ q_1^{21} & q_1^{22} \end{pmatrix}, \begin{pmatrix} q_2^{11} & q_2^{12} \\ q_2^{21} & q_2^{22} \end{pmatrix}, \ldots, \begin{pmatrix} q_M^{11} & q_M^{12} \\ q_M^{21} & q_M^{22} \end{pmatrix} \right) \in \mathbf{Q}_2,$$

the inner product and norm in the space $\mathbf{Q}_2$ are as follows:

$$(\mathbf{p}, \mathbf{q})_{\mathbf{Q}_2} = \sum_{1 \le m \le M} ((p_m^{11}, q_m^{11})_V + (p_m^{12}, q_m^{12})_V + (p_m^{21}, q_m^{21})_V + (p_m^{22}, q_m^{22})_V),$$

$$\|\mathbf{p}\|_{\mathbf{Q}_2} = \sqrt{(\mathbf{p}, \mathbf{p})_{\mathbf{Q}_2}}.$$

Similarly with those in the ROF and vectorial TV models, we mention the following pixel-by-pixel norm:

$$|\mathbf{p}_{i,j}| = \sqrt{\sum_{1\le m\le M} ((p_m^{11})_{i,j}^2 + (p_m^{12})_{i,j}^2 + (p_m^{21})_{i,j}^2 + (p_m^{22})_{i,j}^2)}.$$

By regarding the Hessian as an operator $H : \mathbf{V} \to \mathbf{Q}_2$, we find its adjoint operator $H^* : \mathbf{Q}_2 \to \mathbf{V}$ as

$$H^*(\mathbf{p}) = (H^*p_1, H^*p_2, \dots, H^*p_M),$$

where

$$H^*p_m = \mathring{D}_{xx}^{+-}p_m^{11} + \mathring{D}_{yx}^{--}p_m^{12} + \mathring{D}_{xy}^{--}p_m^{21} + \mathring{D}_{yy}^{+-}p_m^{22}.$$

We then consider the following image restoration problem:

$$\min_{\mathbf{u}\in\mathbf{V}} \left\{ F_{\mathrm{llt}}(\mathbf{u}) = R_{\mathrm{llt}}(H\mathbf{u}) + \frac{\alpha}{2}\|\mathbf{K}\mathbf{u} - \mathbf{f}\|_{\mathbf{V}}^2 \right\}, \tag{7.1}$$

where

$$R_{\mathrm{llt}}(H\mathbf{u}) = \sum_{1\le i,j\le N} |(H\mathbf{u})_{i,j}|,$$

$\mathbf{f} \in \mathbf{V}$ is the observed image, and $\mathbf{K} : \mathbf{V} \to \mathbf{V}$ is the blur operator which is explained in section 6. We have the following assumption:

- $\mathrm{Null}(H) \cap \mathrm{Null}(\mathbf{K}) = \{0\}$.

Given this, the functional $F_{\mathrm{llt}}(\mathbf{u})$ in (7.1) is convex, proper, coercive, and continuous. According to the generalized Weierstrass theorem and Fermat's theorem, problem (7.1) has at least one solution $\mathbf{u}$, which is characterized by

$$0 \in \alpha\mathbf{K}^*(\mathbf{K}\mathbf{u} - \mathbf{f}) + H^*\partial R_{\mathrm{llt}}(H\mathbf{u}), \tag{7.2}$$

where $\partial R_{\mathrm{llt}}(H\mathbf{u})$ is the subdifferential of R_{llt} at $H\mathbf{u}$. Moreover, if $\mathrm{Null}(\mathbf{K}) = \{0\}$, the minimizer is unique.

In the following we present an augmented Lagrangian method to solve (7.1). We give only the algorithm. Convergence results and connections to Chambolle's dual method [38] are similar to those in previous sections. It is also quite straightforward to derive the other two new methods, i.e., the CGM and the split Bregman iteration for this problem, by following our observations in the previous section. Here we omit these details.

We first reformulate (7.1) to be the following constrained optimization problem:

$$\begin{aligned} \min_{\mathbf{u}\in\mathbf{V},\mathbf{p}\in\mathbf{Q}_2} & \left\{ G_{\mathrm{llt}}(\mathbf{u},\mathbf{p}) = R_{\mathrm{llt}}(\mathbf{p}) + \frac{\alpha}{2}\|\mathbf{K}\mathbf{u} - \mathbf{f}\|_{\mathbf{V}}^2 \right\} \\ \text{s.t.} \quad & \mathbf{p} = H\mathbf{u}. \end{aligned} \tag{7.3}$$

To solve (7.3), we define the augmented Lagrangian functional as

$$\mathscr{L}_{\mathrm{llt}}(\mathbf{v},\mathbf{q};\mu) = R_{\mathrm{llt}}(\mathbf{q}) + \frac{\alpha}{2}\|\mathbf{K}\mathbf{v} - \mathbf{f}\|_{\mathbf{V}}^2 + (\mu, \mathbf{q} - H\mathbf{v})_{\mathbf{Q}_2} + \frac{r}{2}\|\mathbf{q} - H\mathbf{v}\|_{\mathbf{Q}_2}^2, \tag{7.4}$$

where $\mu \in \mathbf{Q}_2$, and consider the following saddle-point problem:

$$\begin{aligned} &\text{Find} \quad (\mathbf{u},\mathbf{p};\lambda) \in \mathbf{V}\times\mathbf{Q}_2\times\mathbf{Q}_2 \\ &\text{s.t.} \quad \mathscr{L}_{\mathrm{llt}}(\mathbf{u},\mathbf{p};\mu) \le \mathscr{L}_{\mathrm{llt}}(\mathbf{u},\mathbf{p};\lambda) \le \mathscr{L}_{\mathrm{llt}}(\mathbf{v},\mathbf{q};\lambda) \quad \forall (\mathbf{v},\mathbf{q};\mu) \in \mathbf{V}\times\mathbf{Q}_2\times\mathbf{Q}_2. \end{aligned} \tag{7.5}$$

Similarly with Theorem 4.1, $\mathbf{u} \in \mathbf{V}$ is a solution of (7.1) if and only if there exist $\mathbf{p} \in \mathbf{Q}_2$ and $\lambda \in \mathbf{Q}_2$ such that $(\mathbf{u}, \mathbf{p}; \lambda)$ is a solution of (7.5). This can be shown by noting (7.2).

An iterative algorithm is given in Algorithm 7.1 to solve the saddle-point problem (7.5). To solve the minimization problem (7.8), we separate it into the following two subproblems:

$$\min_{\mathbf{v}} \frac{\alpha}{2}\|\mathbf{K}\mathbf{v} - \mathbf{f}\|_{\mathbf{V}}^2 - (\lambda^k, H\mathbf{v})_{\mathbf{Q}_2} + \frac{r}{2}\|\mathbf{q} - H\mathbf{v}\|_{\mathbf{Q}_2}^2 \tag{7.6}$$

for a given $\mathbf{q}$, and

$$\min_{\mathbf{q}} R_{\mathrm{llt}}(\mathbf{q}) + (\lambda^k, \mathbf{q})_{\mathbf{Q}_2} + \frac{r}{2}\|\mathbf{q} - H\mathbf{v}\|_{\mathbf{Q}_2}^2 \tag{7.7}$$

for a given $\mathbf{v}$.

Algorithm 7.1. Augmented Lagrangian method for the LLT model.

1. *Initialization*: $\lambda^0 = 0$;
2. *For* $k = 0, 1, 2, \ldots$: Compute $(\mathbf{u}^k, \mathbf{p}^k)$ from

$$(\mathbf{u}^k, \mathbf{p}^k) \approx \arg \min_{(\mathbf{v},\mathbf{q}) \in (\mathbf{V},\mathbf{Q}_2)} \mathscr{L}_{\mathrm{llt}}(\mathbf{v}, \mathbf{q}; \lambda^k), \tag{7.8}$$

and update

$$\lambda^{k+1} = \lambda^k + r(\mathbf{p}^k - H\mathbf{u}^k). \tag{7.9}$$

As with (6.11), (6.12), and (6.13), we give the solutions to (7.6) and (7.7) as follows. From the optimality condition of the subproblem (7.6) and using Fourier transforms, we deduce

$$\alpha\mathcal{F}(\mathbf{K}^*)\mathcal{F}(\mathbf{K})\mathcal{F}(\mathbf{v}) + r\mathcal{F}(H^*)\mathcal{F}(H)\mathcal{F}(\mathbf{v}) = \alpha\mathcal{F}(\mathbf{K}^*)\mathcal{F}(\mathbf{f}) + \mathcal{F}(H^*)\mathcal{F}(\lambda^k) + r\mathcal{F}(H^*)\mathcal{F}(\mathbf{q}), \tag{7.10}$$

from which $\mathcal{F}(\mathbf{v})$ and then $\mathbf{v}$ can be found. The subproblem (7.7) has the following closed form solution:

$$\mathbf{q}_{i,j} = \begin{cases} \left(1 - \frac{1}{r}\frac{1}{|\mathbf{w}_{i,j}|}\right)\mathbf{w}_{i,j}, & |\mathbf{w}_{i,j}| > \frac{1}{r}, \\ 0, & |\mathbf{w}_{i,j}| \le \frac{1}{r}, \end{cases} \tag{7.11}$$

where

$$\mathbf{w} = H\mathbf{v} - \frac{\lambda^k}{r}. \tag{7.12}$$

We then use an iterative procedure to alternatively calculate $\mathbf{v}$ and $\mathbf{q}$ according to (7.10) and (7.11); see Algorithm 7.2.

