A fast edge detection algorithm using binary labels

Shi, Yuying; Gu, Ying; Wang, Li-Lian; Tai, Xue-Cheng

2015

Shi, Y., Gu, Y., Wang, L.-L., & Tai, X.-C. (2015). A fast edge detection algorithm using binary labels. Inverse problems and imaging, 9(2), 551-578.

https://hdl.handle.net/10356/107189

https://doi.org/10.3934/ipi.2015.9.551

Inverse Problems and Imaging
Volume 9, No. 2, 2015, 551–578

doi:10.3934/ipi.2015.9.551

A FAST EDGE DETECTION ALGORITHM USING BINARY LABELS

Yuying Shi

Department of Mathematics and Physics, North China Electric Power University
Beijing, 102206, China

Ying Gu

Institute for Infocomm Research, Agency for Science
Technology and Research (A*STAR), 138632, Singapore

Li-Lian Wang

Division of Mathematical Sciences, School of Physical and Mathematical Sciences
Nanyang Technological University, 637371, Singapore.

Xue-Cheng Tai

Department of Mathematics, University of Bergen
P. O. Box 7800, N-5020, Bergen, Norway

(Communicated by Yunmei Chen)

Abstract. Edge detection (for both open and closed edges) from real images is a challenging problem. Developing fast algorithms with good accuracy and stability for noisy images is difficult yet and in demand. In this work, we present a variational model which is related to the well-known Mumford-Shah functional and design fast numerical methods to solve this new model through a binary labeling processing. A pre-smoothing step is implemented for the model, which enhances the accuracy of detection. Ample numerical experiments on grey-scale as well as color images are provided. The efficiency and accuracy of the model and the proposed minimization algorithms are demonstrated through comparing it with some existing methodologies.

1. **Introduction.** Edge detection plays an immensely important role not only in image processing and computer vision, but also in many other fields, e.g., material science and physics [67, 6]. In the context of image processing, edge detection is to extract the boundaries of some objects of interest from a given image. Many methods have been proposed for this purpose. The capability of classical approaches using image gradient [15, 23, 59] is limited, as the accuracy of edge identification is usually deteriorated by the presence of noise. Another class of methods based on anisotropic diffusions [17, 1, 48] is typically built upon some model that can prevent smoothing near the edges, but encourage diffusion over the homogeneous

2010 *Mathematics Subject Classification.* Primary: 65D15, 68W40, 90C90; Secondary: 65M32.

Key words and phrases. Mumford-Shah model, open/closed curves, offset curves, binary level set method, edge detection.

The first author is supported by NSFC grant(No. 11271126), the Fundamental Research Funds for the Central Universities (No. 2014ZZD10) and Program for Beijing Excellent Talents. The third author is partially supported by the Singapore MOE AcRF Tier 1 Grant (RG 15/12), MOE AcRF Tier 2 Grant (MOE 2013-T2-1-095, ARC 44/13), and Singapore A*STAR-SERC-PSF Grant (122-PSF-007).

regions. However, they may face difficulties in dealing with images with low signal to noise ratios (SNRs). Note that some reaction and diffusion equations [61, 70], edge flow [35], snake/active contour models [30, 16, 4], universal gravity [60] and wavelet frames [28] have been developed for edge detection. We also remark that recent research interest in using a variety of filter banks to improve the accuracy of edge detection, and the interested readers are referred to [10, 37, 34, 49, 64] and the references therein.

In distinctive contrast with region-based image segmentation, edge detection also targets at locating open curves, which are constituent component of edges, but do not have interior and exterior regional separation. Although an open-ended curve can be expressed by a parametric form, it is not a natural representation for evolution. Among a few techniques for manipulating open curves, Leung and Zhao [33] proposed a grid-based particle method to represent an open curve (also see [58, 5, 40, 53] for different ideas). It is known that the level-set method [47] has emerged as a versatile tool for various applications involving curve/surface evolution [31].

In a recent conference report [68], we proposed to embed an open (or a closed) curve into a narrow region (or band), formed by the curve and its parallel curve (also known as the offset curve [66]). We then integrated the MS model with the binary level-set method, leading to a model dubbed as the modified Mumford-Shah (MMS) model. However, only a very preliminary algorithm was proposed for this modified model [68]. This paper aims to present a new MMS model (incorporated with minimizing the area of the narrow band), and introduce two fast algorithms using the proximity algorithm [41, 42] and split Bregman method [27]. We also compare the performance of the algorithms with the method by Ambrosio and Tortorelli (AT) [2, 3], and the adaptive splitting (EDAS-1) algorithm in Llanas [34] for edge detection. We demonstrate that the proposed methods outperform the AT's approach and the EDAS-1 algorithm in terms of ease of implementation, quality of detection, and robustness to noise.

Huang et al. [29] showed that the alternating direction method (ADM) with exactly solving inner subproblems enjoys a linear convergence in the context of variational image restoration. We present a similar convergence result (see Theorem 3.6) for our proposed algorithm under the assumption that the underlying inner subproblem has exact solution. As with [29], such a convergence rate can be observed even with a few inner loops.

The rest of the paper is organized as follows. In Section 2, we introduce the new MMS model [68]. In Section 3, we describe the algorithms for solving the minimization problem, including the proximity algorithm and the split Bregman algorithm. In Section 4, we make a comparison study and provide sufficient numerical results to demonstrate the strengths of the proposed methods. In Section 5, we introduce detailed coupled pretreatments. In Section 6, we extend the proposed model and algorithms for edge detection involving color images.

2. The modified Mumford-Shah model.

We start with the well-known Mumford-Shah model [43]:

$$\min_{u,\Gamma}\Big\{E(u,\Gamma)=\mu\int_{\Omega\setminus\Gamma}|\nabla u|^2 d\boldsymbol{x}+\frac{\nu}{2}\int_{\Omega}(u-I)^2 d\boldsymbol{x}+|\Gamma|\Big\}, \tag{1}$$

where I is a given image on an open bounded domain $\Omega\subset\mathbb{R}^2$, the minimizer u is expected to be a "good" piecewise smooth approximation of I, $|\Gamma|$ is the total length of the targeted union of the edges Γ, and μ, ν are positive tuning parameters. Indeed, the MS model with level-set implementation becomes a very useful tool for region-based image segmentation. In the literature, there are some papers on the variant of Mumford-Shah model for image segmentation and other applications (see e.g., [18, 50] and the references therein).

When it comes to edge detection using a variational model, the most important issue appears how to characterize the edges. To fix the idea, let $\boldsymbol{r} \in \Gamma$ be a curve with the parameterization: $\boldsymbol{r}(t) = (x(t), y(t)), t \in [0,1]$. Without loss of generality, assume that $\boldsymbol{r}(t)$ is regular (i.e., $|\boldsymbol{r}'(t)| \neq 0$ for all $t \in (0,1)$), which has finite length and curvature. Then the parallel or offset curve generated by $\boldsymbol{r}(t)$ is defined by (cf. [24, 66]):

$$\boldsymbol{r}_d(t) = \boldsymbol{r}(t) + d\,\boldsymbol{n}(t), \quad \forall t \in [0,1], \tag{2}$$

where $\boldsymbol{n}(t)$ is the unit normal to $\boldsymbol{r}(t)$ at each point, and d is a preassigned signed distance. This defines a positive (exterior, $d > 0$) or negative (interior, $d < 0$) offset (see Figure 1). Denoting the total length of $\boldsymbol{r}$ and $\boldsymbol{r}_d$ by L and L_d, respectively, we have (cf. [24, Lemma 3.1]):

$$L_d = \int_0^1 |1 + \kappa d|\,|\boldsymbol{r}'|dt = \int_0^L |1 + \kappa d|\,ds = |L + d\,\Delta\theta|,$$

where κ is the curvature and $\Delta\theta$ is the total angle of rotation of the normal $\boldsymbol{n}$ to $\boldsymbol{r}$ between $t = 0$ and $t = 1$, measured by the right-handed rule. Moreover, the area, denoted by A_d, between the generator $\boldsymbol{r}$ and the offset $\boldsymbol{r}_d$ is given by (see [24] again):

$$A_d = \frac{1}{2}(L + L_d)|d|. \tag{3}$$

Note that the curve $\boldsymbol{r}$ is immersed in the (closed) band (with area A_d) denoted by R_d.

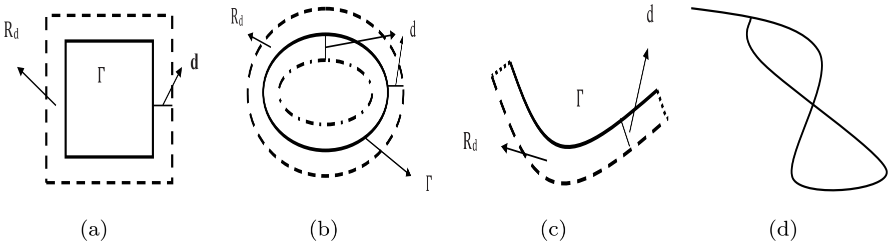


Figure 1. The closed narrow region R_d formed by the curve $\Gamma : \boldsymbol{r}(t)$ and its exterior parallel curve (dashed line) for closed curves and open curves (the dotted lines connected the corresponding starting points and end-points of the curve Γ and its exterior parallel curve, respectively), where the dot-dashed line is the interior parallel curve. The last one is an example of an intersected curve that can be split into simple open or closed curves.

We introduce the binary level-set function: $\psi = 1$, if $\boldsymbol{x} \in R_d$, while $\psi = 0$ otherwise. Recall that the total variation of the indicator function of R_d characterizes the perimeter of R_d, so we deduce from (2) that for $0 < d \ll 1$,

$$TV(\psi) = 2L + O(d), \tag{4}$$

where the total variation is defined by

$$TV(u) = \sup_{\boldsymbol{p}\in S} \int_\Omega u \,\mathrm{div}\boldsymbol{p}\, d\boldsymbol{x}, \quad \text{where } \; S := \big\{\boldsymbol{p} \in C_c^1(\Omega;\mathbb{R}^2) : |\boldsymbol{p}| \le 1\big\}. \tag{5}$$

Here, $C_c^1(\Omega;\mathbb{R}^2)$ is the space of vector-valued functions compactly supported in Ω with first-order partial derivatives being continuous. In [68], the following modified MS model was considered:

$$\min_{\psi\in\{0,1\},u} \Big\{\mu \int_\Omega (1-\psi)^2 |\nabla u|^2 d\boldsymbol{x} + \frac{\nu}{2}\int_\Omega |u-I|^2 d\boldsymbol{x} + TV(\psi)\Big\}, \tag{6}$$

where μ, ν are positive constants. Observe that $(\psi, u) = (1, I)$ is the minimizer of (6), which is somehow not desirable.

To fix this deficiency, we further modify the functional in (6) by adding an L^1-term that characterizes the area of R_d in (3), that is,

$$A_d = \int_\Omega |\psi|\, d\boldsymbol{x}. \tag{7}$$

Moreover, for ease of implementation, we relax the non-convex set $\{0, 1\}$ to the convex set $[0, 1]$ (see e.g., [11]). In summary, we are concerned with the new MMS model:

$$\min_{u,\psi\in[0,1]} \Big\{\mu \int_\Omega (1-\psi)^2|\nabla u|^2 d\boldsymbol{x} + \frac{\nu}{2}\int_\Omega |u-I|^2 d\boldsymbol{x} + \tau\int_\Omega |\psi|\; d\boldsymbol{x} + TV(\psi)\Big\}, \tag{8}$$

where μ, ν, τ are positive constants. We reiterate that the introduction of the L^1-term (i) can characterize the small area of the narrow band surrounding the targeted open edge; (ii) can rule out the unwanted minimizer of (6); and (iii) does not add any difficulty for the minimization algorithm (cf. [32, 21]).

It is important to point out that Ambrosio and Tortorelli [2, 3] associated Γ with the jump Γ_u of the unknown u, and they proposed the model:

$$\begin{aligned} E_{AT}(u, v) =& \mu \int_\Omega (v^2 + o_\varepsilon)|\nabla u|^2 d\boldsymbol{x} + \frac{\nu}{2}\int_\Omega (u-I)^2 d\boldsymbol{x} \\ &+ \int_\Omega \Big(\varepsilon|\nabla v|^2 + \frac{1}{4\varepsilon}(v-1)^2\Big) d\boldsymbol{x}, \end{aligned} \tag{9}$$

where ε is a sufficient small parameter, and o_ε is any non-negative infinitesimal quantity approaching 0 faster than ε. Here, v plays a role similar to ψ in (8), that is, $v \approx 0$ if $x \in \Gamma_u$, and $v \approx 1$ otherwise. A rigorous analysis (see e.g., [2, 7]) shows that the last term converges to the length of $|\Gamma_u|$ in Γ-convergence sense [22]. The width of transition from $v = 0$ to 1 is about $O(\varepsilon)$. This Ambrosio and Tortorelli approximation (9) has been extensively studied and extended for segmentation and other applications (see e.g., [55, 36, 65, 25, 56]).

3. The minimization algorithms.

This section is to introduce the minimization algorithms for solving (8), which largely consist of the proximity algorithm and split Bregman method for resolving the binary level-set function ψ, and an iterative method for finding u in (8).

