Hindawi
Discrete Dynamics in Nature and Society
Volume 2021, Article ID 3289477, 19 pages
https://doi.org/10.1155/2021/3289477

Research Article

A LogTVSCAD Nonconvex Regularization Model for Image Deblurring in the Presence of Impulse Noise

Zhijun Luo,[1,2] **Zhibin Zhu**,[3] **and Benxin Zhang**[1]

[1]*School of Electronic Engineering and Automation, Guangxi Key Laboratory of Automatic Detecting Technology and Instruments, Guilin University of Electronic Technology, Guilin 541004, China*
[2]*School of Mathematics and Finance, Hunan University of Humanities, Science and Technology, Loudi 417000, China*
[3]*School of Mathematics and Computing Science, Guangxi Colleges and Universities Key Laboratory of Data Analysis and Computation, Guilin University of Electronic Technology, Guilin 541004, China*

Correspondence should be addressed to Zhibin Zhu; optimization_zhu@163.com

Received 11 June 2021; Accepted 5 October 2021; Published 26 October 2021

Academic Editor: Manuel De la Sen

This paper proposes a nonconvex model (called LogTVSCAD) for deblurring images with impulsive noises, using the log-function penalty as the regularizer and adopting the smoothly clipped absolute deviation (SCAD) function as the data-fitting term. The proposed nonconvex model can effectively overcome the poor performance of the classical TVL1 model for high-level impulsive noise. A difference of convex functions algorithm (DCA) is proposed to solve the nonconvex model. For the model subproblem, we consider the alternating direction method of multipliers (ADMM) algorithm to solve it. The global convergence is discussed based on Kurdyka–Lojasiewicz. Experimental results show the advantages of the proposed nonconvex model over existing models.

1. Introduction

Image deblurring is a hot research topic of digital image processing, which is widely used in engineering and medicine fields [1]. In this paper, we focus on how to recover an image degraded by blur and impulsive noise. In this paper, we focus on how to recover an image degraded by blur and impulsive noise. Image blur noise may result from inaccurate focus, object relative movement, and optical degradation in the process of digital image acquisition and transmission. Impulse noise, such as salt-and-pepper noise (SP) and random-value noise (RV), is caused in the storage and transmission process dues to low-quality sensors or electromagnetic interference [2]. The mathematical model of image deblurring is usually expressed as

$$f = \mathcal{N}_{\text{imp}}(\mathbf{Ku}), \tag{1}$$

where f denotes the observed noisy and blurry image, $\mathcal{N}_{\text{imp}}(\cdot)$ represents the formation mechanism of the impulsive noise, and $\mathbf{K}$ and $\mathbf{u}$ denote a bounded blurring operator and the original image, respectively. For given the blurring operator $\mathbf{K}$, our goal is to recover the original image $\mathbf{u}$ from the observation f. In general, the operator matrix $\mathbf{K}$ is often ill-conditioned, which cannot recover the original image $\mathbf{u}$ from f by direct inversion. To stabilize the recovery of $\mathbf{u}$, one popular approach is the variational method, which includes a data fitting term and a regularization term. Rudin, Osher and Fatemi [3] first proposed total variation (TV) regularization model, which is widely used, for it can better keep the object boundaries information of the signal [4, 5]. The popular model is

$$\min_{\mathbf{u}} \|\mathbf{u}\|_{\text{TV}} + \mu\|\mathbf{Ku} - \mathbf{f}\|_2^2, \tag{2}$$

where $\mathbf{u}$ and $\mathbf{f}$ are the original image and the observed image, respectively; $\mathbf{K}$ denotes the linear blurring; $\mu > 0$ is the regularization parameter used to balance the regularization term and data-fitting term. This TVL2 model is optimal

when the measurement noise is Gaussian distributed. However, non-Gaussian noise is more common in practice, and the performance of ℓ_2-norm based methods may severely degrade. TVL1 model combining TV regularization and ℓ_1-norm [6–9] was proposed to deal with impulsive noise (a typical non-Gaussian noise). Its mathematical formulation can be expressed as follows:

$$\min_{\mathbf{u}} \|\mathbf{u}\|_{\mathrm{TV}} + \mu\|\mathbf{Ku} - \mathbf{f}\|_1, \tag{3}$$

where $\|\mathbf{u}\|_{\mathrm{TV}}$ is called the TV norm of the variable $\mathbf{u}$, which can take the ℓ_1-norm and ℓ_2-norm, i.e.,

$$\|\mathbf{u}\|_{\mathrm{TV}} \coloneqq \|\mathbf{Du}\|_1 \text{ (anisotropic) and } \|\mathbf{u}\|_{\mathrm{TV}} \coloneqq \|\mathbf{Du}\|_2 \text{ (isotropic)}, \tag{4}$$

where $\mathbf{D}$ is a finite difference operator. Generally, the TVL1 model with anisotropic TV norm can express a linear system, which is easier to deal with than the isotropic one. However, the isotropic TV norm is more realistic and more effective [10]. Numerically, some efficient algorithms have been proposed to solve the TVL1 model (3), such as the split Bregman [11, 12], the primal-dual method [7, 8, 13], and the alternating direction method of multipliers [9, 14, 15].

However, the classical TV norm regularization model often underestimates the amplitudes of signal discontinuities [16, 17]. For high-level impulsive noise, the solution of TVL1 model (3) is biased, because the penalties of data fitting term for all data are equal [15]. In order to improve the performance of image restoration, nonconvex approaches are considered, e.g., the smoothly clipped absolute deviation (SCAD) [18], ℓ_q-norm $(0 < q < 1)$ [19, 20], log-function [21, 22], and minimax-concave penalty (MCP) [23, 24]. This nonconvex technique plays an increasingly important role in solving image restoration problems. Because the nonconvex model can obtain a better approximation solution, it can improve the bias problem of the ℓ_1-norm [25–28]. In [26], the authors have developed a nonconvex model called TVSCAD with the SCAD penalty function as data fitting term. In this model, they suggested that if the observation data is not severely damaged, data fitting should be enforced; otherwise, less or null penalize those data. Based on this work, the authors [27] very recently proposed a TV-Log model via using the TV as regularizer and Log penalty function as data fitting.

In this paper, we continue to study the problem of image restoration with impulse noise. Our goal is to obtain a higher quality recovery solution through the newly constructed nonconvex model. Using the Log-function penalty as a nonconvex regularizer and the SACD-function penalty as a data fitting term, a nonconvex model is proposed:

$$\min_{\mathbf{u}} F_{\log}(\mathbf{Du}) + \mu F_{\mathrm{scad}}(\mathbf{Ku} - \mathbf{f}), \quad 0 \le \mathbf{u} \le 1, \tag{5}$$

where $\mu > 0$ is the parameter. Note that here we add the bound constraint $0 \le \mathbf{u} \le 1$, which can improve the image recovery quality [26]. $F_{\log}(\cdot)$ and $F_{\mathrm{scad}}(\cdot)$ are defined as

$$\begin{aligned} F_{\log}(v) &= \sum f_s(v_i), \\ F_{\mathrm{scad}}(v) &= \sum f_\gamma(v_i), \end{aligned} \tag{6}$$

$$f_s(x) = \frac{1}{s}\log(1 + s|x|),$$

$$f_\gamma(x) = \begin{cases} |x|, & |x| < \gamma_1, \\ \dfrac{2\gamma_2|x| - x^2 - \gamma_1^2}{2(\gamma_2 - \gamma_1)}, & \gamma_1 \le |x| < \gamma_2, \\ \dfrac{\gamma_1 + \gamma_2}{2}, & |x| \ge \gamma_2, \end{cases} \tag{7}$$

where $s > 0, \gamma \coloneqq (\gamma_1, \gamma_2) > 0$ are the threshold parameters. This is a "nonconvex + nonconvex" model, which can have some desirable properties simultaneously. Such penalty of concave functions $f_s(x)$ and $f_\gamma(x)$ for all elements is nonuniform, which makes $f_s(x)$ and $f_\gamma(x)$ closer to ℓ_0-norm than ℓ_1-norm. This result can be easily seen in Figure 1. Since the proposed model is nonconvex and nonsmooth, it is difficult to find an effective algorithm. To solve the proposed nonconvex model, we combine the difference-of-convex algorithm (DCA) [29, 30] with the proximal splitting method [31, 32]. In summary, the main contributions of this article are as follows:

(1) A new "nonconvex + nonconvex" model for image restoration with impulsive noise is proposed. The core idea is that the Log-function penalty as a regularizer and the SACD-function penalty as a data fidelity term are used. Therefore, this model approximates the ℓ_0-norm more closely than ℓ_1-norm and is useful in image restoration with impulse noise.