Here L can be chosen using some convergence test techniques and is usually simply set to be $L = 1$, as in the ROF and vectorial TV restoration problems.

Algorithm 7.2. Augmented Lagrangian method for the LLT model—solve the minimization problem (7.8).

- *Initialization*: $\mathbf{u}^{k,0} = \mathbf{u}^{k-1}, \mathbf{p}^{k,0} = \mathbf{p}^{k-1}$;
- *For* $l = 0, 1, 2, \ldots, L-1$: Compute $\mathbf{u}^{k,l+1}$ from (7.10) for $\mathbf{q} = \mathbf{p}^{k,l}$, and then compute $\mathbf{p}^{k,l+1}$ from (7.11) for $\mathbf{v} = \mathbf{u}^{k,l+1}$;
- $\mathbf{u}^k = \mathbf{u}^{k,L}, \mathbf{p}^k = \mathbf{p}^{k,L}$.

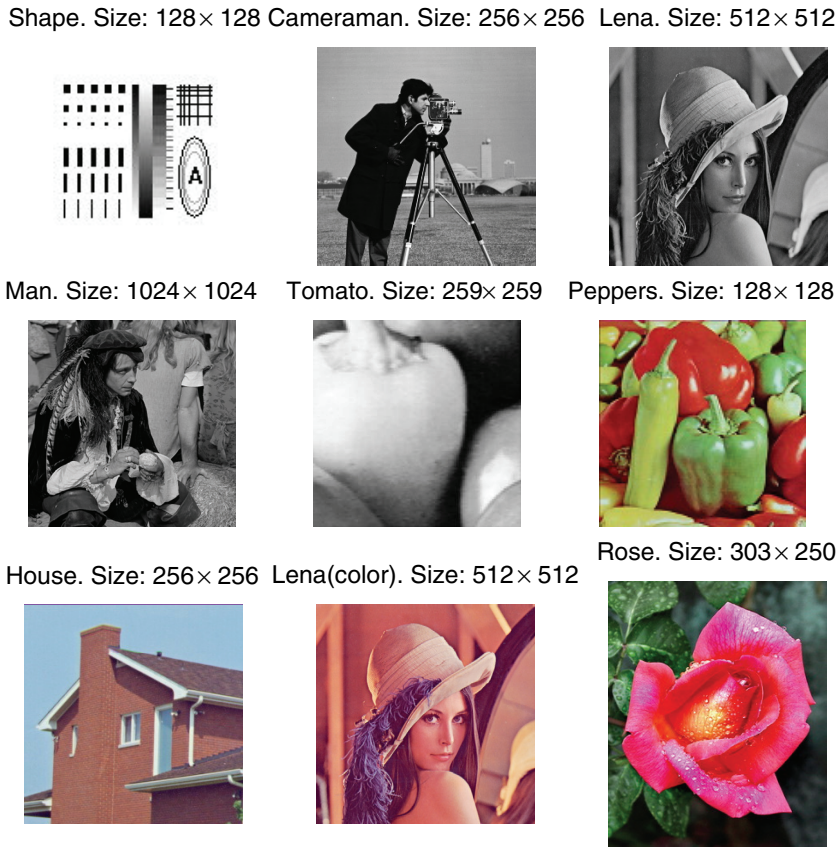

Figure 1. *The test images used in this paper.*

8. Examples and discussion. In this section we provide some numerical examples. We will first present some results of the augmented Lagrangian method (ALM) applied to ROF, vectorial TV, and LLT models, and then compare ALM to the CGM, Chambolle's dual method, the split Bregman iteration (SB) based on Gauss–Seidel iteration, MATLAB built-in image deblurring functions, and fast TV deconvolution (FTVd). All the experiments were performed using Windows Vista and MATLAB R2007a (Version 7.4.0.287) on a desktop with Intel CPU (Core 2) at 2.13GHz and 2GB memory.

8.1. Test images, practical implementation, and stopping condition. In this paper, we tested lots of images, including the well-known Cameraman, Lena, etc. See Figure 1 where

the image sizes are also given.

We used the MATLAB functions *imfilter* and *imfilter33* to generate blurring effects for grayscale and color images, respectively. For the grayscale image case, we tested three kinds of blurring kernels: Gaussian, motion, and average. For the convenience of description, we denote the Gaussian blur with a blurring size *size* and a standard deviation *sigma* as (G, *size*, *sigma*). The motion blur with a motion length *len* and an angle *theta* is denoted as (M, *len*, *theta*). Similarly, the average blur with a blurring size *size* is denoted by (A, *size*). We generated these kernels using MATLAB function *fspecial* with various testing parameters. For the color image case, we combined various blurring kernels to generate some within-channel blurs and cross-channel blurs. Specially, we used the method in [43] to generate cross-channel blurs, which contains three steps as follow:

1. Generate 9 kernels:
   {(G, 11, 9), (G, 21, 11), (G, 31, 13), (M, 21, 45), (M, 41, 90), (M, 61, 135), (A, 13), (A, 15), (A, 17)};
2. Randomly assign the above 9 kernels to
   $\{K_{11}, K_{12}, K_{13}; K_{21}, K_{22}, K_{23}; K_{31}, K_{32}, K_{33}\}$;
3. Multiply 0.8 to the diagonal kernels and 0.1 to the off-diagonal kernels.

In the experiments, we used MATLAB function *imnoise* to add Gaussian noise with zero mean and various deviations to the signals. For simplicity, we denote the deviation as *dev*.

For all of the tested algorithms, we used the following stopping condition:

- Given a tolerance $\epsilon > 0$, the iteration runs until $\frac{\|u^k - u^{k-1}\|}{\|u^{k-1}\|} \leq \epsilon$ (or $\frac{\|\mathbf{u}^k - \mathbf{u}^{k-1}\|}{\|\mathbf{u}^{k-1}\|} \leq \epsilon$ in the case of color images).

In the following we present some results of the ALM applied to ROF, vectorial TV, and LLT models, as well as some comparisons between ALM, CGM, Chambolle's dual method, the SB, MATLAB built-in functions, and FTVd. We mention that, in all of the figures, SNR and t denote the signal-noise-ratio and CPU time, respectively.

8.2. Results of the ALM applied to the ROF, vectorial TV, and LLT models. In this subsection, we show some numerical examples about ALM applied to the ROF, vectorial TV, and LLT models. See Figures 2, 3, 4, 5, 6, and 7. Different blur kernels, cross-channel blurs, and Gaussian noises with different deviations were used to test our algorithms. The cross-channel blurs were generated using the method in the last subsection. As is well known, to restore an image from a both blurry and noisy observation is a very hard problem. Our algorithms can generate good restoration results very efficiently for common-sized images (such as the Cameraman and Lena images). The example illustrated in Figure 7 shows the ALM applied to the LLT model to reduce the staircase effect of the ROF model; see the zoomed images in Figure 7.

8.3. Comparisons with CGM, Chambolle's dual method, and the SB. In this subsection, we make comparisons between ALM, CGM, Chambolle's dual method, and the SB. We compare these algorithms in the case of image denoising due to the following reasons. First, Chambolle's dual method and the SB (based on Gauss–Seidel iteration) were both proposed for the image denoising case of the ROF model in [12, 24]. Second, when applied to image deblurring, the CGM requires lots of memory and is quite slow since the matrix in the Newton linearization [16] is dense.