Note that we face the situation that the edge set $\Gamma = \{\psi = 1\}$ could be of measure zero in the limiting process. It is questionable to use the dual approach based on the dual formulation of the total variation. For this reason, we resort to the proximity algorithm and split Bregman method, where the use of sub-differential can bypass this.

Motivated by the technique in e.g., [44, 45], we introduce an auxiliary $q = \psi$ to handle the L^1-term and treat the constraint $q = \psi$ by a penalty method. Then the equation (8) can be approximated by the unconstraint problem:

$$\begin{aligned}\min_{\substack{u,q\\ \psi\in[0,1]}} \Big\{\mu\int_\Omega (1-\psi)^2|\nabla u|^2 d\boldsymbol{x} + \frac{\nu}{2}\int_\Omega |u-I|^2 d\boldsymbol{x} \\ + \tau\int_\Omega |q|\, d\boldsymbol{x} + \frac{r}{2}\int_\Omega (q-\psi)^2 d\boldsymbol{x} + TV(\psi)\Big\}.\end{aligned} \tag{10}$$

Using the alternating optimization technique, we split (10) into three subproblems:

- u-subproblem: for fixed ψ, we solve

$$\min_u \Big\{\mathfrak{R}(u) := \mu\int_\Omega (1-\psi)^2|\nabla u|^2 d\boldsymbol{x} + \frac{\nu}{2}\int_\Omega |u-I|^2 d\boldsymbol{x}\Big\}. \tag{11}$$

- ψ-subproblem: for fixed u and q, we solve

$$\min_\psi \Big\{\mathfrak{F}(\psi) := \mu\int_\Omega (1-\psi)^2|\nabla u|^2 d\boldsymbol{x} + \frac{r}{2}\int_\Omega (q-\psi)^2 d\boldsymbol{x} + TV(\psi)\Big\}. \tag{12}$$

- q-subproblem: for fixed ψ, we solve

$$\min_q \Big\{\tau\int_\Omega |q| d\boldsymbol{x} + \frac{r}{2}\int_\Omega (q-\psi)^2 d\boldsymbol{x}\Big\}. \tag{13}$$

One can verify readily that the solution of (13) can be explicitly expressed as (cf. [62, 57]):

$$q = \psi\ \max\Big\{0, 1 - \frac{\tau}{r|\psi|}\Big\}. \tag{14}$$

Next, we present the algorithms for (11) and (12).

3.1. **Fixed-point iterative method for solving u.** We first consider (11). Notice that for fixed ψ, the functional in (11) is convex, so it admits a minimizer. The corresponding Euler-Lagrange equation takes the form:

$$\begin{cases} -2\mu \mathrm{div}\big((1-\psi)^2\nabla u\big) + \nu(u-I) = 0, & \text{in } \Omega, \\ \partial_{\boldsymbol{n}} u|_{\partial\Omega} = 0, \end{cases} \tag{15}$$

where $\boldsymbol{n}$ is the unit outer normal to $\partial\Omega$. Note that ψ is expected to take value 0 at the homogeneous region, i.e., $1-\psi \approx 1$. To solve this elliptic problem with variable coefficient efficiently, we propose to use a fixed-point iterative scheme based on relaxation method (see e.g., [48, 19] for similar ideas). We adopt the difference scheme:

$$\begin{aligned}\nu u_{i,j} = 2\mu[(1-\psi)^2_{i,j+1}(u_{i,j+1}-u_{i,j}) + (1-\psi)^2_{i,j-1}(u_{i,j-1}-u_{i,j}) \\ + (1-\psi)^2_{i-1,j}(u_{i-1,j}-u_{i,j}) + (1-\psi)^2_{i+1,j}(u_{i+1,j}-u_{i,j})] + \nu I_{i,j},\end{aligned} \tag{16}$$

where $u_{i,j} \equiv u(i,j)$ is the approximate solution of (15) at grid point (i,j) with grid size $\hat{h} = 1$ as usual. Then applying the Gauss-Seidel iteration to (16) leads to

$$\begin{aligned}\big(2\mu(C_E + C_W + C_N + C_S) + \nu\big)u^{k+1}_{i,j} = \\ 2\mu\big[C_E u^k_{i,j+1} + C_W u^{k+1}_{i,j-1} + C_N u^{k+1}_{i-1,j} + C_S u^k_{i+1,j}\big] + \nu I_{i,j},\end{aligned} \tag{17}$$

where

$$C_E = (1-\psi^k)^2_{i,j+1},\ C_W = (1-\psi^k)^2_{i,j-1},\ C_N = (1-\psi^k)^2_{i-1,j},\ C_S = (1-\psi^k)^2_{i+1,j}.$$

To speed up the iteration (17), we implement the relaxation method. For this purpose, we use $r_{i,j}^{k+1}$ to denote the residue obtained by subtracting the right-hand side of (17) from the left-hand side of (17). This yields the iterative scheme:

$$u_{i,j}^{k+1} = u_{i,j}^{k} - \omega_1 r_{i,j}^{k+1}, \tag{18}$$

where $\omega_1 > 0$ is the relaxation factor. Collecting (18) yields the new scheme:

$$u_{i,j}^{k+1} = \frac{u_{i,j}^{k} + \omega_1\big(\nu I_{i,j} + 2\mu[C_E u_{i,j+1}^{k} + C_W u_{i,j-1}^{k+1} + C_N u_{i-1,j}^{k+1} + C_S u_{i+1,j}^{k}]\big)}{1 + \omega_1\big(2\mu(C_E + C_W + C_N + C_S) + \nu\big)}. \tag{19}$$

3.2. **Proximity algorithm for solving ψ.** Now, we turn to the subproblem (12). Let us first introduce some basics of the proximity operator and proximity algorithm. Here, we adopt the setting as in [39, 38, 41, 42], and refer to the relevant references therein for more details.

Definition 3.1. Let H be a real Hilbert space, and let h be a convex functional on H, which is not identically equal to ∞. The sub-differential of h at $x \in H$ is the set defined by

$$\partial h(x) := \big\{y \in H : h(z) \geq h(x) + \langle y, z - x\rangle, \ \ \forall z \in H\big\},$$

where $\langle\cdot,\cdot\rangle$ is the inner product of H.

We remark that any $y \in \partial h(x)$ is called a subgradient of h at x. The sub-differential reduces to the classical gradient if h is differentiable. The definition of the proximity operator introduced by Moreau [41, 42] is stated below.

Definition 3.2. Let h and H be the same as in Definition 3.1. Then the proximity operator of h at $x \in H$ is defined by

$$\mathrm{prox}_h(x) \ : = \arg\min_{u\in H}\Big\{\frac{1}{2}\|u - x\|_2^2 + h(u)\Big\},$$

where $\|v\|_2 = \sqrt{\langle v, v\rangle}$.

A very important property is that the sub-differential of a convex function can be characterized by its proximity operator. We refer to [41, 42] for the detailed proof.

Lemma 3.3. *Let h and H be the same as in Definition 3.1. Then for any $x \in H$, we have*

$$y \in \partial h(x) \ \ \textit{if and only if} \ \ x = \mathrm{prox}_h(x + y).$$

Next, we apply Lemma 3.3 to solve the subproblem (12). For this purpose, we define

$$\rho(\psi) = \mu\int_\Omega (1-\psi)^2|\nabla u|^2 d\boldsymbol{x} + \frac{r}{2}\int_\Omega (q-\psi)^2 d\boldsymbol{x}. \tag{20}$$

Then (12) is equivalent to

$$\min_{\psi}\big\{(g\circ B)(\psi) + \rho(\psi)\big\}, \tag{21}$$

where $(g\circ B)(\psi) = TV(\psi)$ with the understanding that

$$g(\cdot) = \|\cdot\|_{L^1(\Omega)}, \quad B = \nabla. \tag{22}$$

Then by using the following result, which can be proved in a fashion similar to Proposition 1 in [39], we can obtain the solution of (21).

Lemma 3.4. *Let g and B be the same as above, and let ρ be a convex functional, e.g., in (20). If $\psi \in H(= L^2(\Omega))$ is a solution of (21), then for any $\alpha, \beta > 0$, there exists $\boldsymbol{s} \in H \times H$ such that*

$$\psi = \operatorname{prox}_{\frac{1}{\alpha}\rho}\Big(\psi - \frac{\beta}{\alpha} B^* \boldsymbol{s}\Big), \tag{23}$$

and

$$\boldsymbol{s} = (\mathbb{I} - \operatorname{prox}_{\frac{1}{\beta}g})(B\psi + \boldsymbol{s}), \tag{24}$$

where B^* is the adjoint operator of B, and $\mathbb{I}$ is the identity operator.

Conversely, if there exist $\alpha, \beta > 0$, $\boldsymbol{s} \in H \times H$, and $\psi \in H$ satisfying (23) and (24), then ψ is a solution of the model (21).

Equipped with Lemma 3.4, we are able to derive the proximity algorithm for (21). By Definition 3.2 and (23), we obtain

$$\psi = \operatorname{prox}_{\frac{1}{\alpha}\rho}\Big(\psi - \frac{\beta}{\alpha} B^* \boldsymbol{s}\Big) = \arg\min_{\varpi}\Big\{\frac{1}{\alpha}\rho(\varpi) + \frac{1}{2}\Big\|\varpi - \Big(\psi - \frac{\beta}{\alpha} B^* \boldsymbol{s}\Big)\Big\|^2_{L^2(\Omega)}\Big\}. \tag{25}$$

In view of (20), we find the Euler-Lagrangian equation:

$$\varpi - M_1 + \frac{1}{\alpha}\big\{-2\mu(1-\varpi)|\nabla u|^2 + r(\varpi - q)\big\} = 0,$$

where

$$M_1 = \psi - \frac{\beta}{\alpha} B^* \boldsymbol{s}.$$

Therefore, we have the representation of the minimizer of (25):

$$\varpi = \frac{rq + 2\mu|\nabla u|^2 + \alpha M_1}{\alpha + r + 2\mu|\nabla u|^2} = \psi. \tag{26}$$

To further enforce $\psi \in [0,1]$, we apply the "truncation" method (see e.g., [20]):

$$\operatorname{Tr}(\psi) = \begin{cases} 0, & \psi < 0, \\ \psi, & 0 \le \psi \le 1, \\ 1, & \psi > 1. \end{cases} \tag{27}$$

It remains to find $\boldsymbol{s}$ in (26) (i.e., in M_1). The special form of the functional g enables us to formulate the sub-differential and proximity operators in a closed form (see, e.g., [38]):

$$\boldsymbol{s} = (B\psi + \boldsymbol{s}) - \max\big\{|B\psi + \boldsymbol{s}| - 1/\beta, 0\big\} \cdot \operatorname{sign}(B\psi + \boldsymbol{s}). \tag{28}$$

We summarize the full algorithm for (10), called the Proximity Algorithm, as follows.

Proximity Algorithm

```
1. Initialization: set $\boldsymbol{s}^0 = \mathbf{0}$, and input $\psi^0$, $u^0$, $\mu, \nu, \alpha, \beta, \omega_1, r, \tau$.
2. For $n = 0, 1, \cdots$,
    (i) Update $u^{n+1}$ using the iteration scheme (19) with initial value $u^n$ and $\psi^n$
        (in place of $\psi$);
    (ii) Update $q^n$ by
             $q^n = \psi^n \max\Big\{0, 1 - \frac{\tau}{r|\psi^n|}\Big\}.$
```

(iii) Update ψ^{n+1} by

$$\psi^{n+1} = \frac{rq^n + 2\mu|\nabla u^{n+1}|^2 + \alpha\big(\psi^n - \frac{\beta}{\alpha}B^*\boldsymbol{s}^n\big)}{\alpha + r + 2\mu|\nabla u^{n+1}|^2},$$

and $\psi^{n+1} = \mathrm{Tr}(\psi^{n+1})$;

(iv) Update $\boldsymbol{s}^{n+1}$ by

$$\boldsymbol{s}^{n+1} = (B\psi^{n+1} + \boldsymbol{s}^n) - \max\big\{|B\psi^{n+1} + \boldsymbol{s}^n| - 1/\beta, 0\big\} \cdot \mathrm{sign}(B\psi^{n+1} + \boldsymbol{s}^n).$$

3. Endfor till some stopping rule meets.

Some remarks are in order.

(a) In practice, we need to run the fixed-point iteration for solving u in Step 2 (i) for several times. For convenience, we denote the number of iterations by J.

(b) In this context, the operators $B = \nabla$ and $B^* = -\mathrm{div}$. We refer to [19] for their discretizations.

(c) It is seen that under the proximal framework, the algorithm appears simple to implement. While the proximity algorithm introduces two additional parameters α and β, the algorithm is robust and not sensitive to choice of α, β (see the numerical examples in next section).

(d) For a noisy image, we find it is beneficial to implement a thresholding technique aiming at sharpening the edges. More precisely, given a threshold $T \in (0, 1)$, we set $\psi^N = 0$, if $\psi^N < T$, and $\psi^N = 1$, otherwise, where N is the number of loops till the stopping rule meets in Step 3.

3.3. **Split Bregman method for solving ψ.** As already mentioned, another way to get around the imposition of total variation over the edge set Γ (with "zero" measure in the limiting case) is to use the split Bregman method [27], which basically integrates the notion of Bregman distance [8] with augmented Lagrangian techniques (cf. [62]). Indeed, this method has been widely used in image processing (see e.g., [13, 26, 54] and the references therein).