(2) To solve the nonconvex model, we consider the DCA method with ADMM, which has been efficiently used in many nonconvex optimization problems. Then, we prove the proposed algorithm that is globally convergent.

(3) Numerical examples show the effectiveness of the proposed LogTVSCAD method, and we compare it with other recovery algorithms.

The rest of this paper is organized as follows. Section 2 gives some notations and preliminaries. The nonconvex LogTVSCAD model and a DCA method with ADMM are shown in Section 3. Then, in Section 4, we prove that the proposed algorithm converges to a stationary point. Section 5 presents the experimental results, which illustrate the effectiveness of the new nonconvex model. Finally, some conclusions are given.

2. Notations and Preliminaries

In this section, we first give some notations. Then, some properties of the Log-function and SCAD-function penalty are given. Next, we show the definitions of the subdifferentials and basic properties of the Kurdyka–Lojasiewicz functions [33].

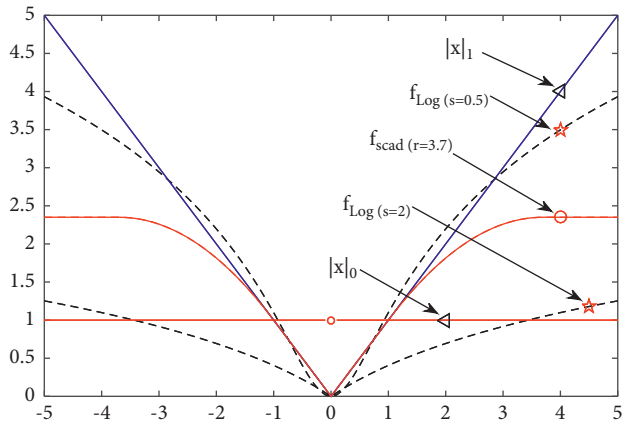


FIGURE 1: Plot of $|x|_1$, $|x|_0$, SCAD function f_{scad} and Log-function $f_{\log}$.

These conclusions will be used later in the proof of convergence.

For any vector $x \in R^n$, $x^T y$ or $\langle x, y\rangle$ $(\forall y \in R^n)$ denote their inner product; $\|x\|_q = (\sum_{i=1}|x_i|^q)^{1/q}$ $(1 > q > 0)$ denotes ℓ_q-norm; $\nabla f(x)$ and $\partial f(x)$ stand for the gradient and subdifferential of the function $f(\cdot)$ at x, respectively; sign (x) is the signum function. Now, we introduce some properties of the functions $f_s(x)$ and $f_\gamma(x)$. For fixed $s > 0$ and $\gamma > 0$, $f_s(x)$ and $f_\gamma(x)$ are continuous on R, also increasing, continuously differentiable, and concave on R_+, as illustrated in Figure 1. Next, we look at another functions $g_s(x)$ and $g_\gamma(x)$, which are induced by $f_s(x)$ and $f_\gamma(x)$:

$$\begin{aligned} g_s(x) &= |x| - f_s(x), \\ g_\gamma(x) &= |x| - f_\gamma(x), \end{aligned} \tag{8}$$

where $f_s(x)$ and $f_\gamma(x)$ are given by (7). Without loss of generality, we consider the multivariate generalization of the functions $g_s(x)$ and $g_\gamma(x)$:

$$\begin{aligned} G_s(v) &= \sum_{i=1}^{n} g_s(v_i), \\ G_\gamma(v) &= \sum_{i=1}^{n} g_\gamma(v_i), \quad v \in R^n. \end{aligned} \tag{9}$$

In fact, the functions $G_s(v)$ and $G_\gamma(v)$ are continuously differentiable and convex on R^n, and then the log function penalty and the SCAD function penalty can be expressed as

$$\begin{aligned} F_{\log}(v) &= \|v\|_1 - G_s(v), \\ F_{\text{scad}}(v) &= \|v\|_1 - G_\gamma(v). \end{aligned} \tag{10}$$

Note that the functions (10) play an important role in the later algorithm construction. The following useful definitions and properties can be obtained from the literature [34–36].

Definition 1. If $f(x_0) \le \liminf_{x \longrightarrow x_0} f(x)$, then the function $f(x)$ is lower semicontinuous at a point x_0. A function $f(x)$ is said to be lower semicontinuous in its domain of definition if it is lower semicontinuous at all $x \in \text{dom}(f) := \{x \in R^m : f(x) < +\infty\}$.

Definition 2. For an extended-real-valued, proper and lower semicontinuous function $f(x)$, its subdifferential ∂f at $x \in \text{dom}(f)$ is defined as

$$\begin{aligned} \partial f(x) := \Big\{ v \in R^m : \exists x^k \longrightarrow x, f\left(x^k\right) \\ \longrightarrow f(x), v^k \in \widehat{\partial} f\left(x^k\right) \longrightarrow v, k \longrightarrow +\infty \Big\}, \end{aligned} \tag{11}$$

where $\widehat{\partial} f(x^k) := \liminf_{z \neq x^k, z \longrightarrow x^k} (1/\|z - x^k\|)$ $(f(z) - tfn(x^k)q - h\langle v^k, z - x^k\rangle)$.

If f is convex function, then the subdifferential is such that

$$\partial f(x) := \{v \in R^m : f(y) \ge f(x) + \langle v, y - x\rangle, \forall y \in R^m\}. \tag{12}$$

Furthermore, if $f(x)$ is continuously differentiable at x_0, then $\partial f(x_0) = \nabla f(x_0)$. If the point $x^* \in R^m$ is a minimizer of $f(x)$, a necessary condition is $0 \in \partial f(x^*)$, which is named a stationary or critical point of $f(x)$.

Definition 3. Let $f(x)$ be a proper and lower semicontinuous function; the proximity operator is defined as

$$\text{prox}_{f,\eta}(t) = \arg\min_x \left\{ f(x) + \frac{\eta}{2}(x - t)^2 \right\}, \tag{13}$$

where $\eta\,(\eta > 0)$ is a penalty parameter.

It is well known that the proximal operator is particularly useful in convex optimization.

Definition 4. A function $f(x)$ is called to possess the *Kurdyka – Łojasiewicz* (**KL**) property at a point $x_0 \in \text{dom}\partial f$ if there exist $\eta > 0$, a neighborhood U of x_0 and a continuous concave function g: $[0, \eta) \longrightarrow \mathbb{R}_+$ such that

(i) $g(0) = 0$, $g(x)$ is continuously differentiable and $g'(x) > 0$ for all $x \in (0, \eta)$

(ii) $\forall x \in U$ satisfying $f(x_0) < f(x) < f(x_0) + \eta$; it holds that $g'(f(x) - f(x_0))\text{dist}(0, \partial f(x)) \ge 1$

A proper closed function $f(x)$ is named a **KL** function if it has the **KL** property at all points in $\text{dom}\partial f$.

3. Model and Algorithm

In this section, we first propose a nonconvex model for image restoration and then use DC programming to give the ADMM algorithm to solve the model.