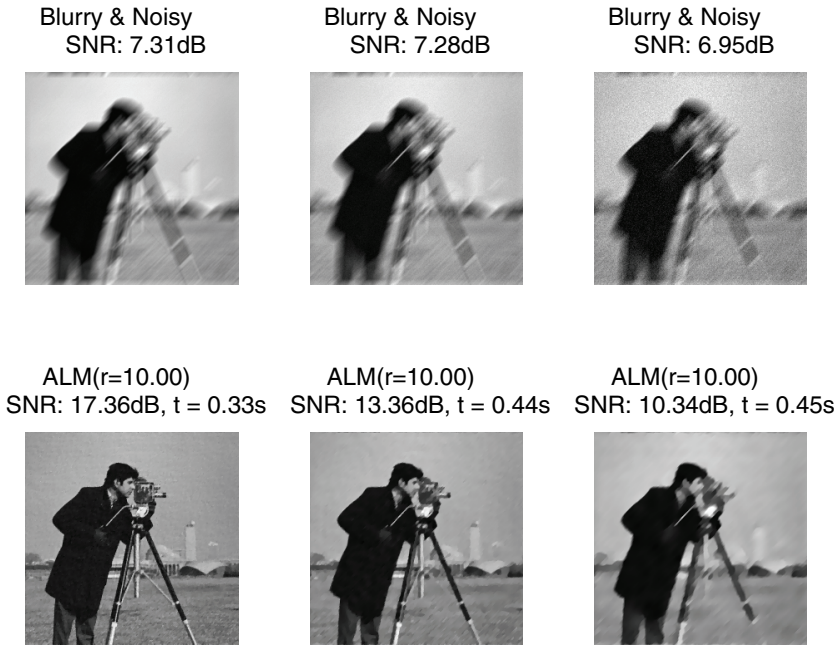

Figure 2. *ALM with parameter $r = 10$ for the ROF restoration. The first row is the Cameraman image corrupted by motion blur $(M, 21, 45)$ and Gaussian noise with dev $= 1.e-5, 1.e-4, 1.e-3$ (from left to right); the second row is the restored images. The stopping tolerance is $\epsilon = 1.e-2$.*

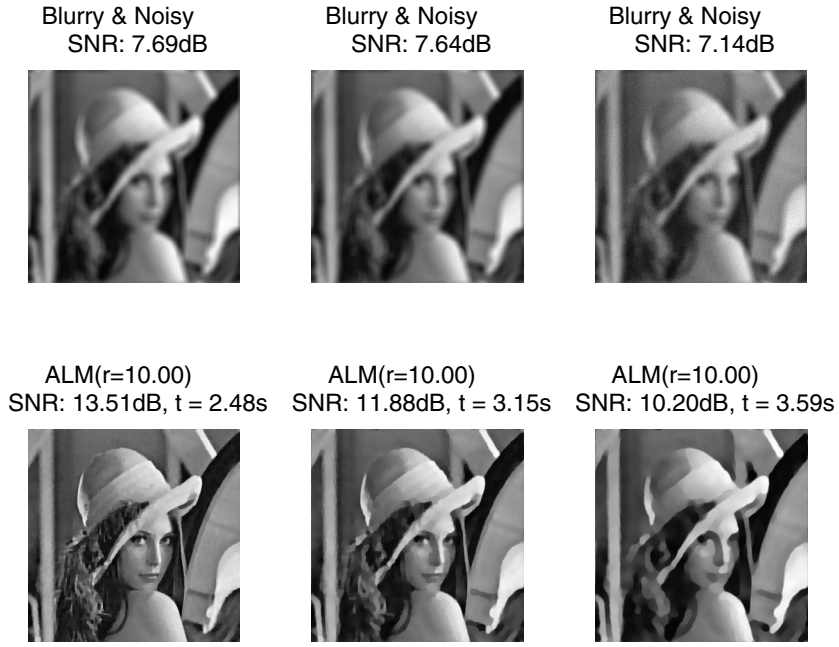

Figure 3. *ALM with parameter $r = 10$ for the ROF restoration. The first row is the Lena image corrupted by Gaussian blur $(G, 21, 11)$ and Gaussian noise with dev $= 1.e-5, 1.e-4, 1.e-3$ (from left to right); the second row is the restored images. The stopping tolerance is $\epsilon = 1.e-2$.*

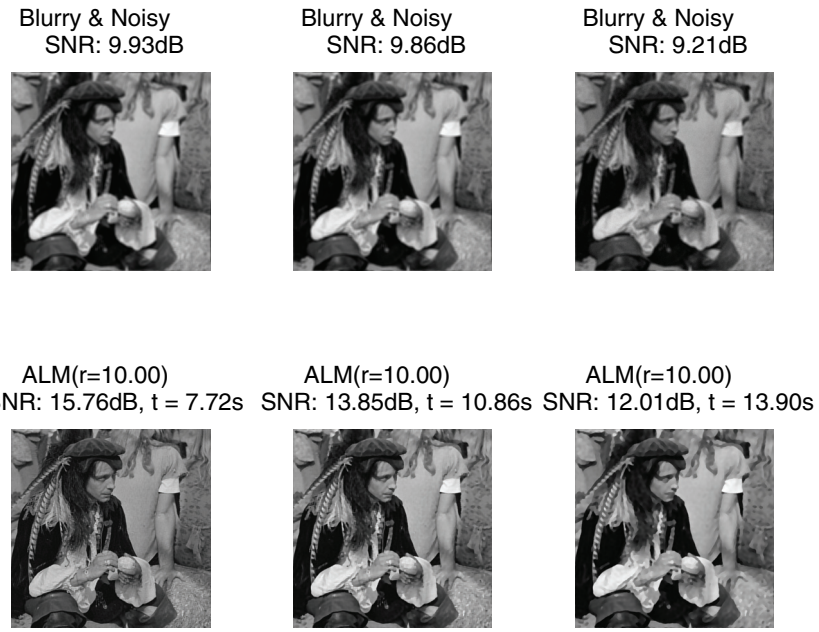

Figure 4. *ALM with parameter $r = 10$ for the ROF restoration. The first row is the Man image corrupted by average blur $(A, 15)$ and Gaussian noise with dev $= 1.e-5, 1.e-4, 1.e-3$ (from left to right); the second row is the restored images. The stopping tolerance is $\epsilon = 1.e-2$.*

Figure 5. *ALM with parameter $r = 10$ for the vectorial TV restoration. The first row is the Lena (color) image corrupted by cross-channel blur (generated in the last subsection) and Gaussian noise with dev $= 1.e-5, 1.e-4, 1.e-3$ (from left to right); the second row is the restored images. The stopping tolerance is $\epsilon = 1.e-2$.*

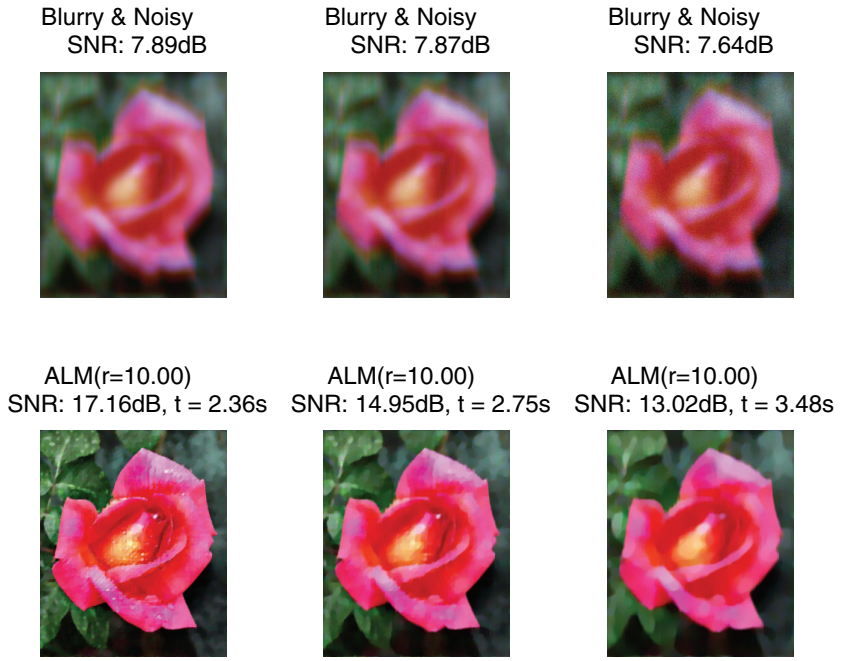

Figure 6. *ALM with parameter $r = 10$ for the vectorial TV restoration. The first row is the Rose image corrupted by cross-channel blur (generated in the last subsection) and Gaussian noise with dev $= 1.e-5, 1.e-4, 1.e-3$ (from left to right); the second row is the restored images. The stopping tolerance is $\epsilon = 1.e-2$.*