We start with introducing an auxiliary variable $\boldsymbol{d}$ and rewrite (21) as

$$\min_{\psi}\Big\{\int_\Omega |\boldsymbol{d}|\, d\boldsymbol{x} + \rho(\psi)\Big\} \quad \text{subject to} \quad \boldsymbol{d} = (d_1, d_2) = \nabla\psi,$$

where $|\boldsymbol{d}| = \sqrt{d_1^2 + d_2^2}$ and $\rho(\psi)$ is defined in (20). Following [27], we formulate the split Bregman iteration as

$$(\psi^{n+1}, \boldsymbol{d}^{n+1}) = \arg\min_{\psi,\boldsymbol{d}}\Big\{\int_\Omega |\boldsymbol{d}|\, d\boldsymbol{x} + \rho(\psi) + \frac{\lambda}{2}\int_\Omega (\boldsymbol{d} - \nabla\psi - \boldsymbol{b}^n)^2 d\boldsymbol{x}\Big\}, \tag{29}$$

for given $\boldsymbol{b}^n$, and

$$\boldsymbol{b}^{n+1} = \boldsymbol{b}^n + \big(\nabla\psi^{n+1} - \boldsymbol{d}^{n+1}\big). \tag{30}$$

It is equivalent to writing (29)-(30) in the component form of $\boldsymbol{d} = (d_1, d_2)$ and $\boldsymbol{b} = (b_1, b_2)$. This yields

$$\begin{aligned}(\psi^{n+1}, d_1^{n+1}, d_2^{n+1}) = \arg\min_{\psi,d_1,d_2}\Big\{&\int_\Omega \sqrt{d_1^2 + d_2^2}\, d\boldsymbol{x} \\ &+ \mu\int_\Omega (1-\psi)^2|\nabla u|^2 d\boldsymbol{x} + \frac{\lambda}{2}\int_\Omega (d_1 - \nabla_x\psi - b_1^n)^2 d\boldsymbol{x} \\ &+ \frac{\lambda}{2}\int_\Omega (d_2 - \nabla_y\psi - b_2^n)^2 d\boldsymbol{x} + \frac{r}{2}\int_\Omega (q-\psi)^2 d\boldsymbol{x}\Big\},\end{aligned} \tag{31}$$

and

$$b_1^{n+1} = b_1^n + \big(\nabla_x \psi^{n+1} - d_1^{n+1}\big), \quad b_2^{n+1} = b_2^n + \big(\nabla_y \psi^{n+1} - d_2^{n+1}\big).$$

The Euler-Lagrange equation of (31) for ψ with fixed d_1 and d_2 is

$$-\lambda\Delta\psi + 2\mu|\nabla u|^2(\psi-1) + r(\psi - q) - \lambda\nabla_x^*(d_1 - b_1^n) - \lambda\nabla_y^*(d_2 - b_2^n) = 0, \tag{32}$$

with the Neumann boundary condition $\partial_{\boldsymbol{n}}\psi = 0$. Following the same idea as in Subsection 3.1, we need to solve the equations about ψ :

$$\psi_{i,j}^{k+1} = \frac{\psi^k + \omega_2\big(F_{i,j}^k + \lambda(\psi_{i+1,j}^k + \psi_{i-1,j}^{k+1} + \psi_{i,j-1}^{k+1} + \psi_{i,j+1}^k)\big)}{1 + \omega_2(2\mu|(\nabla u)_{i,j}^{k+1}|^2 + 4\lambda + r)}, \tag{33}$$

where $\omega_2 > 0$ is the relaxation factor and

$$F := 2\mu|\nabla u|^2 + rq + \lambda\,\nabla_x^*(d_1 - b_1^n) + \lambda\,\nabla_y^*(d_2 - b_2^n).$$

For fixed ψ, the optimality condition of (31) with respect to d_1 and d_2 gives

$$d_1 = \max\left\{\bar{h} - \frac{1}{\lambda}, 0\right\}\frac{h_1}{\bar{h}}, \quad d_2 = \max\left\{\bar{h} - \frac{1}{\lambda}, 0\right\}\frac{h_2}{\bar{h}}, \tag{34}$$

where

$$h_1 = \nabla_x\psi + b_1^n, \quad h_2 = \nabla_y\psi + b_2^n, \quad \bar{h} = \sqrt{h_1^2 + h_2^2}.$$

Now, we are in a position to present the whole algorithm using split Bregman iteration. We replace Steps (iii) and (iv) in the Proximity algorithm by the split Bregman method and summarize the resulted algorithm as follows.

Split Bregman (SB) Algorithm

```
1. Initialization: set $d_1^0 = d_2^0 = b_1^0 = b_2^0 = \mathbf{0}$, and input $\psi^0,\ u^0, \mu, \nu, \omega_1, \omega_2, \lambda, r, \tau$.
2. For $n = 0, 1, \cdots$ ,
    (i) Update $u^{n+1}$ using the iteration scheme (19) with initial value for iteration: $u^n$ and $\psi^n$ (in place of $\psi$);
    (ii) Update $q^n$ by
            $q^n = \psi^n \max\left\{0, 1 - \frac{\tau}{r|\psi^n|}\right\}.$
    (iii) Update $\psi^{n+1}$ using the iteration scheme by (33) with initial value $\psi^n$, $u^{n+1}$ in place of $u$, and $q^n$ in place of $q$, then set $\psi^{n+1} = \mathrm{Tr}(\psi^{n+1})$;
    (iv) Update $d_1^{n+1}$ and $d_2^{n+1}$ by (34) with $\psi^{n+1}$ in place of $\psi$;
    (v) Update $b_1^{n+1}$ and $b_2^{n+1}$ by
            $b_1^{n+1} = b_1^n + \big(\nabla_x\psi^{n+1} - d_1^{n+1}\big), \quad b_2^{n+1} = b_2^n + \big(\nabla_y\psi^{n+1} - d_2^{n+1}\big).$
3. Endfor till some stopping rule meets.
```

As with the proximity algorithm, we can perform the thresholding technique to sharpen the edges (see (d) in the remarks).

3.4. **Convergence analysis.** Noticed that the SB algorithm includes "inner" and "outer" iteration. It is seen from (19) and (33) that some "inner" iterations are needed to solve the u-subproblem and ψ-subproblem, so we next show the convergence of the iterative schemes. We postpone its proof to Appendix A.

Theorem 3.5. *The sequence* $\{u^k\}_{k\geq 0}$ *(resp.* $\{\psi^k\}_{k\geq 0}$*) generated by the inner iterative scheme* (19) *(resp.* (33)*) converges to the solution of the problem* (11) *(resp.* (12)*).*

Following the argument for the convergence analysis in [13, 71], we are able to prove the convergence of the whole algorithm. Note that from the first order optimality condition, we derive

(35)
$$\begin{aligned}
&2\mu\nabla^*((1-\psi^*)^2\nabla u^{n+1}) + \tilde{u}^{n+1} = 0, && \tilde{u}^{n+1} \in \partial H_u(u^{n+1});\\
&-\lambda\triangle\psi^{n+1} + 2\mu(|\nabla u^*|^2(\psi^{n+1}-1)) + \tilde{\psi}^{n+1} - \lambda\nabla(\boldsymbol{d}^n - \boldsymbol{b}^n) = 0, && \tilde{\psi}^{n+1} \in \partial H_\psi(\psi^{n+1});\\
&\boldsymbol{p}^{n+1} + \lambda(\boldsymbol{d}^{n+1} - \nabla\psi^{n+1} - \boldsymbol{b}^n) = 0, && \boldsymbol{p}^{n+1} \in \partial|\boldsymbol{d}^{n+1}|;\\
&\boldsymbol{b}^{n+1} = \boldsymbol{b}^n + (\nabla\psi^{n+1} - \boldsymbol{d}^{n+1}).
\end{aligned}$$

with (u^*,ψ^*) being the pair of minimizers of the subproblems (11)-(12), and

$$H_u(u^{n+1}) = \frac{\nu}{2}\|u^{n+1}-I\|^2, \quad H_\psi(\psi^{n+1}) = \frac{r}{2}\|\psi^{n+1}-q\|^2.$$

Theorem 3.6. *Let* (u^*,ψ^*) *be the pair of minimizers of the subproblems* (11)-(12)*. Given* $\omega_1,\omega_2>0$ *in the alternating split Bregman iteration scheme, we have*

$$\lim_{n\to+\infty}\mathfrak{F}(\psi^n) = \mathfrak{F}(\psi^*), \quad \lim_{n\to+\infty}\mathfrak{R}(u^n) = \mathfrak{R}(u^*). \tag{36}$$

Moreover, if the pair of minimizers is unique, we get

$$\lim_{n\to\infty}\|\psi^n-\psi^*\| = 0, \quad \lim_{n\to\infty}\|u^n-u^*\| = 0. \tag{37}$$

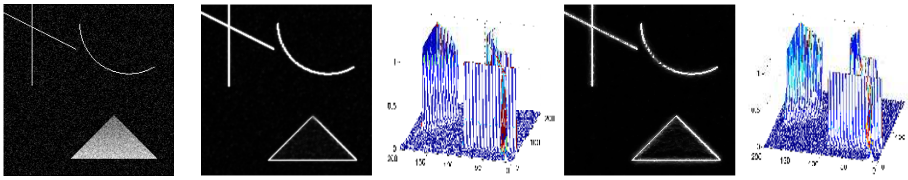
We provide the proof in Appendix B. For the Proximity algorithm, we can also get the convergence theorem in the same way as Theorem 3.6, so we omit the details.

4. **Numerical experiments.**

In this section, we provide ample numerical results to show the performance of the proposed model and algorithms from various perspectives, and to compare with the relevant methods.

4.1. **Convergence test.** In the first experiment, we consider extraction of simple open or closed edges from images polluted by small Gaussian noise (see Figure 2 (a)), and intend to show that the level-set function ψ converges to the indictor function of the narrow band around the targeted edge. We see that with a reasonable number of iterations, ψ^n converges to the expected indicator function, from which we identify the open or closed edges.

4.2. **Algorithm for Ambrosio and Tortorelli model.** As aforementioned, our approach shares certain similarity with the approach by Ambrosio and Tortorelli (AT). Hence, we feel compelled to make a comparison. Many algorithms are proposed to solve the model (9) (see e.g., [12]).



(a) Given image (b) Edge-SB (c) ψ-SB (d) Edge-Proximity (e) ψ-Proximity

Figure 2. Convergence of the level-set function ψ. (a) Input noisy image (with Gaussian noise: $\sigma = 0.01$); (b) Edge segmented by SB algorithm with 80 iterations; (c) ψ-profile obtained by SB algorithm with 80 iterations. (d) Edge segmented by Proximity algorithm with 300 iterations; (e) ψ profile obtained by Proximity algorithm with 300 iterations. The parameters $\mu = 100, \nu = 10, \omega_1 = 5 \times 10^{-4}, r = 100, \tau = 1$ and $\lambda = 0.01, \omega_2 = 1$ in SB algorithm and $\alpha = 0.01, \beta = 0.01$ in Proximity algorithm, respectively. The inner iteration for u is set to be $J = 3$.

The Euler-Lagrange equations of the AT model (9) read

$$\begin{cases} u = \dfrac{2\mu}{\nu}\operatorname{div}\big((v^2 + o_\epsilon)\nabla u\big) + I, & \text{in } \Omega, \\ v\Big(\dfrac{1}{4\epsilon} + \mu|\nabla u|^2\Big) = \epsilon\Delta v + \dfrac{1}{4\epsilon}, & \text{in } \Omega, \\ \dfrac{\partial u}{\partial \boldsymbol{n}} = \dfrac{\partial v}{\partial \boldsymbol{n}} = 0, & \text{on } \partial\Omega. \end{cases} \tag{38}$$

As before, we use the fixed-point iterative method to solve this system (see Subsection 3.1). More precisely, we solve u by the iterative scheme:

$$u_{i,j}^{n+1} = \frac{u_{i,j}^n + \omega_3\Big(\nu I_{i,j} + 2\mu[C_E u_{i,j+1}^n + C_W u_{i,j-1}^{n+1} + C_N u_{i-1,j}^{n+1} + C_S u_{i+1,j}^n]\Big)}{1 + \omega_3\Big(2\mu(C_E + C_W + C_N + C_S) + \nu\Big)}, \tag{39}$$

where $\omega_3 > 0$ is the relaxation factor and

$C_E = (v^n)_{i,j+1}^2 + o_\epsilon, C_W = (v^n)_{i,j-1}^2 + o_\epsilon, C_N = (v^n)_{i-1,j}^2 + o_\epsilon, C_S = (v^n)_{i+1,j}^2 + o_\epsilon.$

We solve v by the fixed-point iteration:

$$v_{i,j}^{n+1} = \frac{v_{i,j}^n + \omega_4\Big(\frac{1}{4\epsilon} + \epsilon(v_{i+1,j}^n + v_{i-1,j}^{n+1} + v_{i,j-1}^{n+1} + v_{i,j+1}^n)\Big)}{1 + \omega_4(\frac{1}{4\epsilon} + 2\mu|\nabla u_{i,j}^{n+1}|^2 + 4\epsilon)}, \tag{40}$$

where $\omega_4 > 0$ is the relaxation factor. Now, we present full algorithm as follows.