In the definitions of (10), we consider replacing v with gradient $\mathbf{Dx}$ and $(\mathbf{Ku} - \mathbf{f})$ respectively, which leads to our definition of the LogTVSCAD model as follows:

$$\min_{\mathbf{u}} F_{\log}(\mathbf{Du}) + \mu F_{\text{scad}}(\mathbf{Ku} - \mathbf{f}), \quad 0 \le \mathbf{u} \le 1. \tag{14}$$

i.e.,

$$\min_{\mathbf{u}} \|\mathbf{u}\|_{\mathrm{TV}} - \mathbf{G_s}(\mathbf{Du}) + \mu\left(\|\mathbf{Ku}-\mathbf{f}\|_1 - \mathbf{G}_\gamma(\mathbf{Ku}-\mathbf{f})\right), \quad 0 \le \mathbf{u} \le 1. \tag{15}$$

Because Log and SCAD penalty functions are nonconvex, this model is nonconvex. To address this nonconvex model, set $\mathbf{j}(\mathbf{u}) = \|\mathbf{u}\|_{\mathrm{TV}} + \mu\|\mathbf{Ku}-\mathbf{f}\|_1$, $\mathbf{h}(\mathbf{u}) = \mathbf{G_s}(\mathbf{Du}) + \mu\mathbf{G}_\gamma(\mathbf{Ku}-\mathbf{f})$, and problem (15) can be expressed as the difference of the convex functions $\mathbf{j}(\mathbf{u})$ and $\mathbf{h}(\mathbf{u})$, i.e.,

$$\min_{\mathbf{u}} \mathbf{F}(\mathbf{u}) = \mathbf{j}(\mathbf{u}) - \mathbf{h}(\mathbf{u}), \quad 0 \le \mathbf{u} \le 1. \tag{16}$$

This is a DC programming problem, which has been efficiently used in many nonconvex optimization problems; for more details, please see [30, 37]. According to the classic DC algorithm (DCA) iteration, for (16), we have

$$\mathbf{u_{k+1}} = \min_{\mathbf{u}} \mathbf{j}(\mathbf{u}) - \mathbf{h}(\mathbf{u_k}) - \mathbf{v_k^T}(\mathbf{u}-\mathbf{u_k}), \quad 0 \le \mathbf{u} \le 1, \tag{17}$$

where $\mathbf{v_k} \in \partial\mathbf{h}(\mathbf{u_k}), \mathbf{k} = 0, 1, 2, \ldots$. To obtain a more accurate solution, we adopt the suggestion of the literature [26] and add a proximal term in our DCA iterations,

$$\mathbf{u_{k+1}} = \min_{\mathbf{u}} \mathbf{j}(\mathbf{u}) - \mathbf{h}(\mathbf{u_k}) - \mathbf{v_k^T}(\mathbf{u}-\mathbf{u_k}) + \frac{\eta}{2}\|(\mathbf{u}-\mathbf{u_k})\|^2, \quad 0 \le \mathbf{u} \le 1, \tag{18}$$

where $\eta > 0$ is a given proximal parameter. For (15), the next iteration can be expressed as

$$\mathbf{u_{k+1}} = \min_{\mathbf{u}} \|\mathbf{u}\|_{\mathrm{TV}} + \mu\|\mathbf{Ku}-\mathbf{f}\|_1 - \left[\mathbf{D^T}\nabla\mathbf{G_s}(\mathbf{Du_k}) + \mu\mathbf{K^T}\nabla\mathbf{G}_\gamma \cdot (\mathbf{Ku_k}-\mathbf{f})\right]^{\mathrm{T}}(\mathbf{u}-\mathbf{u_k}) + \frac{\eta}{2}\|(\mathbf{u}-\mathbf{u_k})\|^2, \quad 0 \le \mathbf{u} \le 1. \tag{19}$$

By omitting the constants of formula (19), we have

$$\mathbf{u_{k+1}} = \min_{\mathbf{u}} \|\mathbf{u}\|_{\mathrm{TV}} + \mu\|\mathbf{Ku}-\mathbf{f}\|_1 - \nabla\mathbf{G_s}(\mathbf{Du_k})^{\mathrm{T}}(\mathbf{Du}) - \mu\nabla\mathbf{G}_\gamma(\mathbf{Ku_k}-\mathbf{f})^{\mathrm{T}}(\mathbf{Ku}-\mathbf{f}) + \frac{\eta}{2}\|(\mathbf{u}-\mathbf{u_k})\|^2, \quad 0 \le \mathbf{u} \le 1. \tag{20}$$

For the above model (20), under certain conditions, its objective function is strongly concave, and $\mathbf{u_{k+1}}$ has a unique solution in every step. To solve the problem of (20), we first introduce auxiliary variables $\mathbf{w}, \mathbf{z}, \mathbf{p}, \mathbf{q}, \mathbf{x}$ and define

$$\begin{aligned} \mathbf{w} &= \mathbf{Du}, \\ \mathbf{p_k} &= \nabla\mathbf{G}_\gamma(\mathbf{Kuk}-\mathbf{f}), \\ \mathbf{x} &= \mathbf{uz} = \mathbf{Ku}-\mathbf{f}, \\ \mathbf{q_k} &= \nabla\mathbf{G_s}(\mathbf{Du_k}). \end{aligned} \tag{21}$$

Then, we rewrite (20) as a constrained minimization problem:

$$\begin{cases} \min\limits_{\mathbf{u},\mathbf{w},\mathbf{z},\mathbf{x}} \|\mathbf{w}\|_2 + \mu\|\mathbf{z}\|_1 - \mu\mathbf{p_k^T}\mathbf{z} - \mathbf{q_k^T}\mathbf{w} + \dfrac{\eta}{2}\|(\mathbf{u}-\mathbf{u_k})\|^2 \\ \text{s.t. } \mathbf{w} = \mathbf{Du}, \\ \mathbf{z} = \mathbf{Ku}-\mathbf{f}, \\ \mathbf{x} = \mathbf{u}, \mathbf{x} \in \Omega = \{\mathbf{u} | 0 \le \mathbf{u} \le 1\}. \end{cases} \tag{22}$$

Let the augmented Lagrangian function of model (22) be

$$\begin{aligned} \mathscr{L}_\beta(\mathbf{u},\mathbf{w},\mathbf{z},\mathbf{x},\lambda_w,\lambda_z,\lambda_x) &= \|\mathbf{w}\|_2 - \lambda_w^T(\mathbf{w}-\mathbf{Du}) + \frac{\beta_w}{2}\|\mathbf{w}-\mathbf{Du}\|^2 - \mathbf{q_k^T}\mathbf{w} \\ &+ \mu\|\mathbf{z}\|_1 - \mu\mathbf{p_k^T}\mathbf{z} - \lambda_z^T(\mathbf{z}-(\mathbf{Ku}-\mathbf{f})) + \frac{\beta_z}{2}\|\mathbf{z}-(\mathbf{Ku}-\mathbf{f})\|^2 \\ &+ \frac{\eta}{2}\|(\mathbf{u}-\mathbf{u_k})\|^2 - \lambda_x^T(\mathbf{x}-\mathbf{u}) + \frac{\beta_w}{2}\|\mathbf{x}-\mathbf{u}\|^2, \end{aligned} \tag{23}$$

where λ_w, λ_z, and λ_x are Lagrange multipliers and $\beta_w > 0, \beta_z > 0$, and $\beta_x > 0$ are penalty parameters. Given $(\lambda_w, \lambda_z, \lambda_x) := (\lambda_w^0, \lambda_z^0, \lambda_x^0)$ and $\mathbf{u} := \mathbf{u}_0$, according to the classical ADMM, the iterative scheme of the problem (23) can be expressed as follows:

$$\begin{cases} \left(\mathbf{w}^{\mathbf{i+1}}, \mathbf{z}^{\mathbf{i+1}}, \mathbf{x}^{\mathbf{i+1}}\right) = \operatorname{argmin}\left\{\mathscr{L}_\beta\left(\mathbf{w}, \mathbf{z}, \mathbf{x}, \mathbf{u_k}, \lambda_w^i, \lambda_z^i, \lambda_x^i\right) | \mathbf{w}, \mathbf{z}, \mathbf{x} \in \Omega\right\}, \\ \mathbf{u}^{\mathbf{i+1}} = \operatorname{argmin}\left\{\mathscr{L}_\beta\left(\mathbf{w}^{\mathbf{i+1}}, \mathbf{z}^{\mathbf{i+1}}, \mathbf{x}^{\mathbf{i+1}}, \mathbf{u}, \lambda_w^i, \lambda_z^i, \lambda_x^i\right)\right\}, \\ \lambda_w^{i+1} = \lambda_w^i - \beta_w\left(\mathbf{w}^{\mathbf{i+1}} - \mathbf{D}\mathbf{u}^{\mathbf{i+1}}\right), \\ \lambda_z^{i+1} = \lambda_z^i - \beta_z\left(\mathbf{z}^{\mathbf{i+1}} - \left(\mathbf{K}\mathbf{u}^{\mathbf{i+1}} - \mathbf{f}\right)\right), \\ \lambda_x^{i+1} = \lambda_x^i - \beta_x\left(\mathbf{x}^{\mathbf{i+1}} - \mathbf{u}^{\mathbf{i+1}}\right). \end{cases} \tag{24}$$