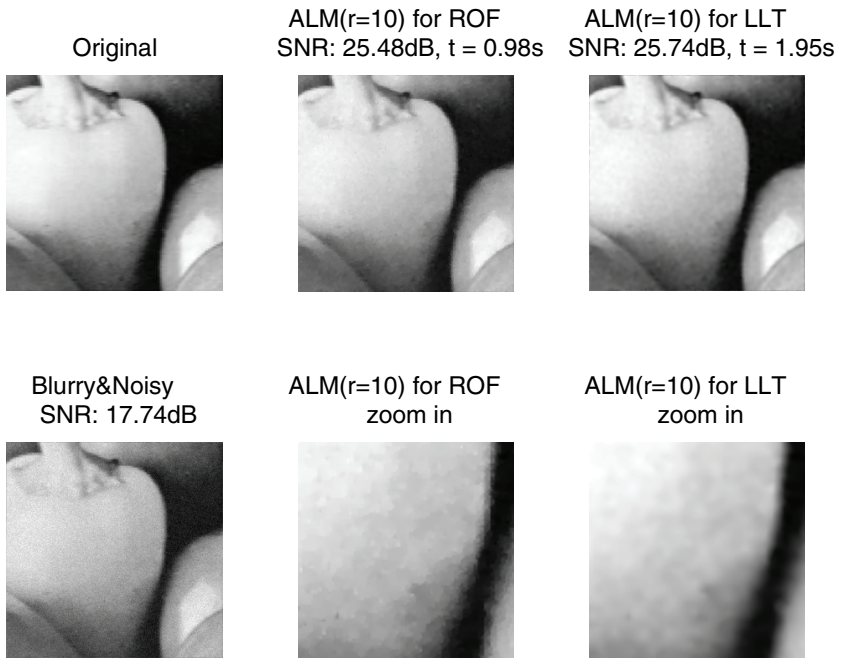

Figure 7. *ALM with parameter $r = 10$ for the ROF and LLT restorations. The blur kernel is $(G, 21, 0.5)$ and the noise level is dev $= 1.e-3$. The stopping tolerance is $\epsilon = 1.e-3$.*

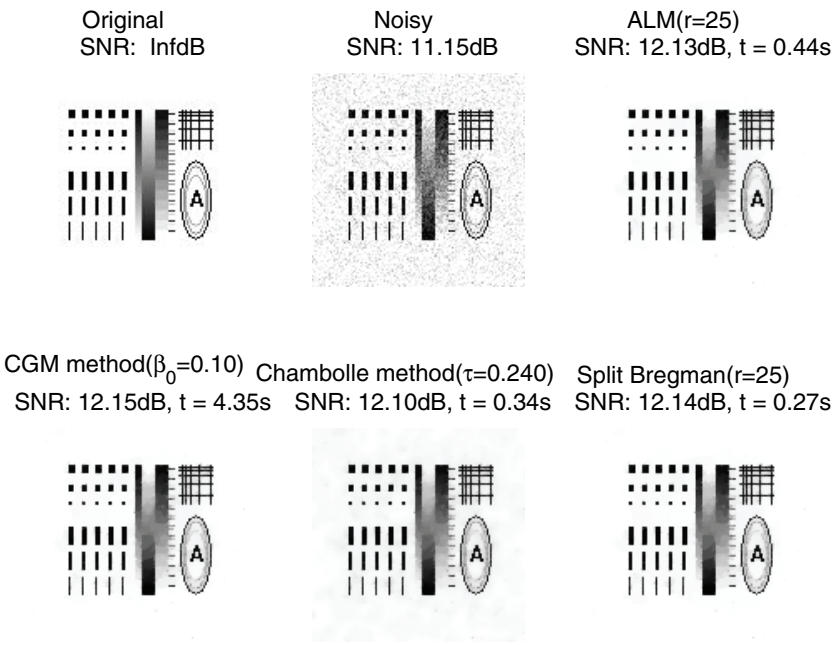

Figure 8. *Comparisons between ALM, CGM, Chambolle's dual method, and the SB for the ROF restoration. The noise level is* $dev = 1.e-2$. *The stopping tolerance is* $\epsilon = 1.e-4$.

The codes for Chambolle's dual method and the CGM for the grayscale image case were obtained from the Internet. Although the SB for grayscale image denoising was implemented in [24] in C++, we did not adopt it due to the consideration of the fairness of comparisons. We implemented the SB in MATLAB for both grayscale and color image cases, where Gauss–Seidel iteration was used to solve the u-subproblems. We also implemented the CGM for the color image case, where a direct solver for the linear system at each iteration was used, and a similar continuation scheme for the smooth parameter β based on a duality gap as in [49] was applied.

Some examples are provided in Figures 8, 9, 10, 11, 12, and 13. We compare these algorithms in both grayscale and color image cases with different-sized images. In the figures, β_0 is the initial value of the smooth parameter β in the CGM, and τ is the (nearly optimal) step size of Chambolle's dual method. (Actually the convergence of Chambolle's method is proved under $\tau < \frac{1}{8}$ but is observed under $\tau < \frac{1}{4}$.) In addition, r is the penalty parameter for the ALM and SB cases. For the grayscale image case, we applied the observation in [24] for setting the parameter r in the ALM. For the color image case, we found that $r = \alpha$ provides good efficiency for ALM and SB. We also recorded the iteration numbers and CPU costs of these algorithms for different stopping tolerances; see Table 1. Considering the length of the paper, we do not provide these data for the color image case, since they have similar statistical features with those for the grayscale image case.

From the figures and the table, we have some conclusions. The SB is in general the fastest method and produces the best restoration results. For large stopping tolerances, Chambolle's dual method becomes the fastest one, but the results are a bit worse. The CGM requires the

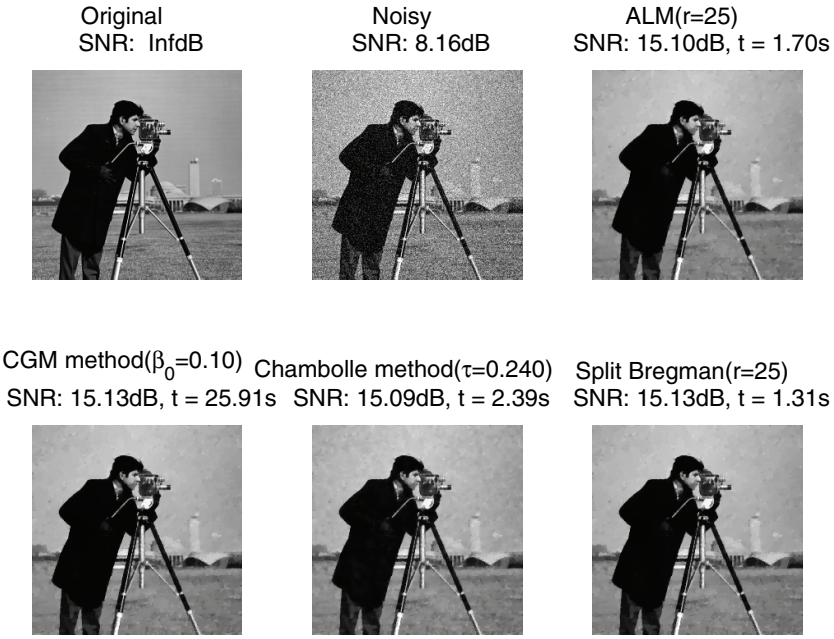

Figure 9. *Comparisons between ALM, CGM, Chambolle's dual method, and SB iteration for the ROF restoration. The noise level is* $dev = 1.e-2$. *The stopping tolerance is* $\epsilon = 1.e-4$.

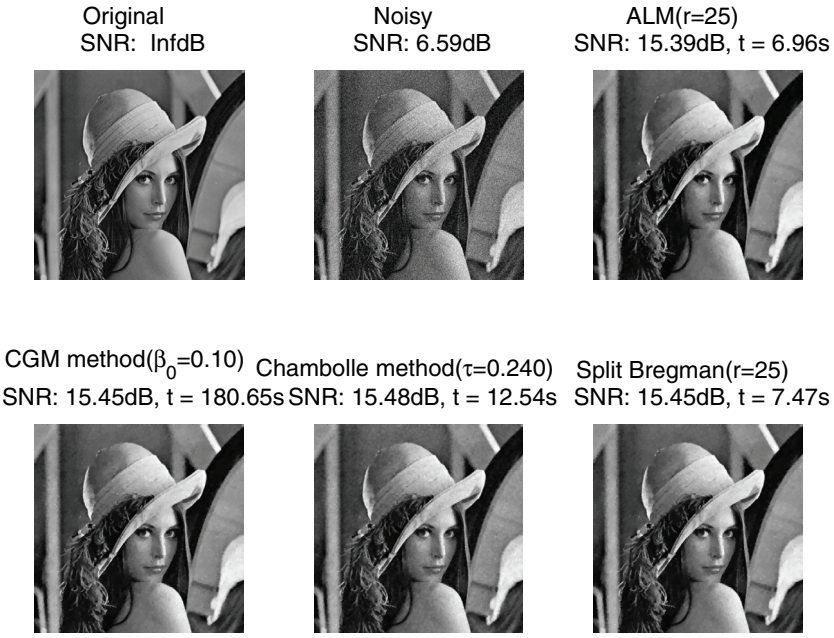

Figure 10. *Comparisons between ALM, CGM, Chambolle's dual method, and the SB for the ROF restoration. The noise level is* $dev = 1.e-2$. *The stopping tolerance is* $\epsilon = 1.e-4$.