AT Algorithm

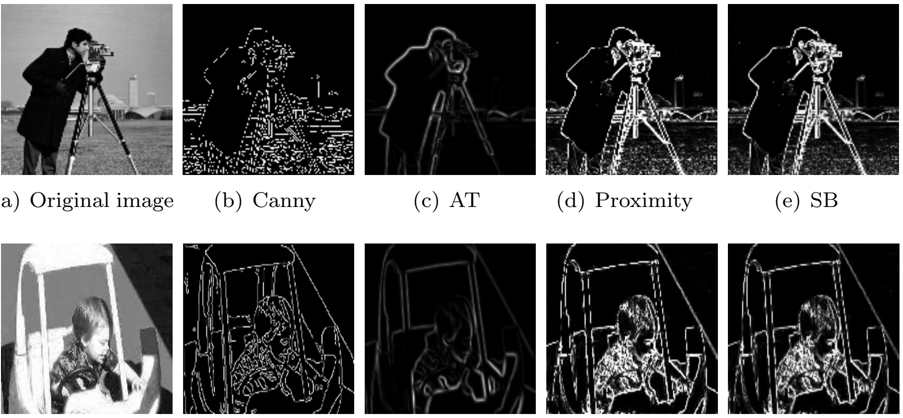
```
1. Initialization: set $o_\epsilon = \epsilon^{\hat{p}}(\hat{p} > 1)$, and input $u^0$, $v^0$, $\mu, \nu$, $\hat{p}$, $\epsilon$, $\omega_3, \omega_4$.
2. For $n = 0, 1, \cdots$,
    (i) Update $u^{n+1}$ by (39);
    (ii) Update $v^{n+1}$ by (40).
3. Endfor till some stopping rule meets.
```

4.3. **Comparison.** Next, we compare the following four algorithms: AT algorithm stated above, the Canny edge detector [15] (based on the code "edge(I,'canny',THRESH)" in `Matlab`), the Proximity algorithm and the split Bregman (SB) algorithm.

Here, we set the stopping rule by fixed numbers of iteration (e.g., Figure 3) or using the relative error (e.g., Figure 7)

$$E_{l^1}(u^{n+1}, u^n) := \|u^{n+1} - u^n\| \leq \eta, \tag{41}$$

where $\|\psi\| = \sum |\psi_i|$, and the summation is over all the pixels, for a prescribed tolerance $\eta > 0$. The initial value u^0 is set to be the noisy image I and ψ^0 is set to be a null matrix, since most of ψ values are 0 as a result. For simplicity, we let $\omega_3 = \omega_4 = \omega_2$ in AT algorithm. The choice of the parameters are specified in the captions of the figures.


(a) Original image (b) Canny (c) AT (d) Proximity (e) SB

(f) Original image (g) Canny (h) AT (i) Proximity (j) SB

Figure 3. Comparisons of four algorithms for clean images. Column 1: the original images; Column 2: edge detected by Canny method; Column 3: edge detected by AT algorithm with 100 iterations; Column 4: edge detected by Proximity algorithm with 100 iterations; Column 5: edge detected by SB algorithm with 50 iterations. From the numerical results, we can get Proximity algorithm and SB algorithm produce better results than the other two algorithms.

In the second experiments, we test two real clean images (see Figure 3 (a) and Figure 3 (f)), and generally choose the same parameters in Proximity algorithm, SB algorithm and AT algorithm with

$$\mu = 100, \quad \nu = 10, \quad \omega_1 = 5 \times 10^{-4}, \quad r = 100, \quad \eta = 1.0e - 4$$

and the parameters $\lambda = 0.01, \omega_2 = 1, \tau = 1$ in SB algorithm, $\epsilon = 0.01, \hat{p} = 2$ in AT algorithm and $\alpha = 0.01, \beta = 0.01, \tau = 1$ in Proximity algorithm, respectively. We present in Figure 3 the input images and the results detected by four different algorithms. We observe from Figure 3 that Proximity algorithm and SB algorithm outperform the Canny method and AT algorithm. The proposed two algorithms are able to detect all the meaningful edges.

In the third experiments, we turn to the comparison of four algorithms for the noisy images in Figure 4 (a) and Figure 4 (f) (small Gaussian noise with $\sigma = 0.01$) and Figure 4 (k) and Figure 4 (p) (large Gaussian noise with $\sigma = 0.04$), respectively.

In this situation, solving u equation usually not only need one iteration in order to smooth the noisy image. Here, we apply 2 iterations to solve u equation for Gaussian noise with $\sigma = 0.01$, while 3 iterations for Gaussian noise with $\sigma = 0.04$. Here, τ is set to be 0.05. We choose the same parameters as the previous example except $\lambda = 0.001$, and present in Figure 4 the original images and the detected results obtained by four different algorithms. We observe from Figure 4 (a) -4 (j) that when the noise level is low, all algorithms produce satisfactory results. However, when we increase the level of noise, Canny method fails to find the correct edges. The other three algorithms lead to acceptable results, but Proximity algorithm and SB algorithm yield better results and AT algorithm smoothes some details of the edge.

In the fourth experiments, we also compared our algorithm with another wholly different algorithm in [34]. B. Llanas et al. [34] proposed an Edge Detection by Adaptive Splitting (EDAS-d with $d = 1, 2$) algorithm to approximate the jump discontinuity set of functions defined on subsets of R^d, based on adaptive splitting of the domain of the function guided by the value of an average integral. The algorithm is easy to compute some integrals and no need to solve any partial differential equation. The algorithm requires the following positive real parameters:

- E_1: maximum local error of the approximant L and detection threshold.
- E_2: the approximation error of the points in jump discontinuity set Γ^J and stopping criterion.
- E_3: the minimum magnitude of jump reported.
- E_4: exploration parameter.
- E_5: adaptivity of cubature formulas parameter.

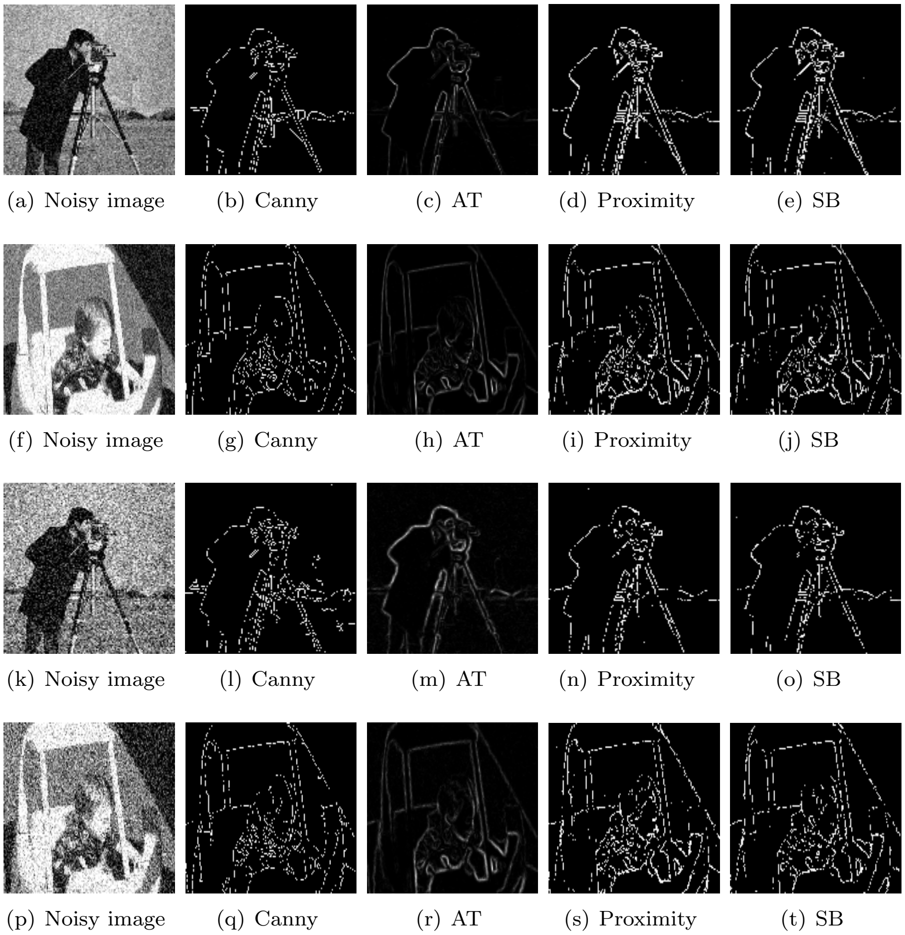
We have applied EDAS-1 algorithm for three images: the "line" image (see Figure 2 (a)), cameraman image (see Figure 4 (a)), kitten image (see Figure 9 (a)). Figure 5 shows the obtained edges with

$$E_1 = 1.0e-3, \quad E_2 = 1.0e-7, \quad E_4 = 10, \quad E_5 = 10, \quad E_3 = 200,$$

for Figure 5 (c), and $E_3 = 25$ for the other images. The parameter E_3 significantly affects the accuracy of edge detection. The choice of larger E_3 might not be able to detect some edges. We notice that some pixels of some edge are missed out sometimes by EDAS-1 algorithm (which has been also pointed out in [34]). For the noisy line image (see Figure 3 (a)), Figure 5 (c) displays some spurious edges.

4.4. **Sensitivity study of the parameters.** Here we show some basic guides to set these parameters in SB algorithm and Proximity algorithm. Firstly the results are not so sensitive for the initial values u^0 and v^0 which can be set to be any constant values between zero and one. But different initial values exactly impact the iterative times with different iterations, which will be described in the forthcoming subsection.

Secondly, a smaller value of μ will have less of a smoothing effect in the restored image and exist more details in the resulting edge detection image, while a larger μ will have a greater smoothing effect in the restored image. Notice that ν has the opposite effects compared with μ. Large μ/ν yields a blurry, over-smoothed restored image, while small μ/ν corresponds to very little noise removal. Thus usually we keep optimal μ/ν value. The threshold $T \in (0, 1)$ is set to distinguish edge points and non-edge points clearly. Larger T may miss some tiny edges, while small one will make the edges look noisy. Usually, T is set to be 0.5 for small noise, while 0.8 for large noise. If the parameter τ is larger than 1 especially for noisy image,



(a) Noisy image (b) Canny (c) AT (d) Proximity (e) SB

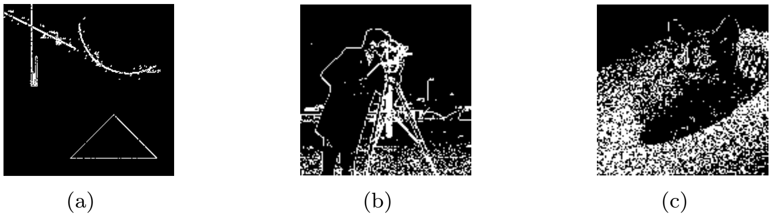
(f) Noisy image (g) Canny (h) AT (i) Proximity (j) SB

(k) Noisy image (l) Canny (m) AT (n) Proximity (o) SB

(p) Noisy image (q) Canny (r) AT (s) Proximity (t) SB

Figure 4. Comparisons of four algorithms for noisy images. Column 1: the original images; Column 2: edge detected by Canny method; Column 3: edge detected by AT algorithm; Column 4: edge detected by Proximity algorithm; Column 5: edge detected by SB algorithm with 50 iterations. Row 1 and Row 2: $T = 0.5, \tau = 0.05, \lambda = 0.001$, 100 iterations; Row 3 and Row 4: $T = 0.8, \tau = 0.05, \lambda = 0.001$, 200 iterations. Proximity algorithm and SB algorithm produce better results than the other two algorithms.

the detected edges will be noisy. Thus, the parameter τ usually needs to be smaller than 1. The penalty parameter r usually is set to be large, since sufficiently large r means that (10) converges to (8). The larger λ implies smoother restored image and simpler edges with losing tiny edges. The relaxation parameter ω_1, ω_2 usually are set to be smaller than 1. Large ω_1, ω_2 values will speed up the convergence process. The obtained results depend mildly on α and β. In the following subsections, we will especially discuss several different parameters and show our model and algorithms are robust and effective.



(a) (b) (c)

Figure 5. Results obtained by EDAS-1 algorithm [34] with $E_1 = 1.0e-3, E_2 = 1.0e-7, E_3 = 25, E_4 = 10, E_5 = 10$.

4.5. **Selection of the initial values u^0 and v^0.** In this subsection, we provide some directions into the selection of the initial values u^0 and v^0. The initial values impacts the number of iterations, computational time, and detected results.