In this scheme, the ADMM method is directly applied to 2−blocks of variables $(\mathbf{w}, \mathbf{z}, \mathbf{x})$ and $\mathbf{u}$. Furthermore, via Theorem 3.2 in [9], we can ensure that the proposed ADMM (24) for solving the subproblem is convergent. In fact, $\mathbf{w}, \mathbf{z}$, and $\mathbf{x}$ in (24) are separable from each other, so this optimization problem can be performed in parallel. Moreover, via the definition of the proximity operator, we can get the explicit solution of $\mathbf{w}^{\mathbf{i+1}}$ and $\mathbf{z}^{\mathbf{i+1}}$. In addition, $\mathbf{x}^{\mathbf{i+1}}$ can be computed by a simple projection onto box Ω. Hence, the $\mathbf{w}, \mathbf{z}, \mathbf{x}$ optimizations have a closed-form solution as

$$\begin{aligned} w^{i+1} &= \max\left\{\left\|\widehat{w}^i\right\|_2 - \frac{1}{\beta_w}, 0\right\}\frac{\widehat{w}^i}{\left\|\widehat{w}^i\right\|_2}, \\ \text{where, } \widehat{w}^i &= Du^i + \frac{\lambda_w^i + q^i}{\beta_w}, \end{aligned} \tag{25}$$

$$\begin{aligned} z^{i+1} &= \operatorname{sign}\left\{\widehat{z}^i\right\}\max\left\{\left|\widehat{z}^i\right| - \frac{\mu}{\beta_z}, 0\right\}, \\ \text{where, } \widehat{z}^i &= Ku^i - f + \frac{\lambda_z^i + \mu p^i}{\beta_z}, \end{aligned} \tag{26}$$

$$x^{i+1} = \max\left\{\min\left(u^i + \frac{\lambda_x^i}{\beta_x}, 1\right), 0\right\}, \tag{27}$$

where $\operatorname{sign}(\cdot)$ is the signum function. Then, we consider how to solve the $\mathbf{u}$-subproblem of (24). Via the first-order optimality conditions, the corresponding normal equation is

$$\begin{aligned} \widehat{\mathbf{U}}^i\mathbf{u}^{i+1} = {} & \mathbf{D}^{\mathbf{T}}\left(\mathbf{w}^{i+1} - \frac{\lambda_w^i}{\beta_w}\right) + \frac{\beta_z}{\beta_w}\mathbf{K}^{\mathbf{T}}\left(\mathbf{z}^{i+1} - \frac{\lambda_z^i}{\beta_z}\right) \\ & + \frac{\beta_z}{\beta_w}\mathbf{K}^{\mathbf{T}}\mathbf{f} + \frac{\beta_x}{\beta_w}\left(\mathbf{x}^{i+1} - \frac{\lambda_x^i}{\beta_x}\right) + \frac{\eta}{\beta_w}\mathbf{u}^i. \end{aligned} \tag{28}$$

where $\widehat{\mathbf{U}}^i = \mathbf{D}^{\mathbf{T}}\mathbf{D} + (\beta_z/\beta_w)\mathbf{K}^{\mathbf{T}}\mathbf{K} + ((\beta_x + \eta)/(\beta_w))\mathbf{I}$ is nonsingular under certain conditions. For problem (28), there is an efficient solution by an inverse fast Fourier transforms [14].

Finally, we propose an ADMM algorithm for solving the proposed LogTVSCAD model (5).

3.1. Algorithm (LogTVSCAD)

Step 0 Initialization and date:

Input parameters $\mu, s, \gamma_1, \gamma_2, \beta_w, \beta_z, \beta_x, \alpha, \eta > 0$, the tolerance $\varepsilon > 0$. Given $\mathbf{u}_0$, let $k := 0$;

Step 1 Given $\lambda_w^0, \lambda_z^0, \lambda_x^0$ and $\mathbf{u}_k$, compute the new iterate by (29);

For $i := 0$;

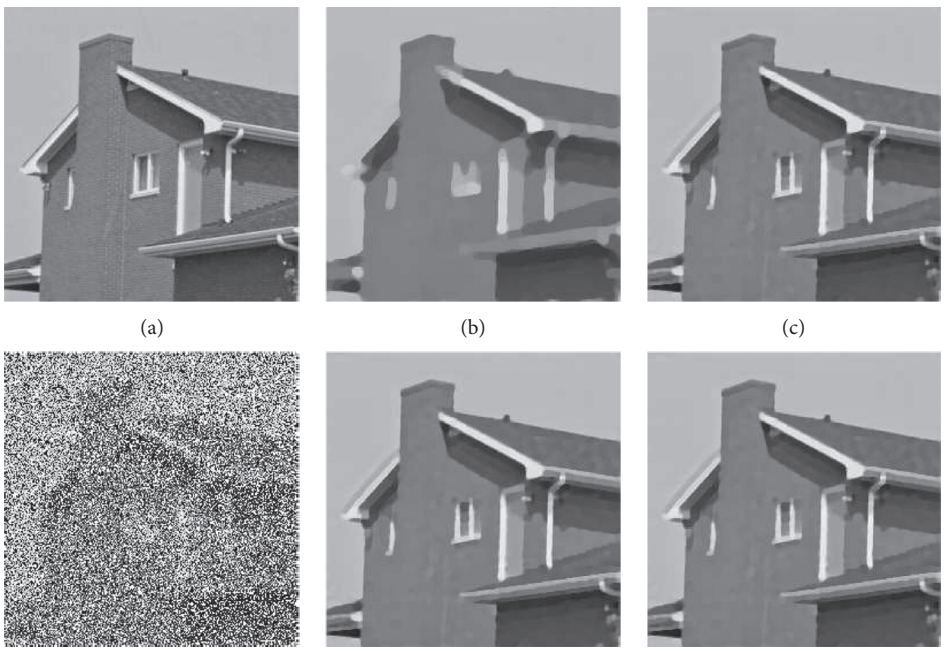
$$\begin{cases} \mathbf{w}^{i+1} = \max\left\{\left\|\widehat{\mathbf{w}}^i\right\|_2 - \frac{1}{\beta_w}, 0\right\}\frac{\widehat{\mathbf{w}}^i}{\left\|\widehat{\mathbf{w}}^i\right\|_2}, \\ \mathbf{z}^{i+1} = \operatorname{sign}\left\{\widehat{\mathbf{z}}^i\right\}\max\left\{\left|\widehat{\mathbf{z}}^i\right| - \frac{\mu}{\beta_z}, 0\right\}, \\ \mathbf{x}^{i+1} = \max\left\{\min\left(\mathbf{u}_k + \frac{\lambda_x^i}{\beta_x}, 1\right), 0\right\}, \\ \mathbf{u}^{i+1} = \widehat{\mathbf{U}}^{\mathbf{i}^{-1}}\widetilde{\mathbf{U}}^i, \\ \lambda_w^{i+1} = \lambda_w^i - \beta_w\left(\mathbf{w}^{\mathbf{i+1}} - \mathbf{D}\mathbf{u}^{\mathbf{i+1}}\right), \\ \lambda_z^{i+1} = \lambda_z^i - \beta_z\left(\mathbf{z}^{\mathbf{i+1}} - \left(\mathbf{K}\mathbf{u}^{\mathbf{i+1}} - \mathbf{f}\right)\right), \\ \lambda_x^{i+1} = \lambda_x^i - \beta_x\left(\mathbf{x}_{\mathbf{k+1}} - \mathbf{u}^{\mathbf{i+1}}\right). \end{cases} \tag{29}$$

where $\widetilde{\mathbf{U}}^i = [\mathbf{D}^{\mathbf{T}}(\mathbf{w}^{i+1} - (\lambda_w^i/\beta_w)) + (\beta_z/\beta_w)\mathbf{K}^{\mathbf{T}}(\mathbf{z}^{i+1} - (\lambda_z^i/\beta_z)) + (\beta_z/\beta_w)\mathbf{K}^{\mathbf{T}}\mathbf{f} + (\beta_x/\beta_w)(\mathbf{x}^{i+1} - (\lambda_x^i/\beta_x)) + (\eta/\beta_w)\mathbf{u}^i]$, $\widehat{\mathbf{w}}^i, \widehat{\mathbf{z}}^i$, and $\widehat{\mathbf{U}}^{\mathbf{i}}$ are given by (25), (26) and (28), respectively. If $(\|\mathbf{u}^i - \mathbf{u}^{i+1}\|^2)/(1 + \|\mathbf{u}^i\|) \le \varepsilon$, break,

end

Step 2 Set $\mathbf{u}_{k+1} = \mathbf{u}^{i+1}$, if $\|\mathbf{u}_k - \mathbf{u}_{k+1}\|^2 \le \varepsilon$, STOP; otherwise let $k = : k + 1$. Go back to step 1.