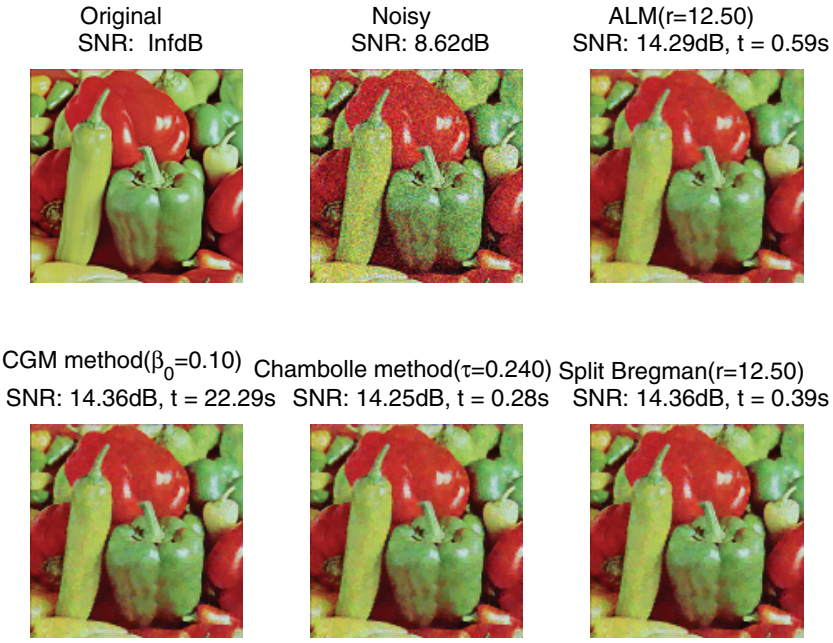

Figure 11. *Comparisons between ALM, CGM, Chambolle's dual method, and the SB for the vectorial TV restoration. The noise level is* $dev = 1.e-2$. *The stopping tolerance is* $\epsilon = 1.e-4$.

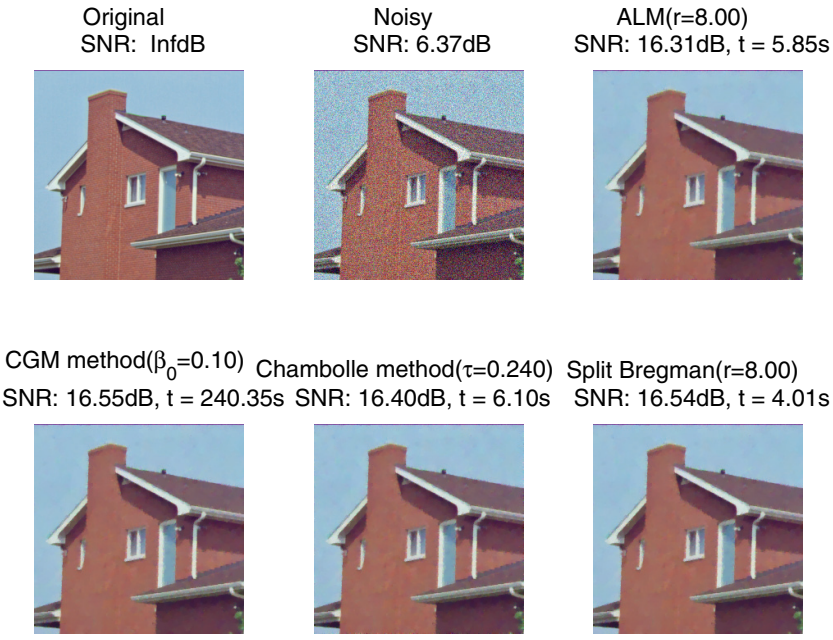

Figure 12. *Comparisons between ALM, CGM, Chambolle's dual method, and the SB for the vectorial TV restoration. The noise level is* $dev = 1.e-2$. *The stopping tolerance is* $\epsilon = 1.e-4$.

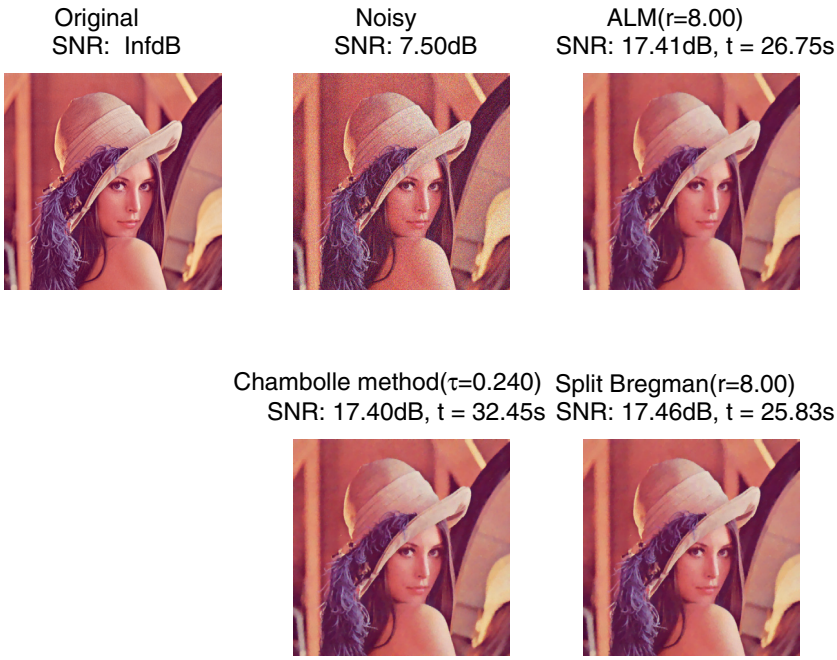

Figure 13. *Comparisons between ALM, CGM, Chambolle's dual method, and the SB for the vectorial TV restoration. The noise level is* $dev = 1.e-2$. *The stopping tolerance is* $\epsilon = 1.e-4$. *The CGM method ran out of memory for this example.*

fewest numbers of iterations but at more CPU cost. The SB is faster than ALM in general for two main reasons: First, SB iteration solves the u-subproblem in the physical domain, but ALM solves it in the frequency domain. Since the u-subproblem for image denoising problems has sparse structure in the physical domain but dense structure in the frequency domain, fewer calculations are needed at each iteration in the SB method. Second, it is not necessary to solve the subproblems exactly in the SB (also in ALM), and one single Gauss–Seidel iteration is enough. However, as image size gets larger, the CPU time of the ALM becomes nearly the same as (sometimes less than) that of the SB. Also, as mentioned before, our FFT-based implementation can handle more general cases (for general blurs K).

8.4. Comparisons with MATLAB functions. Comparisons between ALM and some built-in MATLAB functions, i.e., *deconvwnr.m, deconvreg.m,* and *deconvlucy.m*, are shown in Figures 14, 15, 16, and 17. Since these MATLAB functions were designed for image deblurring, we added quite low level Gaussian noise in the experiments in this subsection, i.e., with $dev = 1.e-6$. (The restoration quality of the MATLAB functions is very bad when the noise level is higher than $dev = 1.e{-}6$.) We also mention that *deconvwnr.m, deconvreg.m,* and *deconvlucy.m* are the only MATLAB built-in functions that can handle color image deblurring, and they require the diagonal blur kernels to be identical, i.e., $K_{11} = K_{22} = K_{33}$, and all the off-diagonal blur kernels to be 0. The kernels listed in the captions of Figures 16 and 17 are the diagonal elements in these two examples, respectively. As one can see, the ALM generates much better restorations than these built-in MATLAB functions at comparable (or even less)

Table 1

Numbers of iterations and CPU costs (in seconds) of ALM, CGM, Chambolle's method, and SB, for the problems shown in Figures 8, 9, and 10. Tol is the stopping tolerance.

(a) Denoising of the Shape image (Figure 8).

	Tol ϵ =1.e-3		Tol ϵ =1.e-4		Tol ϵ =1.e-5		Tol ϵ =1.e-6	
Algorithms	Iter	Time	Iter	Time	Iter	Time	Iter	Time
ALM	27	0.25	50	0.44	119	1.05	353	2.98
CGM	8	2.46	14	4.35	19	6.71	23	10.11
Chambolle	5	0.08	20	0.34	97	1.54	1324	20.48
SB	29	0.16	51	0.27	113	0.59	372	1.90

(b) Denoising of the Cameraman image (Figure 9).