In Figure 6, we test two real clean images (see Figure 3 (a) and Figure 3 (f)) by Proximity algorithm and SB algorithm with different u^0 and v^0 and fix the other parameters as before. Four testing sets are available in Figure 6:

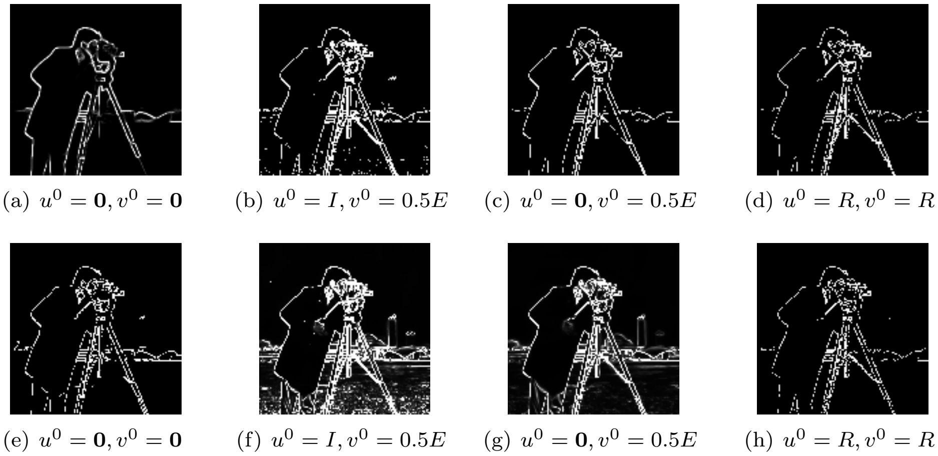
$$u^0 = \mathbf{0}, v^0 = \mathbf{0}; u^0 = I, v^0 = 0.5E; u^0 = \mathbf{0}, v^0 = 0.5E; u^0 = R, v^0 = R;$$

where $\mathbf{0}$ is zero matrix, E is identity matrix and R means the random value matrix. The tolerance $\eta = 1.0e-8$, which needs to be small, since it needs more iterations for some initial values far away from the true solutions, especially for Proximity algorithm. We see that Proximity algorithm and SB algorithm shows the robustness for different initial values in Figure 6. We let $\tau = 0.05$ from column 1 to column 3, and 0.5 for random value matrix as initial values in column 4. The results are shown in Figure 6. We also test the initial values being any constant between 0 and 1, and obtain that both Proximity algorithm and SB algorithm can obtain satisfactory results. Thus especially for real image with a lot of tiny edges, it is better to set the initial values $u^0 = I, v^0 = \mathbf{0}$, which save the computational time and get satisfactory detected edges.

4.6. **Selection of the parameters α and β.** Hereafter, we provide some insights into the selection of parameters α and β. We test Proximity algorithm with different α and β in Figure 7 (a) and Figure 7 (f). We fix the other parameters as before, but choose α, β ranging from 0.01 to 0.5. We see that Proximity algorithm enjoys a similar performance in these tests.

4.7. **Selection of the parameters μ and ν.** Now, we provide some insights into the selection of parameters μ and ν. We test SB algorithm with different μ and ν in Figure 8 (a) and Figure 8 (f). Since the parameter μ/ν impacts the denoised effects, large μ/ν yields a blurry, oversmoothed restored image, and small μ/ν corresponds to very little noise removal. We fix the other parameters as before, but choose μ ranging from 200 to 1000 and ν ranging from 0.1 to 50. We see that SB algorithm enjoys a similar performance in these tests.

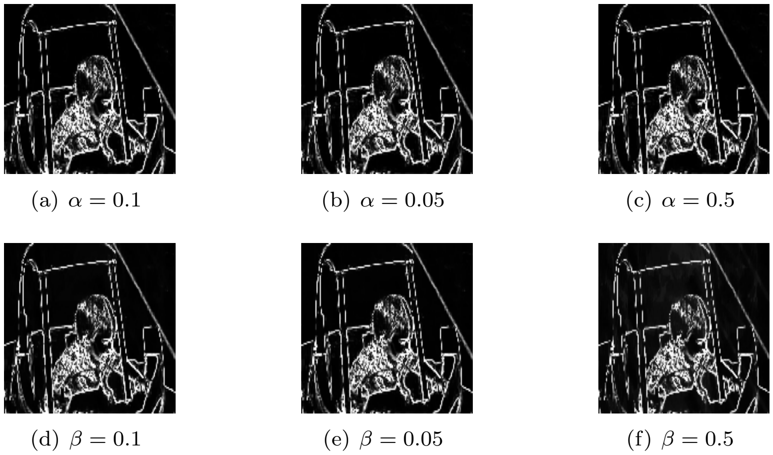
4.8. **Some more tests.** We next test more images by using the Proximity algorithm and SB algorithm in Figure 9. In these tests, both algorithms are terminated with 100 iterations. We refer to the caption of Figure 9 for the choices of the parameters of these two algorithms. It is seen that both the two algorithms yield good results.



(a) $u^0 = \mathbf{0}, v^0 = \mathbf{0}$ (b) $u^0 = I, v^0 = 0.5E$ (c) $u^0 = \mathbf{0}, v^0 = 0.5E$ (d) $u^0 = R, v^0 = R$

(e) $u^0 = \mathbf{0}, v^0 = \mathbf{0}$ (f) $u^0 = I, v^0 = 0.5E$ (g) $u^0 = \mathbf{0}, v^0 = 0.5E$ (h) $u^0 = R, v^0 = R$

Figure 6. Results obtained by Proximity algorithm and SB algorithm with different u^0 and v^0. Row 1: SB algorithm. Row 2: Proximity algorithm. Column 1: $u^0 = \mathbf{0}, v^0 = \mathbf{0}$ and 400 iterations for SB algorithm, and 1000 iterations for Proximity algorithm; Column 2: $u^0 = I, v^0 = 0.5E$ and 100 iterations for SB algorithm, and 1000 iterations for Proximity algorithm; Column 3: $u^0 = \mathbf{0}, v^0 = 0.5E$ and 400 iterations for SB algorithm, and 3000 iterations for Proximity algorithm; Column 4: $u^0 = R, v^0 = R$, $\tau = 0.5$ and 1000 iterations for SB algorithm, and 8500 iterations for Proximity algorithm.

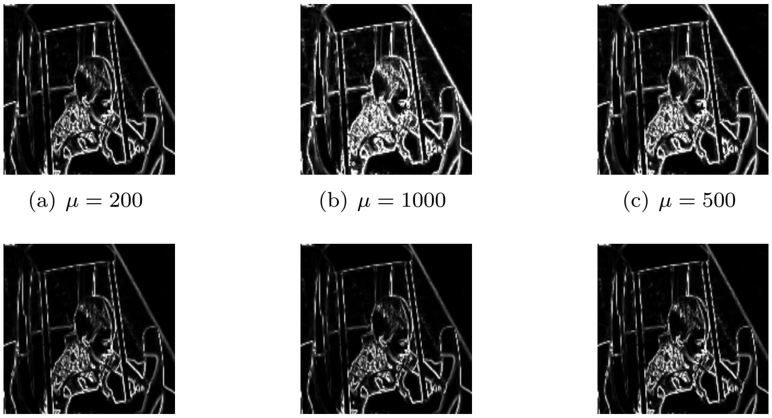


(a) $\alpha = 0.1$ (b) $\alpha = 0.05$ (c) $\alpha = 0.5$

(d) $\beta = 0.1$ (e) $\beta = 0.05$ (f) $\beta = 0.5$

Figure 7. Results obtained by Proximity algorithm with different α and β. Row 1: results obtained with different α; Row 2: results obtained with different β. We indicate the value of α and β.

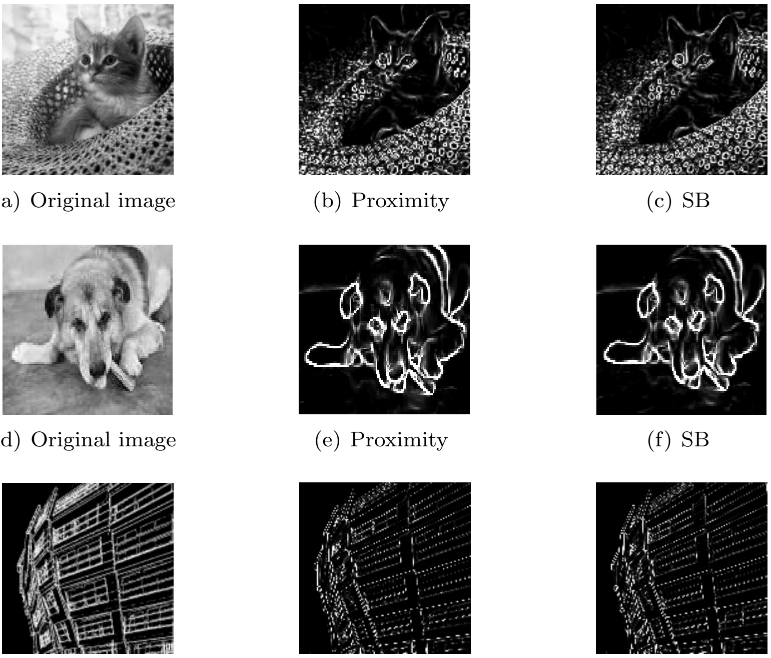
Although the Canny detector is not very desirable for noisy images, the detected edges are usually acceptable for clean images. In practice, a pre-smoothing of the noisy image oftentimes favours the Canny method. Cai et al. [14] proposed a two-stage segmentation algorithm for a convex variant of the Mumford-Shah model,



(a) $\mu = 200$ (b) $\mu = 1000$ (c) $\mu = 500$

(d) $\nu = 1$ (e) $\nu = 0.1$ (f) $\nu = 50$

Figure 8. Results obtained by SB algorithm with different μ and ν. Row 1: results obtained with different μ; Row 2: results obtained with different ν. We indicate the value of μ and ν.



(a) Original image (b) Proximity (c) SB

(d) Original image (e) Proximity (f) SB

(g) Original image (h) Proximity (i) SB

Figure 9. Results obtained by Proximity algorithm and SB algorithm. In the computation, we take $\mu = 100, \nu = 10, r = 100, \lambda = 0.01, \alpha = 0.1, \beta = 0.1, \tau = 0.05$. Both Proximity algorithm and SB algorithm can produce satisfactory results.

which incorporated with smoothing and thresholding. Here, we adopt the smoothing step in [14] and then use Canny detector. In Figure 10, we output the edges with $\mu = 5; \sigma = 2; k = 4; \lambda = 100$, for the images in Figure 2 (a), Figure 4 (a) and Figure

4 (f). We see that better detection can be obtained from our approach (see, e.g., Figure 4).

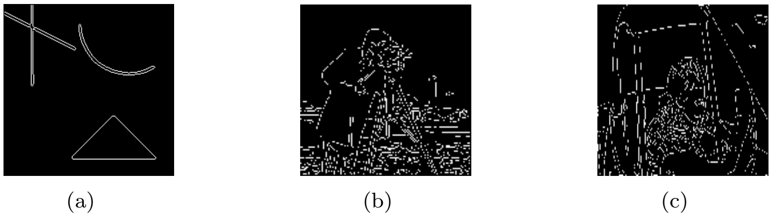

(a) (b) (c)

Figure 10. Results obtained by the algorithm in Cai et al. [14] with $\mu = 5; \sigma = 2; k = 4; \lambda = 100$.

5. **Pretreatment before edge detection.** In order to improve the accuracy of edges and accelerate the computation speed, coupled pretreatments are carried out before proceeding the above model (8). Similarly, a eight direction box spline tight frame with noise removal before edge detection is constructed in [28]. In the following, we show the coupled pretreatments including two steps: first to remove noise from noisy images, then to locate non-edge pixels through hard thresholding the norm of gradients.

Given a noisy image I, we first apply the split Bregman method [62, 27] to solve the ROF model [51]:

$$\mu \int_{\Omega} |\nabla u| d\boldsymbol{x} + \frac{\nu}{2} \int_{\Omega} (u - I)^2 d\boldsymbol{x}, \tag{42}$$

which yields a denoised (i.e., pre-smoothed) image.

Then we set a threshold value ζ to simply distinguish the non-edge pixels for the pre-smoothed image. That is to say, if the gradient of one pixel is less than the threshold value ζ, set the pixel to be a non-edge pixel, i.e., $\psi = 0$. The pretreatment will save the computation since the ψ values of these pixels will maintain unchanged in the remaining computational process. Thus how to choose the value of ζ is important. On the one hand, a big threshold leads to edges loss. But on the other hand, a small threshold increases false edges. In all the experiments, we let the threshold value $\zeta = 0.01$.