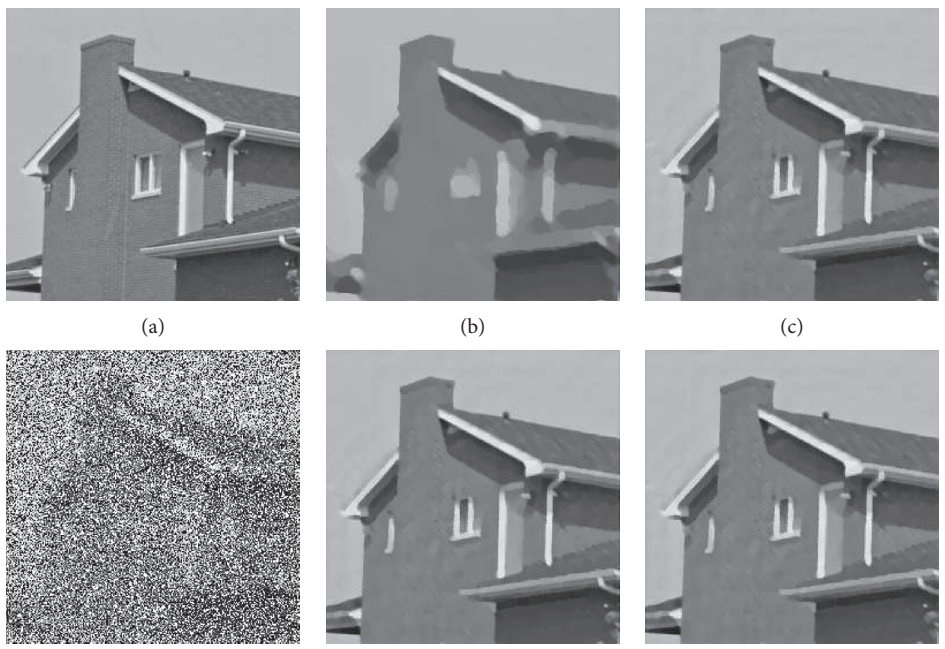
Note: the algorithm is composed of inner and outer loops. The inner loop uses the ADMM algorithm to solve the subproblems, and the outer loop uses the DC programming framework to solve the nonconvex model.



(a) (b) (c)

(d) (e) (f)

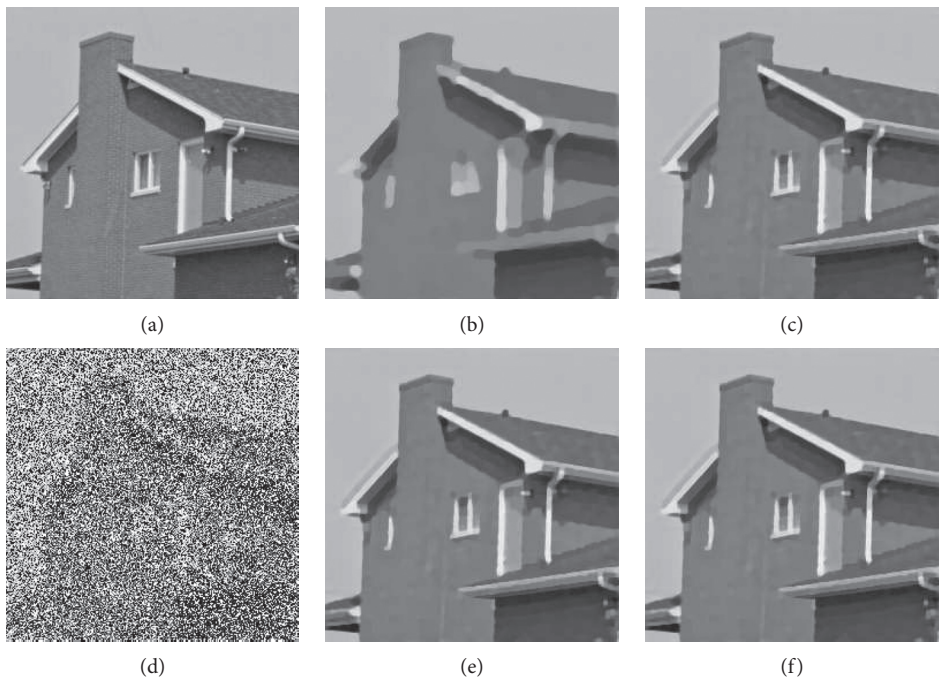
Figure 2: The recovery results (House 256) by Gaussian blur with 60% SP noise. (a) Original. (b) TVL1, PSNR = 24.6266. (c) TVSCAD, PSNR = 30.3598. (d) Gaussian blur (lev = 0.6). (e) TVLog, PSNR = 30.3535. (f) LogTVSCAD, PSNR = 30.3635.



(a) (b) (c)

(d) (e) (f)

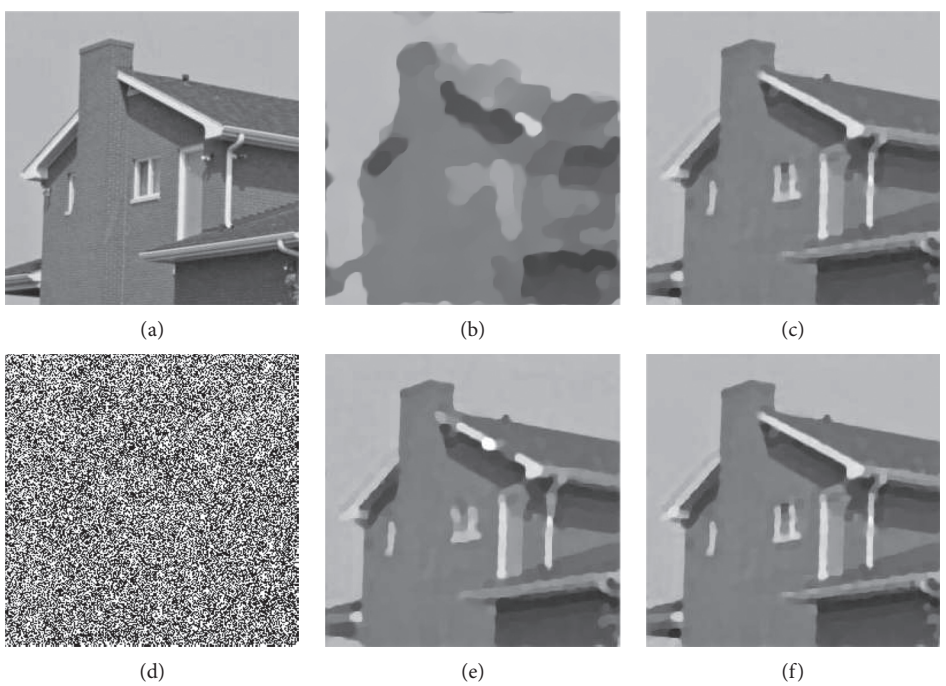
Figure 3: The recovery results (House 256) by Motion blur with 60% SP noise. (a) Original. (b) TVL1, PSNR = 23.739. (c) TVSCAD, PSNR = 29.7605. (d) Motion blur (lev = 0.6). (e) TVLog, PSNR = 29.7593. (f) LogTVSCAD, PSNR = 29.7678.



(a) (b) (c)

(d) (e) (f)

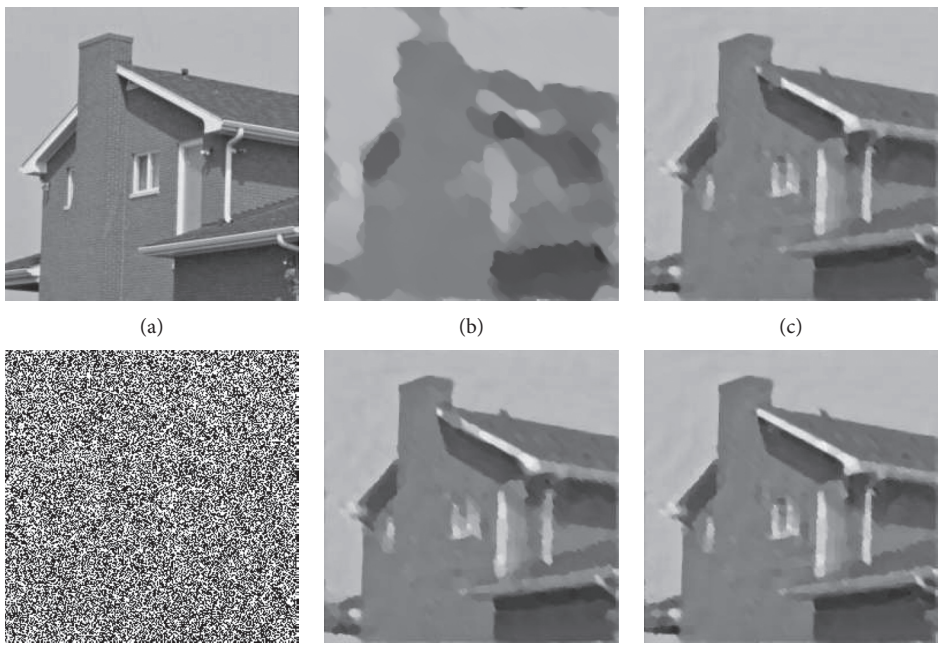
Figure 4: The recovery results (House 256) by Average blur with 60% SP noise. (a) Original. (b) TVL1, PSNR = 24.5554. (c) TVSCAD, PSNR = 30.4234. (d) Average blur (lev = 0.6). (e) TVLog, PSNR = 30.3966. (f) LogTVSCAD, PSNR = 30.4302.