	Tol ϵ =1.e-3		Tol ϵ =1.e-4		Tol ϵ =1.e-5		Tol ϵ =1.e-6	
Algorithms	Iter	Time	Iter	Time	Iter	Time	Iter	Time
ALM	17	0.84	36	1.70	119	5.44	344	15.63
CGM	10	17.25	15	25.91	19	48.92	22	106.88
Chambolle	6	0.66	24	2.39	118	11.47	633	61.48
SB	20	0.70	39	1.31	119	3.88	350	11.65

(c) Denoising of the Lena image (Figure 10). The CGM method ran out of memory for ϵ =1.e-5, 1.e-6.

	Tol ϵ =1.e-3		Tol ϵ =1.e-4		Tol ϵ =1.e-5		Tol ϵ =1.e-6	
Algorithms	Iter	Time	Iter	Time	Iter	Time	Iter	Time
ALM	16	3.70	32	6.96	84	18.17	226	47.77
CGM	11	144.49	15	180.65	-	-	-	-
Chambolle	5	2.98	22	12.54	85	48.53	331	187.50
SB	19	4.04	36	7.47	82	17.36	236	49.59

CPU time costs.

8.5. Comparisons with FTVd. In this subsection we provide several examples to compare the ALM and FTVd [40, 43]. See Figure 18. FTVd is a recently proposed method for solving TV based problems. It is much more efficient than other existing methods, as discussed in [40]. As shown in Figure 18, the ALM is a little bit faster than FTVd and generates a bit better restorations.

8.6. A remark on the computational cost of the ALM. We would like to remark here on the computational costs of the ALM applied to the ROF, vectorial TV, and high order models, respectively. As one can see, our method contains two iterations, one inner iteration and one outer iteration. Since the update of Lagrange multipliers is extremely fast, the cost of the algorithm at each outer iteration is dominated by the inner iteration (see Algorithms 4.1, 6.1, and 7.1). For simplicity of description, in the following we assume N^2 as the number of total pixels.

In Algorithm 4.2 for the ROF model, there are mainly two operations: the FFT-based solver (4.11) for the v-subproblem and the shrinkage operation (4.12) for the q-subproblem. The shrinkage operation (4.12) has a linear computational complexity, $O(N^2)$, and each FFT has a complexity of $O(N^2 \log N)$. The Fourier transforms of the operators $K, \mathring{D}_x^-, \mathring{D}_y^-, \triangle$

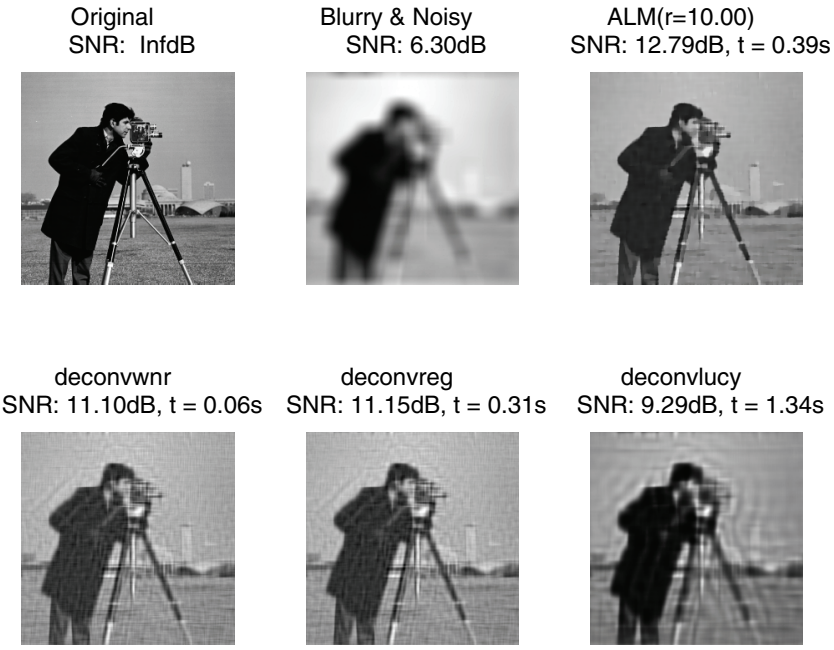

Figure 14. *ALM for the ROF restoration, and comparisons to* MATLAB *built-in functions. The blur kernel is Gaussian blur* $(G, 21, 11)$. *The noise level is dev* $= 1.e-6$. *The tolerance is* $1.e-2$.

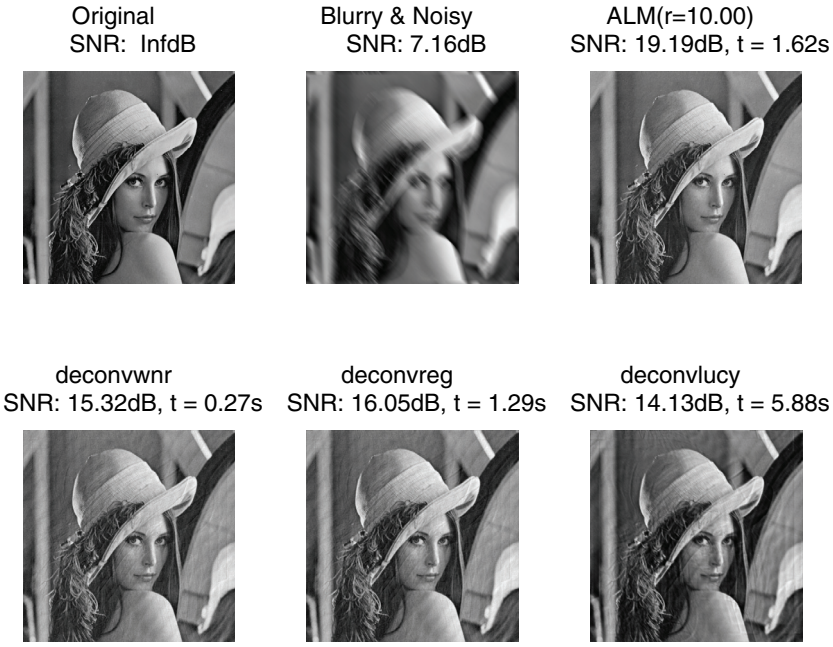

Figure 15. *ALM for the ROF restoration, and comparisons to* MATLAB *built-in functions. The blur kernel is motion blur* $(M, 30, 135)$. *The noise level is dev* $= 1.e-6$. *The tolerance is* $1.e-2$.

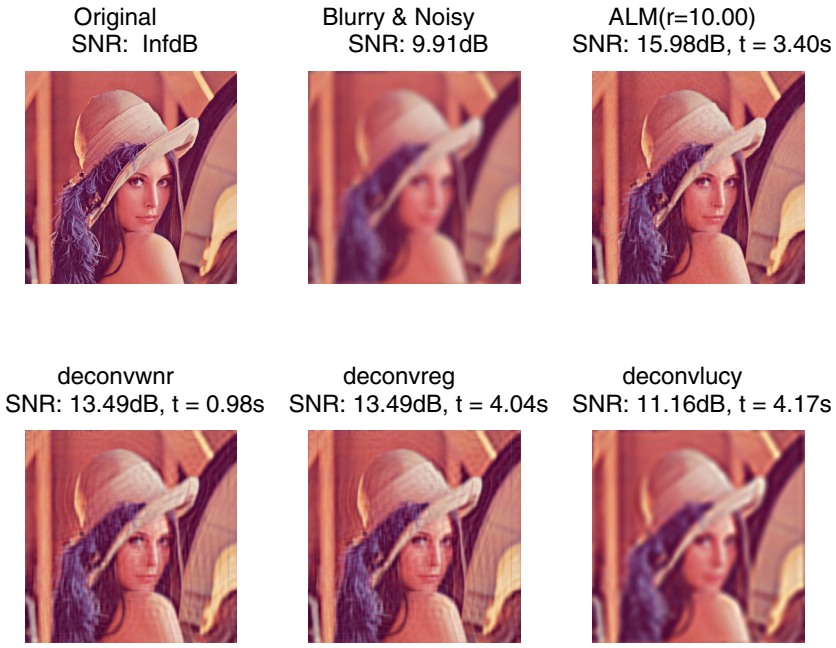

Figure 16. *ALM for the vectorial TV restoration, and comparisons to* MATLAB *built-in functions. The blur kernel is Gaussian blur* $(G, 21, 11)$*. The noise level is dev* $= 1.e-6$*. The tolerance is* $1.e-1$*.*