At last, we just need to proceed the minimization model (8) in Section 2 to recalculate the ψ values of the remaining pixels. In our experiments, we let $\mu = 100, \nu = 1000, \eta = 1.0e - 6, \tau = 1$ for the experiments in Figure 11. Here is the pretreatment split Bregman (Pre-SB) algorithm:

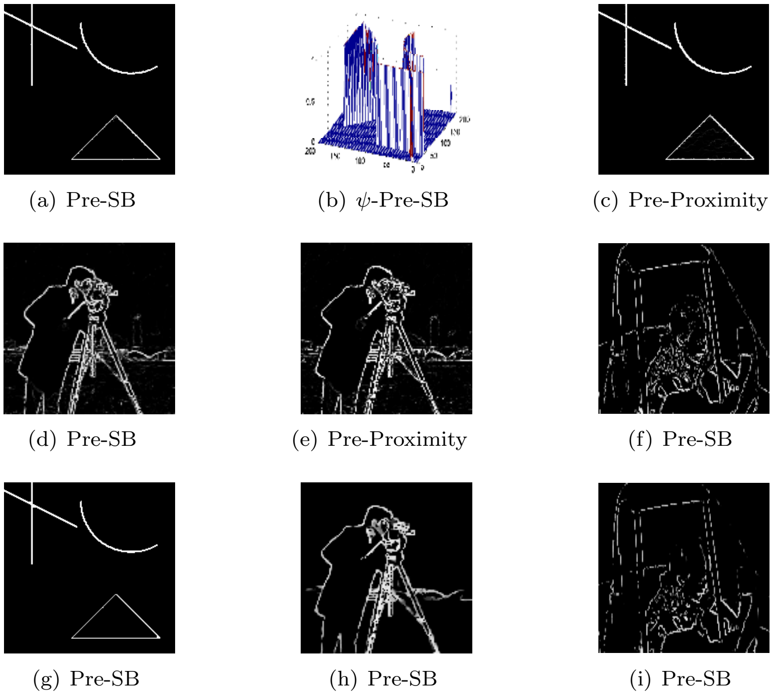
Pretreatment Split Bregman Algorithm

```
1. Pre-smooth the noisy image.
2. Compute the gradient for each pixel of the pre-smoothed image.
3. Fix $\psi = 0$ on these non-edge pixels whose gradient magnitudes are less than
   a threshold value $\zeta$.
4. Compute the remaining pixels using **Split Bregman Algorithm**.
```

Using the Proximity algorithm to solve the model (8), we also can get the pretreatment proximity algorithm (Pre-Proximity) as follows:

Pretreatment Proximity Algorithm

1. Pre-smooth the noisy image.
2. Compute the gradient for each pixel of the pre-smoothed image.
3. Fix $\psi = 0$ on these non-edge pixels whose gradient magnitudes are less than a threshold value ζ.
4. Compute the remaining pixels using **Proximity Algorithm**.



(a) Pre-SB (b) ψ-Pre-SB (c) Pre-Proximity

(d) Pre-SB (e) Pre-Proximity (f) Pre-SB

(g) Pre-SB (h) Pre-SB (i) Pre-SB

Figure 11. Results obtained by Pre-Proximity algorithm and Pre-SB algorithm for noisy images (Figure 2 (a), Figure 4 (a) and Figure 4 (f)). In the computation, we take $\mu = 100, \nu = 1000, r = 100, \lambda = 0.01, \alpha = 0.1, \beta = 0.1, \eta = 1.0e - 6, \tau = 1$. Last row: edges with 5000 iterations. Both Pre-Proximity and Pre-SB algorithm can produce satisfactory results.

Figure 11 show the results for the noisy images in Figure 4 (a) and Figure 4 (f) (corrupted by Gaussian noise with $\sigma = 0.01$). Since ψ are fixed to be 0 on non-edge pixels in the third step of Pre-SB algorithm, the edges in Figure 11 are clear, not like the noisy edge images in Figure 4 where ψ are not zero on the non-edge pixels. To show that the convergence of Pre-SB algorithm is better than SB algorithm, we have several tests (see the last row in Figure 11) with 5000 iterations. In this situation, the stopping tolerance η is much less than $1.0e - 16$. Of course, the tiny images sometimes disappear with too many iterations (see Figure 11 (i)).

To show the coupled pretreatments indeed accelerate the computation speed, we will compare the iterations and computation times in Table 1. For Pre-SB

	SB		Pre-SB		Proximity		Pre-Proximity	
	time	iterations	time	iterations	time	iterations	time	iterations
Figure 3 (a)	3.56	80	2.38	57	2.45	58	2.12	53
Figure 3 (f)	3.93	79	2.57	55	2.41	55	2.32	52

Table 1. Computation time and iterations of the cameraman and baby images.

algorithm and Pre-Proximity algorithm, satisfactory edges can be gotten by small $\frac{\mu}{\nu}$ which can not denoise noisy image for SB algorithm and Proximity algorithm and the edges can not be detected. So here we only test the clean images (see Figure 3 (a) and 3 (f)). Table 1 shows that the computation time and iterations with $\mu = 100, \nu = 1000, \eta = 1.0e-4, \tau = 1$ and the other parameters are the same as above. The detected edges are satisfactory and we will not show the results here. These data show that the coupled pretreatments need less iterations and less computation times.

6. **Extension to color images.** Finally, we extend the proposed model and algorithm for edge detection involving color images. To this end, let us denote

$$\boldsymbol{I} = (I_r, I_g, I_b), \quad \boldsymbol{u} = (u_r, u_g, u_b), \quad \boldsymbol{\psi} = (\psi_r, \psi_g, \psi_b).$$

Meanwhile, we introduce the multichannel TV (cf. [63, 9, 52]):

$$TV(\boldsymbol{\psi}) = \int_\Omega \sqrt{\sum_{i=r,g,b} |\nabla \psi_i|^2 d\boldsymbol{x}}.$$

The counterpart of the model (8) takes the form

$$\min_{\boldsymbol{u},\boldsymbol{\psi}} \Big\{ E_{CL}(\boldsymbol{u},\boldsymbol{\psi}) = \mu \int_\Omega (1-\boldsymbol{\psi})^2 |\nabla \boldsymbol{u}|^2 d\boldsymbol{x} + \frac{\nu}{2}\int_\Omega |\boldsymbol{u}-\boldsymbol{I}|^2 d\boldsymbol{x} + TV(\boldsymbol{\psi}) + \int_\Omega |\boldsymbol{\psi}| d\boldsymbol{x} \Big\}.$$

Similarly, the algorithm in Section 3 can be straightforwardly extended to this vectorial setting. Here, we just sketch the split Bregman method for solving $\boldsymbol{\psi}$. We start with introducing the new variable:

$$\boldsymbol{d} = (\boldsymbol{d}_r, \boldsymbol{d}_g, \boldsymbol{d}_b) = \nabla \boldsymbol{\psi} = (\nabla \psi_r, \nabla \psi_g, \nabla \psi_b),$$

and apply the split Bregman iteration

$$\begin{aligned}(\boldsymbol{\psi}^{n+1}, \boldsymbol{d}^{n+1}) = \arg\min_{\boldsymbol{\psi},\boldsymbol{d}} \Big\{ &\int_\Omega \sqrt{\sum_{i=r,g,b} |\boldsymbol{d}_i|^2}\, d\boldsymbol{x} + \mu \int_\Omega (1-\boldsymbol{\psi})^2 |\nabla \boldsymbol{u}|^2 d\boldsymbol{x} \\ &+ \frac{\lambda}{2} \sum_{i=r,g,b} \int_\Omega |\boldsymbol{d}_i - \nabla \psi_i - \boldsymbol{b}_i^n|^2 d\boldsymbol{x} + \int_\Omega |\boldsymbol{q}| d\boldsymbol{x} + \frac{r}{2}\int_\Omega (\boldsymbol{q}-\boldsymbol{\psi})^2 d\boldsymbol{x} \Big\},\end{aligned}$$

and

$$\boldsymbol{b}^{n+1} = \boldsymbol{b}^n + \big(\nabla \boldsymbol{\psi}^{n+1} - \boldsymbol{d}^{n+1}\big).$$

Thus, we can solve ψ_i^{n+1} as in (33) with fixed $\boldsymbol{d}$ for each $i = r, g, b$, respectively. Meanwhile, the solution of $\boldsymbol{d}^{n+1}$ with fixed $\boldsymbol{\psi}$ can be obtained by the multi-dimensional shrinkage formula (cf. [69]):

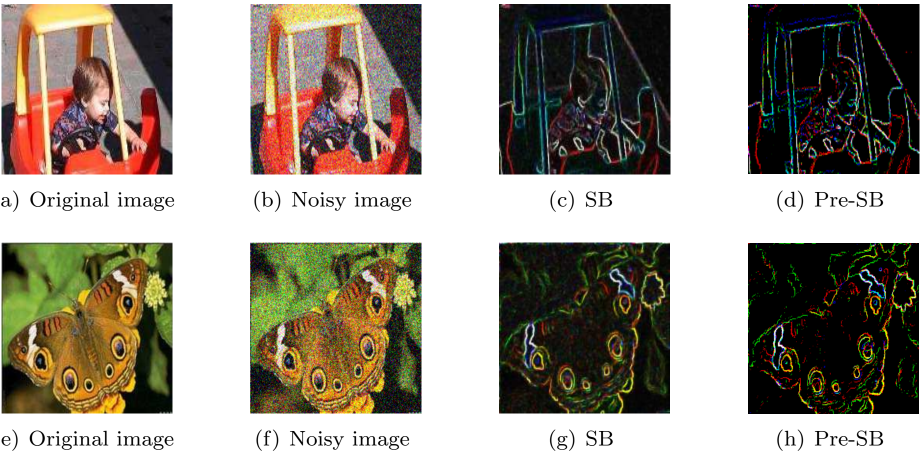
$$\boldsymbol{d} = \max\Big\{\boldsymbol{h} - \frac{1}{\lambda}, 0\Big\} \frac{\boldsymbol{h}_1}{\boldsymbol{h}},$$

where

$$\boldsymbol{h}_1 = \nabla \boldsymbol{\psi} + \boldsymbol{b}^n, \quad \boldsymbol{h} = |\nabla \boldsymbol{\psi} + \boldsymbol{b}^n|.$$

We test the algorithms on the color images with Gaussian noise ($\sigma = 0.04$), and refer to the caption of Figure 12 for the choices of parameters. In the following experiments, μ is changed to be 200.

Indeed, there has been limited work on edge detection involving color images. Our proposed model and algorithms provide viable means for such a purpose.


(a) Original image (b) Noisy image (c) SB (d) Pre-SB

(e) Original image (f) Noisy image (g) SB (h) Pre-SB

Figure 12. Results obtained by SB algorithm with the parameters: $\mu = 200, \nu = 10, r = 100, \tau = 1, \lambda = 1e-2$ and 80 iterations.

Acknowledgments. The authors wish to thank the anonymous referees for many constructive comments which lead to much improvement of the paper. The authors are indebted to Professors Bernardo Llanas and Sagrario Lantarón from indictor Politcnica de Madrid, Professor Tieyong Zeng from Hong Kong Baptist University for sharing the codes.

Appendix A. **Proof of Theorem 3.5.** The proof is motivated by the idea in [62]. Rewriting the iterative scheme (19) as

(A.43)
$$\begin{aligned}&[1+\omega_1(2\mu(C_E+C_W+C_N+C_S)+\nu)]u_{i,j}^{k+1} - 2\mu\omega_1 C_W u_{i,j-1}^{k+1} - 2\mu\omega_1 C_N u_{i-1,j}^{k+1}\\ =&u_{i,j}^k + 2\mu\omega_1 C_S u_{i+1,j}^k + 2\mu\omega_1 C_E u_{i,j+1}^k + \nu\omega_1 I_{i,j},\end{aligned}$$

where

$C_E = (1-\psi^k)^2_{i,j+1},\ C_W = (1-\psi^k)^2_{i,j-1},\ C_N = (1-\psi^k)^2_{i-1,j},\ C_S = (1-\psi^k)^2_{i+1,j}.$

Thus it can be written as the matrix formula

$$(L + \acute{D} + \acute{U})U^{k+1} = (E + \grave{D} + \grave{U})U^k + I, \tag{A.44}$$

where U^k is made up of elements $u_{i,j}^k$ column-wise, E is identity matrix, the matrix L is diagonal matrix with

$$L_{i,i} = 1 + 2\omega_1\mu(C_E + C_W + C_N + C_S) + \omega_1\nu$$

and 0 otherwise, the matrix $\acute{U}$ is lower diagonal matrix with the minor diagonal element $\acute{U}_{i-1,i} = 2\mu\omega_1 C_W$ and 0 otherwise, the matrix $\acute{D}$ is upper diagonal matrix with the minor diagonal element $\acute{D}_{i,i-1} = 2\mu\omega_1 C_N$ and 0 otherwise, the matrix $\grave{U}$ is lower diagonal matrix with the minor diagonal element $\acute{U}_{i-1,i} = 2\mu\omega_1 C_E$ and 0 otherwise, the matrix $\grave{D}$ is upper diagonal matrix with the minor diagonal element $\acute{D}_{i,i-1} = 2\mu\omega_1 C_S$ and 0 otherwise.

Suppose the eigenvalue of the iterative matrix

$$A = (L + \acute{D} + \acute{U})^{-1}(E + \grave{D} + \grave{U})$$

be $\acute{\lambda}$, thus we have

$$\det(\acute{\lambda}(L + \acute{D} + \acute{U}) - (E + \grave{D} + \grave{U})) = 0.$$

According to Hadamard theorem, there exists one k, such that

$$|a_{kk}\acute{\lambda}| \leq |\acute{\lambda}| \sum_{j=1}^{k-1} |a_{kj}| + \sum_{j=k+1}^{n} |a_{kj}|. \tag{A.45}$$

That means

$$\acute{\lambda}[1 + 2\omega_1\mu(C_E + C_W + C_N + C_S) + \omega_1\nu] \leq \acute{\lambda}2\mu\omega_1(C_W + C_S) + 1 + 2\mu\omega_1(C_E + C_N),$$

which yields

$$\acute{\lambda} \leq \frac{1 + 2\mu\omega_1(C_E + C_N)}{1 + 2\omega_1\mu(C_E + C_W + C_N + C_S) + \omega_1\nu - 2\mu\omega_1(C_W + C_S)} < 1$$

because of $\omega_1\nu > 0$. That implies $\lim\limits_{k\to\infty} u^k = u^*$.