(a) (b) (c)

(d) (e) (f)

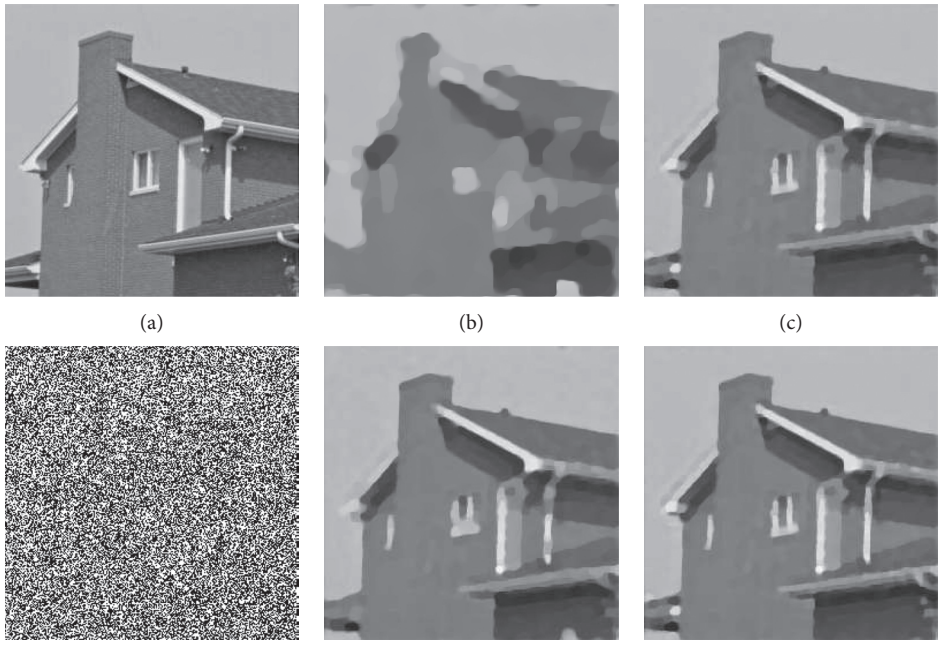
Figure 5: The recovery results (House 256) by Gaussian blur with 90% SP noise. (a) Original. (b) TVL1, PSNR = 19.0225. (c) TVSCAD, PSNR = 27.0674. (d) Gaussian blur (lev = 0.9). (e) TVLog, PSNR = 24.362. (f) LogTVSCAD, PSNR = 27.2301.



(a) (b) (c)

(d) (e) (f)

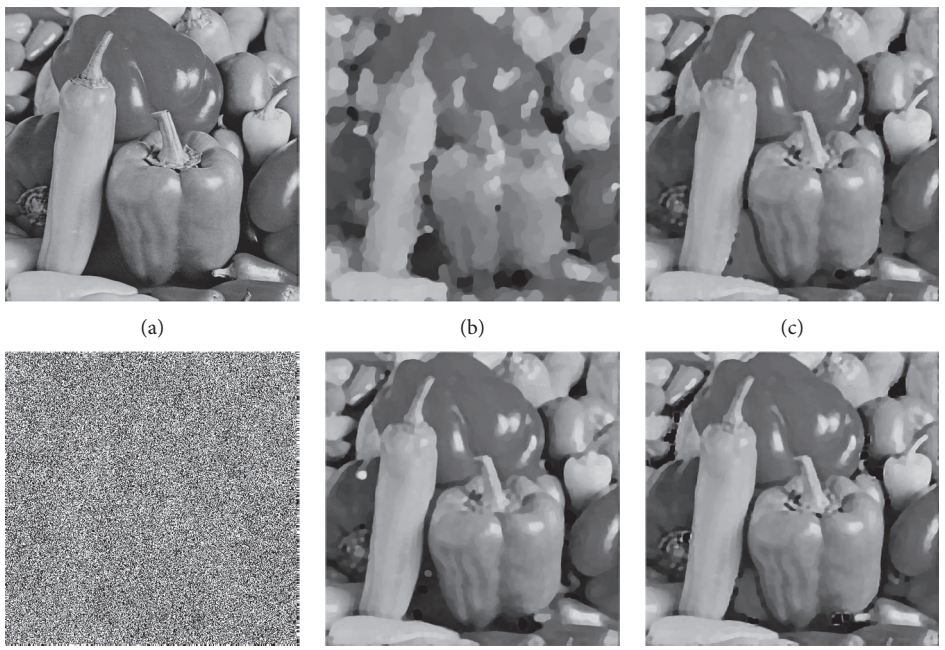
Figure 6: The recovery results (House 256) by Motion blur with 90% SP noise. (a) Original. (b) TVL1, PSNR = 18.7106. (c) TVSCAD, PSNR = 25.0094. (d) Motion blur (lev = 0.9). (e) TVLog, PSNR = 24.2055. (f) LogTVSCAD, PSNR = 25.8013.



(a) (b) (c)

(d) (e) (f)

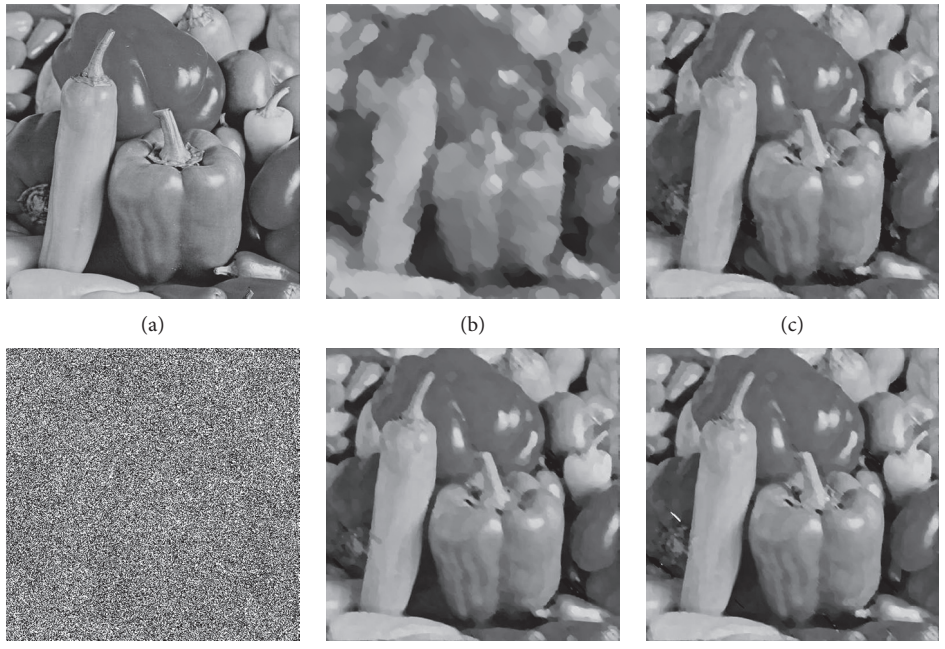
Figure 7: The recovery results (House 256) by Average blur with 90% SP noise. (a) Original. (b) TVL1, PSNR = 18.9538. (c) TVSCAD, PSNR = 26.9624. (d) Average blur (lev = 0.9). (e) TVLog, PSNR = 25.0969. (f) LogTVSCAD, PSNR = 27.1779.