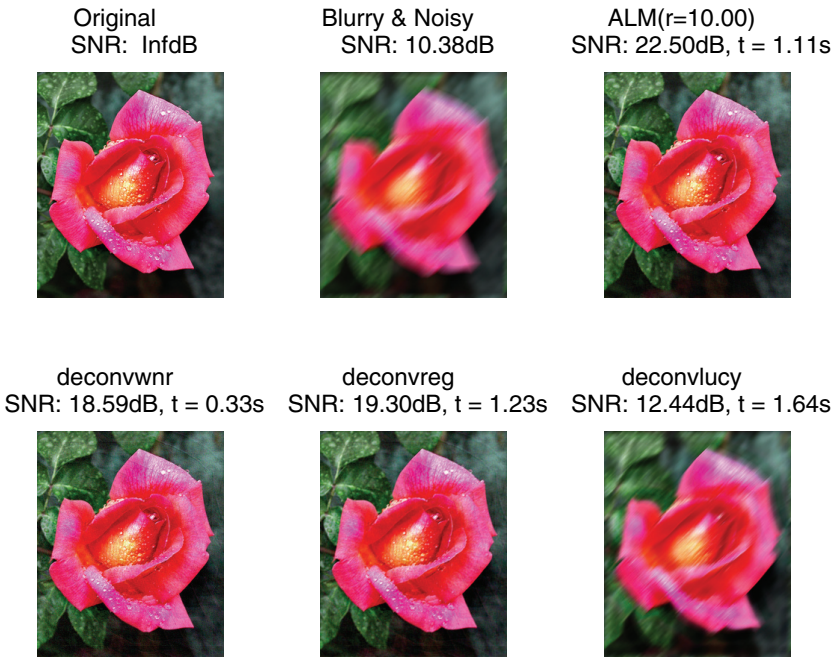

Figure 17. *ALM for the vectorial TV restoration, and comparisons to* MATLAB *built-in functions. The blur kernel is motion blur* $(M, 21, 45)$*. The noise level is dev* $= 1.e-6$*. The tolerance is* $1.e-1$*.*

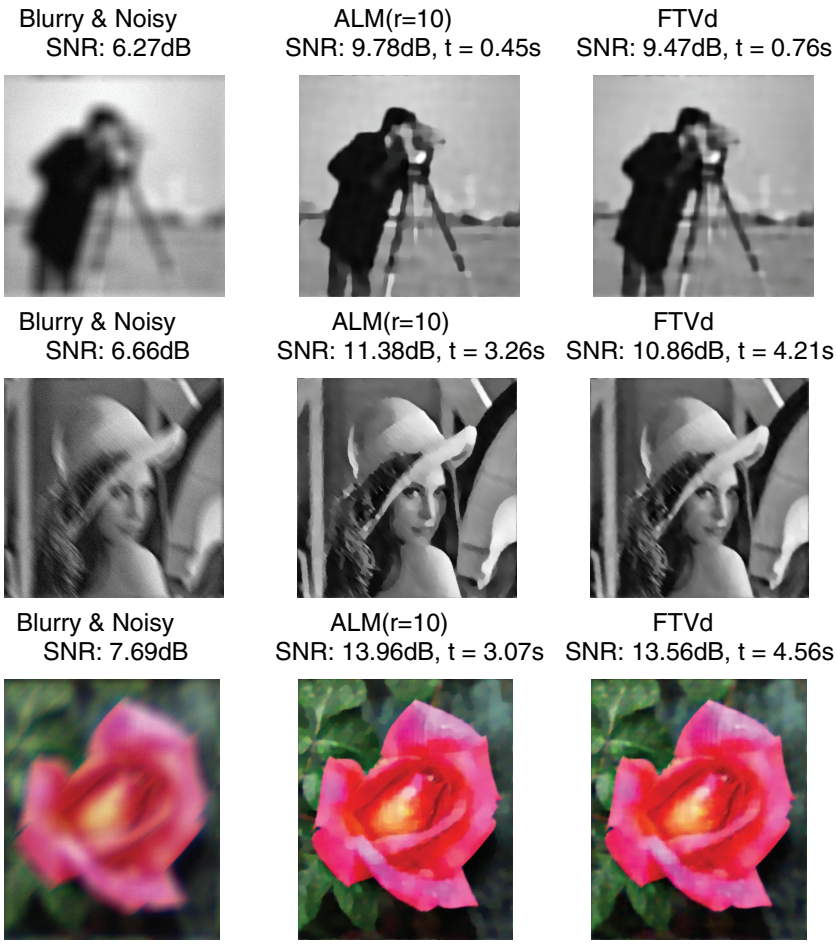

Figure 18. *Comparisons between ALM and FTVd for the ROF (the first two rows) and vectorial TV (the third row) restorations. The first row is with Gaussian blur* $(G, 21, 11)$ *and Gaussian noise with dev* $= 1.e-4$. *The second row is with motion blur* $(M, 30, 135)$ *and Gaussian noise with dev* $= 1.e-3$. *The third row is with cross-channel blur (generated in section 8.1) and Gaussian noise with dev* $= 1.e-4$. *The tolerance is* $1.e-2$.

and the observed image f are constants, and thus are calculated only once during the whole algorithm. At each iteration, Algorithm 4.2 performs 2 FFTs, 1 inverse FFT, and 1 shrinkage. Since in this paper we set $L = 1$ in Algorithm 4.2, the total cost per outer iteration is just one round of inner iteration, i.e., 2 FFTs, 1 inverse FFT, and 1 shrinkage.

In Algorithm 6.2 for the vectorial TV model, there are also mainly two operations: the FFT-based solver (6.11) and the shrinkage operation (6.12). Different from the ROF model, here we need to solve an algebraic linear system in (6.11). We apply Gaussian elimination to solve the system, whose cost is very small compared to FFTs [43]. Therefore, at each iteration the cost of (6.11) is dominated by that of the FFTs. By a similar analysis as for Algorithm 4.2, Algorithm 6.2 performs $2M$ FFTs, M inverse FFTs, and 1 shrinkage at each iteration. Due to $L = 1$ in this paper, the total cost per outer iteration is thus about $2M$ FFTs, M inverse FFTs, and 1 shrinkage. In the color image case where $M = 3$, the cost per outer

iteration is 6 FFTs, 3 inverse FFTs, and 1 shrinkage.

By similar analysis as in the former two cases, we have, for Algorithm 7.2, that the per iteration cost is dominated by $4M$ FFTs, M inverse FFTs, and 1 shrinkage. Also, the total cost per outer iteration is about $4M$ FFTs, M inverse FFTs, and 1 shrinkage, which is 12 FFTs, 3 inverse FFTs, and 1 shrinkage in the case of color images.

The remaining question is how many outer iterations our method (i.e., Algorithms 4.1, 6.1, and 7.1) needs to meet a prescribed accuracy, i.e., the tolerance ϵ. The larger ϵ is, the fewer iterations are enough. The smaller ϵ is, the more iterations are needed. It is hard to give a theoretical result on this issue. According to our numerical experiments, to get a visually good restoration, ϵ =1.e-1, 1.e-2 is in general enough for problems with low level noises (such as $dev = 1.e-6, 1.e-5, 1.e-4$), and ϵ =1.e-3, 1.e-4 is enough for problems with high level noises (such as $dev = 1.e-2$); see the given figures. Therefore, the higher the noise level is, the more iterations are needed. Our method needs in general about 10 outer iterations to obtain visually quite good restorations. For image deblurring problems (where the noise level is low), the number of outer iterations is even less, e.g., 5 or 6. For image denoising problems (where the noise level is high), however, more iterations are needed; see Table 1.

9. Conclusions and future works. In this paper, we have presented the augmented Lagrangian method for solving the ROF, vectorial TV, and high order models, in the discrete setting, which is very clear for practical implementation in digital image processing. As demonstrated by the examples, the proposed algorithms benefit from both accuracy and efficiency. Some convergence analysis for our approach was also provided. In addition, we described, in the discrete setting, close connections between the augmented Lagrangian method and several other efficient approaches, such as CGM and Chambolle's dual method. Using the extensions to vectorial TV and high order models, we demonstrated how to obtain some new methods for vectorial TV and high order models, e.g., CGM and the split Bregman iteration applied to these models. A possible future avenue is to further extend the method to models with other data fidelity terms, e.g., the TV-L_1 model. We noticed that recently a variant of the ROF model was proposed [28] which avoids the staircase effect of the original ROF model. To apply our approach to this variant is also a valuable subject for future research.

Acknowledgments. We thank Dr. M. Zhu, Dr. X. Bresson, Dr. T. Goldstein, and Prof. W. Yin for their making their codes available online.