The convergence process about ψ^k follows the same argument as above.

Appendix B. **Proof of Theorem 3.6.** Let ψ^* be an exact solution of (12). By the first order optimality condition, ψ^* satisfies

$$2\mu|\nabla u^*|^2(\psi^* - 1) + \partial(TV(\psi^*)) + \tilde{\psi}^* = 0, \tag{B.1}$$

with fixed u^* and $\tilde{\psi}^* \in \partial H_\psi(\psi^*) = \partial(\frac{r}{2}\|\psi^* - q\|_2^2)$. Similarly, let u^* be an arbitrary exact solution of (11). By the first order optimality condition, u^* satisfies

$$2\mu\nabla^*((\psi^* - 1)^2\nabla u) + \tilde{u}^* = 0, \tag{B.2}$$

with fixed ψ^* and $\tilde{u}^* \in \partial H_u(u^*) = \partial(\frac{\nu}{2}\|u^* - I\|_2^2)$. By the first order optimality condition, $(u^*, \psi^*, \boldsymbol{b}^*, q^*)$ satisfy

$$\left\{\begin{array}{ll} 2\mu\nabla^*((1 - \psi^*)^2\nabla u^*) + \tilde{u}^* = 0, & \tilde{u}^* \in \partial H_u(u^*); \\ -\lambda\triangle\psi^* + 2\mu(|\nabla u^*|^2(\psi^* - 1)) + \tilde{\psi}^* - \lambda\nabla(\boldsymbol{d}^* - \boldsymbol{b}^*) = 0, & \tilde{\psi}^* \in \partial H_\psi(\psi^*); \\ \boldsymbol{p}^* + \lambda(\boldsymbol{d}^* - \nabla\psi^* - \boldsymbol{b}^*) = 0, & \boldsymbol{p}^* \in \partial|\boldsymbol{d}^*|; \\ \boldsymbol{b}^* = \boldsymbol{b}^* + (\nabla\psi^* - \boldsymbol{d}^*). & \end{array}\right. \tag{B.3}$$

The above equations (B.3) means that $(u^*, \psi^*, \boldsymbol{b}^*, q^*)$ is the fixed point of (35).

Denote the errors by

$$\begin{aligned} &u_e^n = u^n - u^*, \psi_e^n = \psi^n - \psi^*, \tilde{u}_e^n = \tilde{u}^n - \tilde{u}^*, \tilde{\psi}_e^n = \tilde{\psi}^n - \tilde{\psi}^*, \\ &\boldsymbol{p}_e^n = \boldsymbol{p}^n - \boldsymbol{p}^*, \boldsymbol{b}_e^n = \boldsymbol{b}^n - \boldsymbol{b}^*, \boldsymbol{d}_e^n = \boldsymbol{d}^n - \boldsymbol{d}^*. \end{aligned}$$

Subtracting the second equation of (35) by the second equation of (B.3), we obtain

$$-\lambda\triangle\psi_e^{n+1} + 2\mu(|\nabla u^*|^2(\psi^{n+1} - 1) - (|\nabla u^*|^2)(\psi^* - 1)) + \tilde{\psi}_e^n - \lambda\nabla^*(\boldsymbol{d}_e^n - \boldsymbol{b}_e^n) = 0.$$

Taking the inner product of the left- and right- hand sides with respect to ψ_e^{n+1}, we have

$$\lambda\|\nabla\psi_e^{n+1}\|^2 + 2\mu|\nabla u^*|^2\|\psi_e^{n+1}\|^2 + \langle\tilde{\psi}_e^n, \psi_e^{n+1}\rangle - \lambda\langle\nabla^*(\boldsymbol{d}_e^n - \boldsymbol{b}_e^n), \psi_e^{n+1}\rangle = 0. \tag{B.4}$$

Applying the same technique to the third equation of (35) and the third equation of (B.3), we obtain

$$\langle\boldsymbol{p}_e^n, \boldsymbol{d}_e^{n+1}\rangle + \lambda\|\boldsymbol{d}_e^{n+1}\|^2 - \lambda\langle\boldsymbol{d}_e^{n+1}, \nabla\psi_e^{n+1} + \boldsymbol{b}_e^n\rangle = 0. \tag{B.5}$$

Summing up (B.4) and (B.5), we get

$$\begin{aligned}&\lambda\|\nabla\psi_e^{n+1}\|^2 + 2\mu|\nabla u^*|^2\|\psi_e^{n+1}\|^2 + \langle\tilde{\psi}_e^n, \psi_e^{n+1}\rangle + \langle\boldsymbol{p}_e^n, \boldsymbol{d}_e^{n+1}\rangle\\ &+ \lambda\|\boldsymbol{d}_e^{n+1}\|^2 - \lambda\langle\nabla\psi_e^{n+1}, \boldsymbol{d}_e^{n+1} + \boldsymbol{d}_e^n\rangle + \lambda\langle\nabla\psi_e^{n+1} - \boldsymbol{d}_e^{n+1}, \boldsymbol{b}_e^n\rangle = 0.\end{aligned} \tag{B.6}$$

Furthermore, subtracting the fourth equation of (35) and the fourth equation of (B.3), we obtain

$$\boldsymbol{b}_e^{n+1} = \boldsymbol{b}_e^n + \nabla\psi_e^{n+1} - \boldsymbol{d}_e^n, \tag{B.7}$$

which leads to

$$\|\boldsymbol{b}_e^{n+1}\|^2 = \|\boldsymbol{b}_e^n\|^2 + \|\nabla\psi_e^{n+1} - \boldsymbol{d}_e^{n+1}\|^2 + 2\langle\boldsymbol{b}_e^n, \nabla\psi_e^{n+1} - \boldsymbol{d}_e^{n+1}\rangle. \tag{B.8}$$

That is,

$$\langle\boldsymbol{b}_e^n, \nabla\psi_e^{n+1} - \boldsymbol{d}_e^{n+1}\rangle = \frac{1}{2}(\|\boldsymbol{b}_e^{n+1}\|^2 - \|\boldsymbol{b}_e^n\|^2) - \frac{1}{2}\|\nabla\psi_e^{n+1} - \boldsymbol{d}_e^{n+1}\|^2. \tag{B.9}$$

Substituting (B.9) into (B.6), we have

$$\begin{aligned}\frac{\lambda}{2}(\|\boldsymbol{b}_e^n\|^2 - \|\boldsymbol{b}_e^{n+1}\|^2) &= \lambda\|\nabla\psi_e^{n+1}\|^2 + 2\mu|\nabla u^*|^2\|\psi_e^{n+1}\|^2\\ &\quad + \langle\tilde{\psi}_e^n, \psi_e^{n+1}\rangle + \langle\boldsymbol{p}_e^n, \boldsymbol{d}_e^{n+1}\rangle + \lambda\|\boldsymbol{d}_e^{n+1}\|^2 - \lambda\langle\nabla\psi_e^{n+1}, \boldsymbol{d}_e^{n+1} + \boldsymbol{d}_e^n\rangle\\ &\quad - \frac{\lambda}{2}\|\nabla\psi_e^{n+1} - \boldsymbol{d}_e^{n+1}\|^2.\end{aligned} \tag{B.10}$$

Then, by summing the above inequality bilaterally from 0 to N, we obtain

$$\begin{aligned}\frac{\lambda}{2}(\|\boldsymbol{b}_e^0\|^2 - \|\boldsymbol{b}_e^{N+1}\|^2) =& 2\mu\sum_{n=0}^{N}|\nabla u^*|^2\|\psi_e^{n+1}\|^2 + \sum_{n=0}^{N}(\langle\tilde{\psi}_e^n, \psi_e^{n+1}\rangle + \langle\boldsymbol{p}_e^n, \boldsymbol{d}_e^{n+1}\rangle)\\ &+ \frac{\lambda}{2}\Big(\sum_{n=0}^{N}\|\nabla\psi_e^{n+1} - \boldsymbol{d}_e^n\|^2 + \|\boldsymbol{d}_e^{N+1}\|^2\Big) - \frac{\lambda}{2}\|\boldsymbol{d}_e^0\|^2.\end{aligned} \tag{B.11}$$

Noting that all terms involved in (B.11) are nonnegative, and the facts $0 \leq \psi^* \leq 1$, $|\cdot|$ and $\|\cdot\|^2$ are convex, we derive that

$$\begin{aligned}\frac{\lambda}{2}\big(\|\boldsymbol{d}_e^0\|^2 + \|\boldsymbol{b}_e^0\|^2\big) \geq& 2\mu\sum_{n=0}^{N}|\nabla u^*|^2\|\psi_e^{n+1}\|^2 + \sum_{n=0}^{N}(\langle\tilde{\psi}_e^n, \psi_e^{n+1}\rangle + \langle\boldsymbol{p}_e^n, \boldsymbol{d}_e^{n+1}\rangle)\\ &+ \lambda\Big(\frac{1}{2}\sum_{n=0}^{N}\|\nabla\psi_e^{n+1} - \boldsymbol{d}_e^n\|^2 + \frac{1}{2}\|\boldsymbol{d}_e^{N+1}\|^2\Big).\end{aligned} \tag{B.12}$$

Firstly, (B.12) leads to

$$\sum_{n=0}^{N} |\nabla u^*|^2 \|\psi_e^{n+1}\|^2 < \infty, \tag{B.13}$$

which, together with Theorem 3.5, implies

$$\lim_{n\to\infty} |\nabla u^n|^2 \|\psi^n - \psi^*\|^2 = 0. \tag{B.14}$$

Denote the first term in (12) by

$$\mathcal{F}(\psi) = 2\mu \int_\Omega |\nabla u^n|(\psi - 1)^2 d\boldsymbol{x}.$$

Thus, using the nonnegativity of the Bregman distance, we have $\lim\limits_{n\to\infty} B_{\mathcal{F}(\cdot)}^{\boldsymbol{p}_\psi^*}(\psi^n, \psi^*) = 0$, i.e.,

$$\lim_{n\to\infty} \big(\mathcal{F}(\psi^n) - \mathcal{F}(\psi^*) - \langle |\nabla u^n|^2(\psi^* - 1), \psi^n - \psi^* \rangle\big) = 0. \tag{B.15}$$

Secondly, (B.12) leads to

$$\sum_{n=0}^{N} \langle \boldsymbol{p}_e^n, \boldsymbol{d}_e^{n+1}\rangle < +\infty, \quad \text{so} \quad \lim_{n\to\infty} \langle \boldsymbol{p}_e^n, \boldsymbol{d}_e^{n+1}\rangle = 0.$$

Associating it with the nonnegativity of the Bregman distance (see e.g., [46, 13]), we obtain

$$\lim_{n\to\infty} \big(|\boldsymbol{d}^n| - |\boldsymbol{d}^*| - \langle \boldsymbol{d}^n - \boldsymbol{d}^*, \boldsymbol{p}^*\rangle\big) = 0. \tag{B.16}$$

Thirdly, (B.12) leads also to $\sum_{n=0}^{N} \|\nabla \psi_e^{n+1} - \boldsymbol{d}_e^n\|^2 < \infty$, which means

$$\lim_{n\to\infty} \|\nabla \psi_e^{n+1} - \boldsymbol{d}_e^n\|^2 = 0.$$

By $\nabla \psi^* = \boldsymbol{d}^*$, we have

$$\lim_{n\to\infty} \|\nabla \psi^{n+1} - \boldsymbol{d}^n\| = 0. \tag{B.17}$$

Moreover, by the continuous property of $|\cdot|$, we obtain

$$\lim_{n\to\infty} \big(|\nabla \psi^n| - |\nabla \psi^*| - \langle \nabla \psi^n - \nabla \psi^*, \boldsymbol{p}^*\rangle\big) = 0. \tag{B.18}$$

Similarly, we have

$$\lim_{n\to\infty} \big(H_\psi(\psi^n) - H_\psi(\psi^*) - \langle \psi^n - \psi^*, \tilde{\psi}^*\rangle\big) = 0. \tag{B.19}$$

Combining (B.15), (B.18), (B.19) and (B.1), we have

$$\lim_{n\to\infty} \big(\mathcal{F}(\psi^n) + |\nabla \psi^n| + H_\psi(\psi^n)\big) = \mathcal{F}(\psi^*) + |\nabla \psi^*| + H_\psi(\psi^*). \tag{B.20}$$

This gives the first equation in (36). In the similar way as above, by denoting the first term in (11) be $\mathcal{G}(u)$, we have

$$\lim_{n\to\infty} \big(\mathcal{G}(u^n) + H_u(u^n)\big) = \mathcal{G}(u^*) + H_u(u^*), \tag{B.21}$$

where we have used the formula (B.2). This gives the second equation in (36).