(a) (b) (c)

(d) (e) (f)

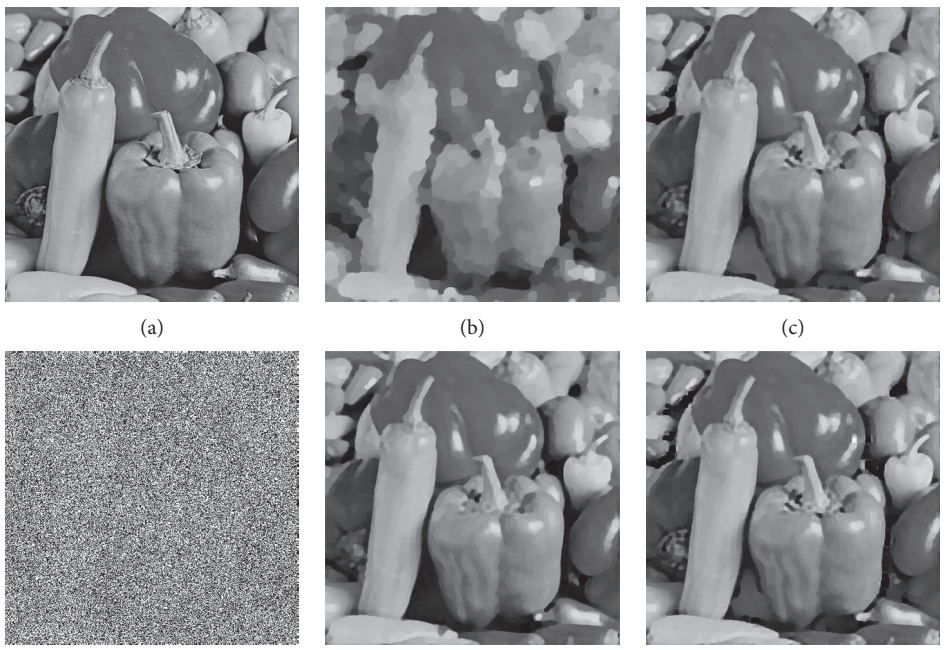
Figure 8: The recovery results (Peppers 512) by Gaussian blur with 90% SP noise. (a) Original. (b) TVL1, PSNR = 18.8935. (c) TVSCAD, PSNR = 24.1732. (d) Gaussian blur (lev = 0.9). (e) TVLog, PSNR = 24.7241. (f) LogTVSCAD, PSNR = 25.6992.



(a) (b) (c)

(d) (e) (f)

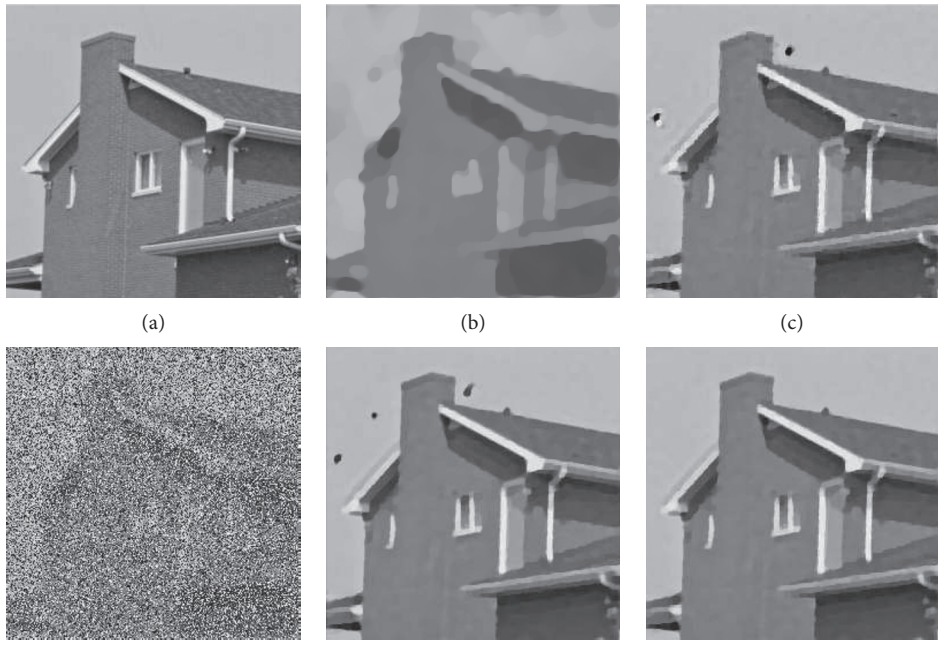
Figure 9: The recovery results (Peppers 512) by Motion blur with 90% SP noise. (a) Original. (b) TVL1, PSNR = 19.4711. (c) TVSCAD, PSNR = 25.3923. (d) Motion blur (lev = 0.9). (e) TVLog, PSNR = 25.5329. (f) LogTVSCAD, PSNR = 25.7575.



(a) (b) (c)

(d) (e) (f)

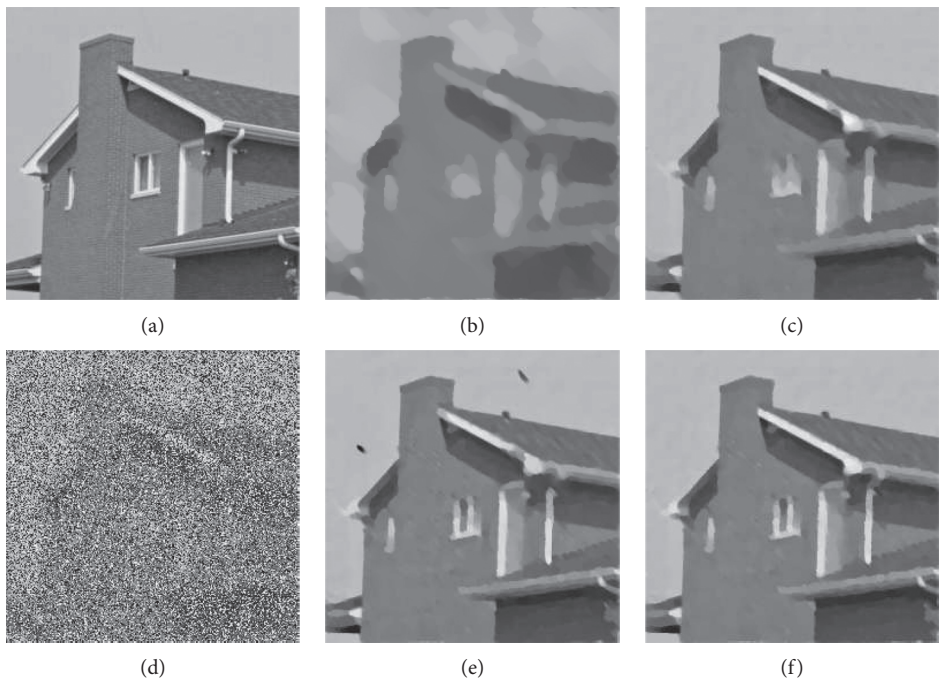
Figure 10: The recovery results (Peppers 512) by Average blur with 90% SP noise. (a) Original. (b) TVL1, PSNR = 19.2008. (c) TVSCAD, PSNR = 24.3041. (d) Average blur (lev = 0.9). (e) TVLog, PSNR = 25.2903. (f) LogTVSCAD, PSNR = 25.7948.