REFERENCES

[1] D.P. Bertsekas, *Multiplier methods: A survey*, Automatica. J. IFAC, 12 (1976), pp. 133–145.
[2] D.P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, Belmont, MA, 1996.
[3] D.P. Bertsekas and J.N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, Athena Scientific, Belmont, MA, 1997.
[4] P. Blomgren and T.F. Chan, *Color TV: Total variation methods for restoration of vector-valued images*, IEEE Trans. Image Process., 7 (1998), pp. 304–309.
[5] L.M. Bregman, *The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming*, USSR Comput. Math. Math. Phys., 7 (1967), pp. 200–217.

[6] X. Bresson and T.F. Chan, *Fast dual minimization of the vectorial total variation norm and applications to color image processing*, Inverse Probl. Imaging, 2 (2008), pp. 455–484.
[7] A. Buades, B. Coll, and J.M. Morel, *The staircasing effect in neighborhood filters and its solution*, IEEE Trans. Image Process., 15 (2006), pp. 1499–1505.
[8] A. Caboussat, R. Glowinski, and V. Pons, *An augmented Lagrangian approach to the numerical solution of a non-smooth eigenvalue problem*, J. Numer. Math., 17 (2009), pp. 3–26.
[9] J. Cai, S. Osher, and Z. Shen, *Linear Bregman iterations for compressed sensing*, Math. Comp., 78 (2009), pp. 1515–1536.
[10] J.-F. Cai, S. Osher, and Z. Shen, *Split Bregman methods and frame based image restoration*, Multiscale Model. Simul, 8 (2009), pp. 337–369.
[11] J.L. Carter, *Dual Methods for Total Variation-Based Image Restoration*, Ph.D. thesis, Department of Mathematics, UCLA, Los Angeles, CA, 2001.
[12] A. Chambolle, *An algorithm for total variation minimization and applications*, J. Math. Imaging Vis., 20 (2004), pp. 89–97.
[13] A. Chambolle and P.L. Lions, *Image recovery via total variation minimization and related problems*, Numer. Math., 76 (1997), pp. 167–188.
[14] T.F. Chan, S. Esedoglu, and F.E. Park, *A Fourth Order Dual Method for Staircase Reduction in Texture Extraction and Image Restoration Problems*, UCLA CAM Report cam05-28, Department of Mathematics, UCLA, Los Angeles, CA, 2005.
[15] T. Chan, S. Esedoglu, F. Park, and A. Yip, *Recent developments in total variation image restoration*, in Mathematical Models of Computer Vision, N. Paragios, Y. Chen, and O. Faugeras, eds., Springer-Verlag, Berlin, 2005.
[16] 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), pp. 1964–1977.
[17] T.F. Chan, S.H. Kang, and J.H. Shen, *Total variation denoising and enhancement of color images based on the CB and HSV color models*, J. Visual Commun. Image Rep., 12 (2001), pp. 422–435.
[18] T. Chan, A. Marquina, and P. Mulet, *High-order total variation-based image restoration*, SIAM J. Sci. Comput., 22 (2000), pp. 503–516.
[19] D.L. Donoho, *De-noising by soft-thresholding*, IEEE Trans. Inform. Theory, 41 (1995), pp. 613–627.
[20] J. Eckstein and D.P. Bertsekas, *On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Programming, 55 (1992), pp. 293–318.
[21] I. Ekeland and R. Témam, *Convex Analysis and Variational Problems*, Classics in Appl. Math. 28, SIAM, Philadelphia, 1999.
[22] E. Esser, *Applications of Lagrangian-Based Alternating Direction Methods and Connections to Split Bregman*, UCLA CAM Report 09–31, Department of Mathematics, UCLA, Los Angeles, CA, 2009.
[23] R. Glowinski and P. Le Tallec, *Augmented Lagrangians and Operator-Splitting Methods in Nonlinear Mechanics*, SIAM Stud. Appl. Math. 9, SIAM, Philadelphia, 1989.
[24] T. Goldstein and S. Osher, *The split Bregman method for L1-regularized problems*, SIAM J. Imaging Sci., 2 (2009), pp. 323–343.
[25] M.R. Hestenes, *Multiplier and gradient methods*, J. Optim. Theory Appl., 4 (1969), pp. 303–320.
[26] W. Hinterberger and O. Scherzer, *Variational methods on the space of functions of bounded Hessian for convexification and denoising*, Computing, 76 (2006), pp. 109–133.
[27] Y. Huang, M.K. Ng, and Y.W. Wen, *A fast total variation minimization method for image restoration*, Multiscale Model. Simul., 7 (2008), pp. 774–795.
[28] K. Jalalzai and A. Chambolle, *Enhancement of blurred and noisy images based on an original variant of the total variation*, Second International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 2009), Voss, Norway, 2009, Lecture Notes in Comput. Sci. 5567, Springer-Verlag, Berlin, 2009, pp. 368–376.
[29] M. Lysaker, A. Lundervold, and X.-C. Tai, *Noise removal using fourth-order partial differential equations with applications to medical magnetic resonance images in space and time*, IEEE Trans. Image Process., 12 (2003), pp. 1579–1590.
[30] M. Lysaker and X.-C. Tai, *Iterative image restoration combining total variation minimization and a second-order functional*, Int. J. Computer Vision, 66 (2006), pp. 5–18.
[31] S. Osher, M. Burger, D. Goldfarb, J.J. Xu, and W.T. Yin, *An iterative regularization method for total variation-based image restoration*, Multiscale Model. Simul., 4 (2005), pp. 460–489.

[32] M.J.D. Powell, *A method for nonlinear constraints in minimization problems*, in Optimization, R. Fletcher, ed., Academic Press, New York, 1972, pp. 283–298.
[33] R.T. Rockafellar, *A dual approach to solving nonlinear programming problems by unconstrained optimization*, Math. Programming, 5 (1973), pp. 354–373.
[34] L. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Phys. D, 60 (1992), pp. 259–268.
[35] G. Sapiro and D.L. Ringach, *Anisotropic diffusion of multivalued images with applications to color filtering*, IEEE Trans. Image Process., 5 (1996), pp. 1582–1586.
[36] O. Scherer, *Denoising with higher order derivatives of bounded variation and an application to parameter estimation*, Comput., 60 (1998), pp. 1–27.
[37] S. Setzer, *Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage*, in Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 2009), Voss, Norway, 2009, Lecture Notes in Comput. Sci. 5567, Springer Verlag, Berlin, 2009, pp. 464-476.
[38] G. Steidl, *A note on the dual treatment of higher-order regularization functionals*, Comput., 76 (2006), pp. 135–148.
[39] X.-C. Tai and C. Wu, *Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model*, in Proceedings of the Second International Conference on Scale Space and Variational Methods in Computer Vision (SSVM 2009), Voss, Norway, 2009, Lecture Notes in Comput. Sci. 5567, Springer-Verlag, Berlin, 2009, pp. 502–513.
[40] Y. Wang, J. Yang, W. Yin, and Y. Zhang, *A new alternating minimization algorithm for total variation image reconstruction*, SIAM J. Imaging Sci., 1 (2008), pp. 248–272.
[41] Y.L. Wang, W.T. Yin, and Y. Zhang, *A Fast Algorithm for Image Deblurring with Total Variation Regularization*, UCLA CAM Report 07-22, Department of Mathematics, UCLA, Los Angeles, CA, 2007.
[42] J. Weickert, *Anisotropic Diffusion in Image Processing*, B.G. Teubner, Stuttgart, 1998.
[43] J. Yang, W. Yin, Y. Zhang, and Y. Wang, *A fast algorithm for edge-preserving variational multichannel image restoration*, SIAM J. Imaging Sci., 2 (2009), pp. 569–592.
[44] W.T. Yin, *Analysis and Generalizations of the Linearized Bregman Method*, UCLA CAM Report 09-42, Department of Mathematics, UCLA, Los Angeles, CA, 2009.
[45] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, *Bregman iterative algorithms for l_1-minimization with applications to compressed sensing*, SIAM J. Imaging Sci., 1 (2008), pp. 143–168.
[46] Y.-L. You and M. Kaveh, *Fourth-order partial differential equation for noise removal*, IEEE Trans. Image Process., 9 (2000), pp. 1723–1730.
[47] Y.-L. You, W. Xu, A. Tannenbaum, and M. Kaveh, *Behavioral analysis of anisotropic diffusion in image processing*, IEEE Trans. Image Process., 5 (1996), pp. 1539–1553.
[48] X. Zhang, M. Burger, X. Bresson, and S. Osher, *Bregmanized nonlocal regularization for deconvolution and sparse reconstruction*, SIAM J. Imaging Sci., 3 (2010), pp. 253–276.
[49] M. Zhu, S.J. Wright, and T.F. Chan, *Duality-based algorithms for total-variation-regularized image restoration*, Comput. Optim. Appl., 2008; DOI: 10.1007/s10589-008-9225-2.