Next, we prove the second equation in (37) by assuming that (11) has a unique solution. The argument is by contradiction. Assume the second equation in (37) does not hold, there exists a subsequence u^{n_i} such that $\|u^{n_i} - u^*\| > \epsilon$ for some

$\epsilon > 0$ and for all i. Let $c = tu^* + (1-t)u^{n_i}$ with $t \in (0,1)$. By the convexity of $\mathfrak{R}$ and u^* is the unique minimizer of $\mathfrak{R}(u)$, we have

$$\mathfrak{R}(u^{n_i}) > t\mathfrak{R}(u^*) + (1-t)\mathfrak{R}(u^{n_i}) \geq \mathfrak{R}(c) \geq \min\{\mathfrak{R}(u) : \|u - u^*\| = \epsilon\}.$$

Denote

$$\bar{u} = arg\min_u\{\mathfrak{R}(u) : \|u - u^*\| = \epsilon\}.$$

By applying the second equation in (37), we have

$$\mathfrak{R}(u^*) = \lim_{i\to\infty} \mathfrak{R}(u^{n_i}) \geq \mathfrak{R}(\bar{u}) > \mathfrak{R}(u^*),$$

which is a contradiction.

The first equation in (37) follows the same arguments as above.

REFERENCES

[1] L. Alvarez, P. L. Lions and J. M. Morel, Image selective smoothing and edge detection by nonlinear diffusion II, *SIAM J. Numer. Anal.*, **29** (1992), 845–866.

[2] L. Ambrosio and V. M. Tortorelli, Approximation of functions depending on jumps by elliptic functions via Γ-convergence, *Comm. Pure Appl. Math.*, **43** (1990), 999–1036.

[3] L. Ambrosio and V. M. Tortorelli, *On the Approximation of Functionals Depending on Jumps by Quadratic, Elliptic Functionals, Boll. Un. Mat. Ital.*, (1992).

[4] N. Badshah and K. Chen, Image selective segmentation under geometrical constraints using an active contour approach, *Commun. Comput. Phys.*, **7** (2010), 759–778.

[5] S. Basu, D. P. Mukherjee and S. T. Acton, Implicit evolution of open ended curves, in *IEEE International Conference on Image Processing*, Vol. 1, 2007, 261–264.

[6] B. Berkels, A. Rätz, M. Rumpf and A. Voigt, Extracting grain boundaries and macroscopic deformations from images on atomic scale, *J. Sci. Comput.*, **35** (2008), 1–23.

[7] A. Braides, *Approximation of Free-Discontinuity Problems*, Springer-Verlag, 1998.

[8] 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, *(Russian) v'Z. Vyv'cisl. Mat. i Mat. Fiz.*, **7** (1967), 620–631.

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

[10] A. Brook, R. Kimmel and N. A. Sochen, Variational restoration and edge detection for color images, *J. Math. Imaging Vis.*, **18** (2003), 247–268.

[11] E. S. Brown, T. F. Chan and X. Bresson, *A Convex Relaxation Method for a Class of Vector-Valued Minimization Problems with Applications to Mumford-Shah Segmentation*, UCLA cam report, (2010), 10–43.

[12] J. F. Cai, R. H. Chan and M. Nikolova, Two-phase approach for deblurring images corrupted by impulse plus Gaussian noise, *Inverse Probl. Imag.*, **2** (2008), 187–204.

[13] J. F. Cai, S. Osher and Z. W. Shen, Split Bregman methods and frame based image restoration, *Multiscale Model. Sim.*, **8** (2009), 337–369.

[14] X. Cai, R. Chan and T. Zeng, A two-stage image segmentation method using a convex variant of the Mumford–Shah model and thresholding, *SIAM J. Imaging Sci.*, **6** (2013), 368–390.

[15] J. Canny, A computational approach to edge detection, *IEEE T. Pattern Anal.*, **PAMI-8** (1986), 679–698.

[16] V. Caselles, R. Kimmel and G. Sapiro, Geodesic active contours, *Computer Vision, 1995. Proceedings., Fifth International Conference on*, 1995, 694–699.

[17] F. Catté, P. L. Lions, J. M. Morel and T. Coll, Image selective smoothing and edge detection by nonlinear diffusion, *SIAM J. Numer. Anal.*, **29** (1992), 182–193.

[18] A. Chambolle, Image segmentation by variational methods: Mumford and shah functional and the discrete approximations, *SIAM J. Appl. Math.*, **55** (1995), 827–863.

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

[20] R. H. Chan, M. Tao and X. M. Yuan, Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers, *SIAM J. Imaging Sci.*, **6** (2013), 680–697.

[21] T. F. Chan, S. Esedoglu and M. Nikolova, Algorithms for finding global minimizers of image segmentation and denoising models, *SIAM J. Appl. Math.*, **66** (2006), 1632–1648.

[22] G. Dal Maso, *Introduction to Γ-Convergence*, Birkhäuser, 1993.

[23] R. Deriche, Using Canny's criteria to derive a recursively implemented optimal edge detector, *Int. J. Comput. Vis.*, **1** (1987), 167–187.

[24] R. T. Farouki and C. A. Neff, Analytic properties of plane offset curves, *Computer Aided Geometric Design*, **7** (1990), 83–99.

[25] A. Giacomini, Ambrosio-Tortorelli approximation of quasi-static evolution of brittle fractures, *Calc. Var. Partial Differ. Equ.*, **22** (2005), 129–172.

[26] T. Goldstein, X. Bresson and S. Osher, Geometric applications of the split Bregman method: segmentation and surface reconstruction, *J. Sci. Comput.*, **45** (2010), 272–293.

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

[28] W. Guo and M. J. Lai, Box spline wavelet frames for image edge analysis, *SIAM J. Imaging Sci.*, **6** (2013), 1553–1578.

[29] Y. Huang, D. Lu and T. Zeng, Two-step approach for the restoration of images corrupted by multiplicative noise, *SIAM J. Sci. Comput.*, **35** (2013), A2856–A2873.

[30] M. Kass, A. P. Witkin and D. Terzopoulos, Snakes: Active contour models, *Int. J. Comput. Vis.*, **1** (1988), 321–331.

[31] R. Lai and T. F. Chan, A framework for intrinsic image processing on surfaces, *Comput. Vis. Image Underst.*, **115** (2011), 1647–1661.

[32] S. Lee, H. Lee, P. Abbeel and A. Y. Ng, Efficient l_1 regularized logistic regression, in *Proceedings of the National Conference on Artificial Intelligence*, Vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, 401.

[33] S. Leung and H. Zhao, A grid based particle method for evolution of open curves and surfaces, *J. Comput. Phys.*, **228** (2009), 7706–7728.

[34] B. Llanas and S. Lantarón, Edge detection by adaptive splitting, *J. Sci. Comput.*, **46** (2011), 485–518.

[35] W. Y. Ma and B. S. Manjunath, Edgeflow: A technique for boundary detection and image segmentation, *IEEE T. Image Process.*, **9** (2000), 1375–1388.

[36] R. March and M. Dozio, A variational method for the recovery of smooth boundaries, *Image Vis. Comput.*, **15** (1997), 705–712.

[37] E. Meinhardt, E. Zacur, A. F. Frangi and V. Caselles, 3D edge detection by selection of level surface patches, *J. Math. Imaging Vis.*, **34** (2009), 1–16.

[38] C. A. Micchelli, L. X. Shen and Y. S. Xu, Proximity algorithms for image models: Denoising, *Inverse Probl. Imag.*, **27** (2011), 045009, 30pp.

[39] C. A. Micchelli, L. X. Shen, Y. S. Xu and X. Y. Zeng, Proximity algorithms for the L1/TV image denoising model, *Adv. Comput. Math.*, **38** (2013), 401–426.

[40] R. Mohieddine and L. A. Vese, An open level set framework for image segmentation and restoration using the Mumford and Shah model, in *Proc. SPIE 7873*, Computational Imaging IX, 787309, 2011, p787309.

[41] J. J. Moreau, Fonctions convexes duales et points proximaux dans un espace hilbertien, *C.R. Acad. Sci. Paris Sér. A Math.*, **255** (1962), 2897–2899.

[42] J. J. Moreau, Proximité et dualité dans un espace hilbertien, *Bull. Soc. Math. France*, **93** (1965), 273–299.

[43] D. Mumford and J. Shah, Optimal approximations by piecewise smooth functions and associated variational problems, *Comm. Pure Appl. Math*, **42** (1989), 577–685.

[44] M. Nikolova and M. Ng, Fast image reconstruction algorithms combining half-quadratic regularization and preconditioning, in *Proceedings of 2001 International Conference on Image Processing*, Vol. 1, IEEE, 2001, 277–280.

[45] M. Nikolova and M. Ng, Analysis of half-quadratic minimization methods for signal and image recovery, *SIAM J. Sci. Comput.*, **27** (2005), 937–966.

[46] S. Osher, M. Burger, D. Goldfarb, J. Xu and W. Yin, An iterative regularization method for total variation-based image restoration, *Multiscale Modeling & Simulation*, **4** (2005), 460–489.

[47] S. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations, *J. Comput. Phys.*, **79** (1988), 12–49.

[48] P. Perona and J. Malik, Scale-space and edge-detection using anisotropic diffusion, *IEEE T. Pattern Anal.*, **12** (1990), 629–639.

[49] T. Pock, D. Cremers, H. Bischof and A. Chambolle, An algorithm for minimizing the Mumford-Shah functional, in *12th International Conference on Computer Vision*, IEEE, 2009, 1133–1140.

[50] R. Ramlau and W. Ring, A Mumford–Shah level-set approach for the inversion and segmentation of X-ray tomography data, *J. Comput. Phys.*, **221** (2007), 539–557.

[51] L. I. Rudin, S. Osher and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D: Nonlinear Phenomena*, **60** (1992), 259–268.

[52] G. Sapiro and D. L. Ringach, Anisotropic diffusion of multivalued images with applications to color filtering, *IEEE T. Image Process.*, **5** (1996), 1582–1586.

[53] H. Schaeffer and L. Vese, Active contours with free endpoints, *J. Math. Imaging Vis.*, **49** (2014), 20–36.

[54] S. Setzer, Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage, in *Scale Space and Variational Methods in Computer Vision*, Vol. 5567, Springer, Berlin, 2009, 464–476.

[55] J. Shah, A common framework for curve evolution, segmentation and anisotropic diffusion, in *IEEE Conference on Computer Vision and Pattern Recognition*, 1996, 136–142.

[56] J. Shen, A stochastic-variational model for soft Mumford-Shah segmentation, *Int. J. Biome.*, **2006** (2006), 2–16.

[57] Y. Y. Shi, L. L. Wang and X. C. Tai, Geometry of total variation regularized L^p-model, *J. Comput. Appl. Math.*, **236** (2012), 2223–2234.

[58] P. Smereka, Spiral crystal growth, *Physica D: Nonlinear Phenomena*, **138** (2000), 282–301.

[59] S. M. Smith, *Edge Thinning Used in the SUSAN Edge Detector*, Technical Report, TR95SMS5, 1995.

[60] Y. Sun, P. Wu, G. W. Wei and G. Wang, Evolution-operator-based single-step method for image processing, *Int. J. Biomed. Imaging*, **2006** (2006), 1–28.

[61] Y. Suzuki, T. Takayama, I. N. Motoike and T. Asai, A reaction-diffusion model performing stripe-and spot-image restoration and its LSI implementation, *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, **90** (2007), 20–29.

[62] X. C. Tai and C. Wu, Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model, in *Scale Space and Variational Methods in Computer Vision, Proceedings*, Lecture Notes in Computer Science, Elsevier, 5567, Springer, Berlin-Heidelberg, 2009, 502–513.

[63] B. Tang, G. Sapiro and V. Caselles, Color image enhancement via chromaticity diffusion, *IEEE T. Image Process.*, **10** (2001), 701–707.

[64] W. Tao, F. Chang, L. Liu, H. Jin and T. Wang, Interactively multiphase image segmentation based on variational formulation and graph cuts, *Pattern Recogn.*, **43** (2010), 3208–3218.

[65] Z. Tari, J. Shah and H. Pien, Extraction of shape skeletons from grayscale images, *Computer Vis. Image Und.*, **66** (1997), 133–146.

[66] V. Toponogov, *Differential Geometry of Curves and Surfaces: A Concise Guide*, Birkhäuser, 2006.

[67] M. Upmanyu, R. W. Smith and D. J. Srolovitz, Atomistic simulation of curvature driven grain boundary migration, *Interface Sci.*, **6** (1998), 41–58.

[68] L. L. Wang, Y. Y. Shi and X. C. Tai, Robust edge detection using Mumford-Shah model and binary level set method, in *Scale Space and Variational Methods in Computer Vision*, Lecture Notes in Computer Science, 6667, Springer, Berlin-Heidelberg, 2012, 291–301.

[69] Y. L. Wang, J. F. Yang, W. T. Yin and Y. Zhang, A new alternating minimization algorithm for total variation image reconstruction, *SIAM J. Imaging Sci.*, **1** (2008), 248–272.

[70] G. W. Wei and Y. Q. Jia, Synchronization-based image edge detection, *EPL (Europhysics Letters)*, **59** (2002), 814–819.

[71] C. Wu and X. C. Tai, Augmented Lagrangian method, dual methods, and split Bregman iteration for ROF, vectorial TV, and high order models, *SIAM J. Imaging Sci.*, **3** (2010), 300–339.

Received April 2013; revised April 2014.

E-mail address: yyshi@amss.ac.cn
E-mail address: guy@i2r.a-star.edu.sg
E-mail address: LiLian@ntu.edu.sg
E-mail address: tai@math.uib.no