(a) (b) (c)

(d) (e) (f)

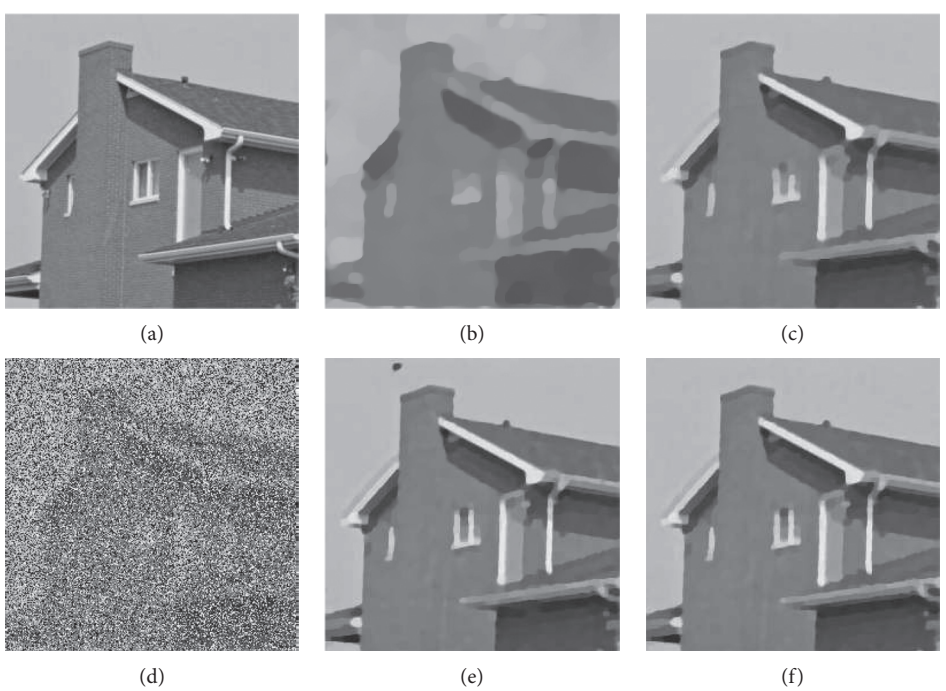
Figure 11: The recovery results (House 256) by Gaussian blur with 70% RV noise. (a) Original. (b) TVL1, PSNR = 20.9161. (c) TVSCAD, PSNR = 26.9073. (d) Gaussian blur (lev = 0.7). (e) TVLog, PSNR = 27.6482. (f) LogTVSCAD, PSNR = 29.0782.



(a) (b) (c)

(d) (e) (f)

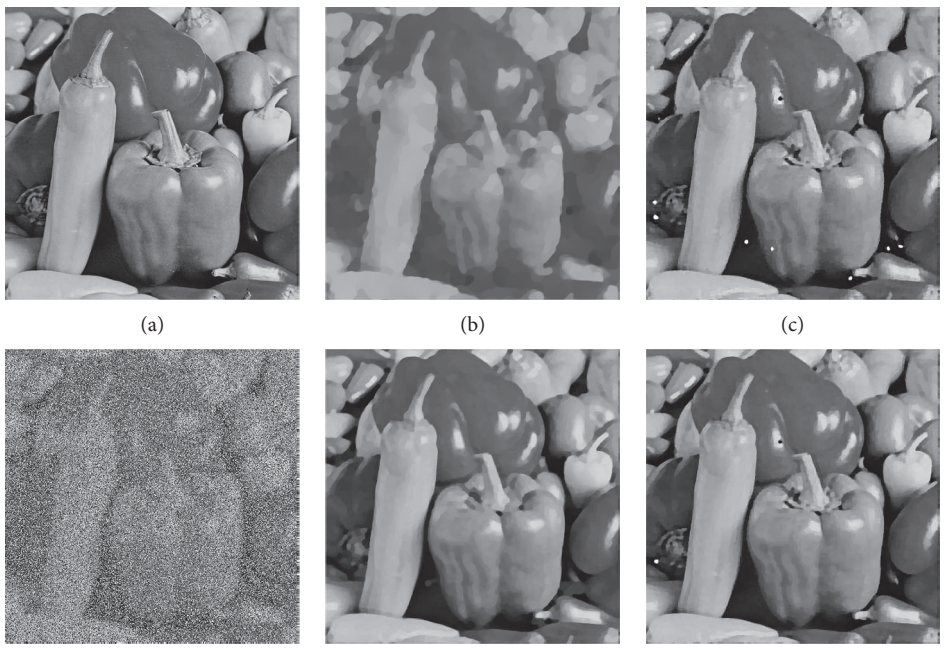
Figure 12: The recovery results (House 256) by Motion blur with 70% RV noise. (a) Original. (b) TVL1, PSNR = 20.4153. (c) TVSCAD, PSNR = 25.1866. (d) Motion blur (lev = 0.7). (e) TVLog, PSNR = 25.9254. (f) LogTVSCAD, PSNR = 27.1259.



(a) (b) (c)

(d) (e) (f)

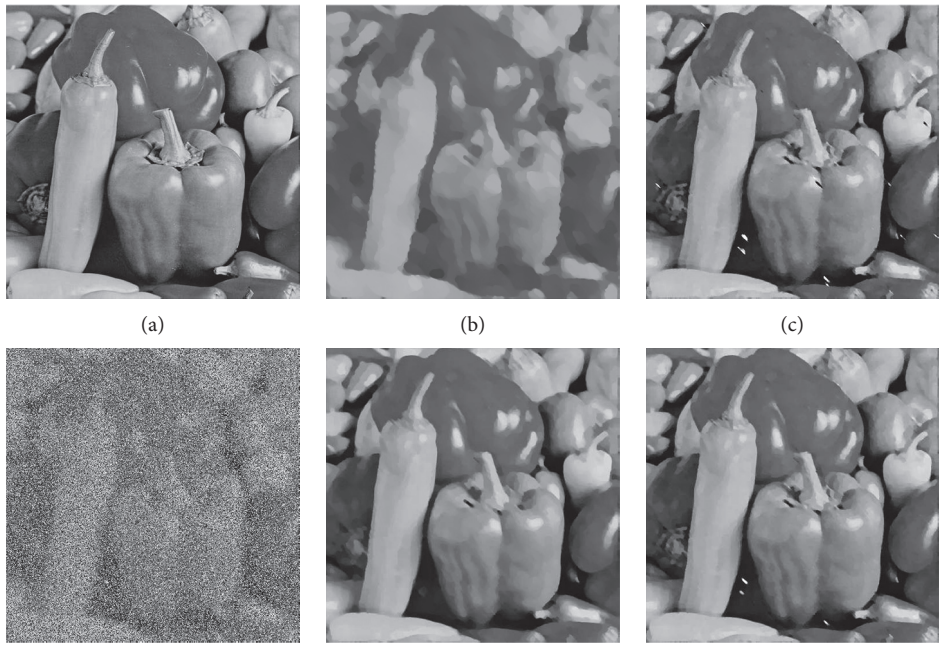
Figure 13: The recovery results (House 256) by Average blur with 70% RV noise. (a) Original. (b) TVL1, PSNR = 20.6806. (c) TVSCAD, PSNR = 26.7422. (d) Average blur (lev = 0.7). (e) TVLog, PSNR = 27.9585. (f) LogTVSCAD, PSNR = 28.5088.



(a) (b) (c)

(d) (e) (f)

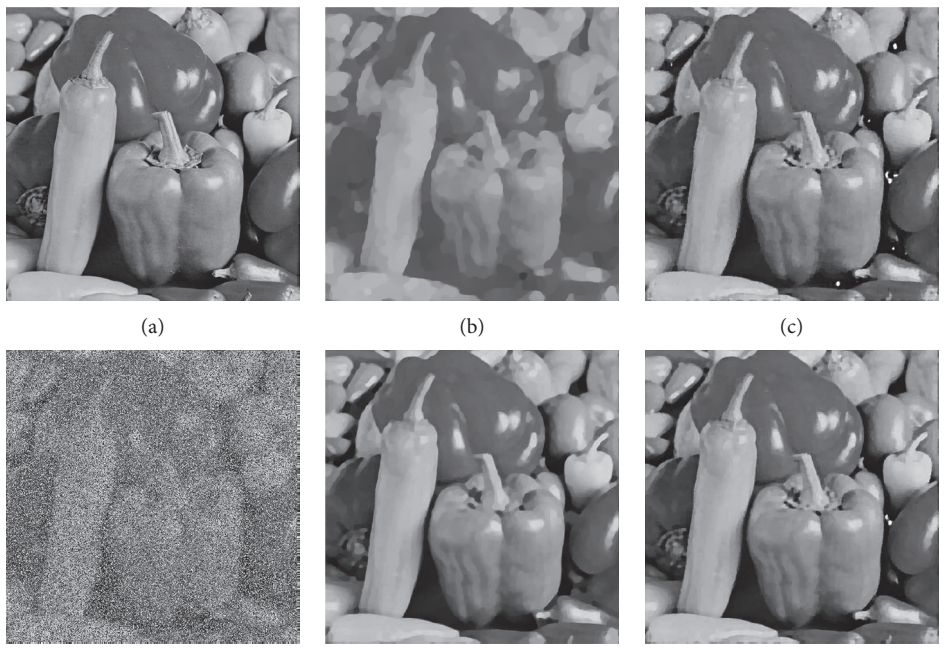
Figure 14: The recovery results (Peppers 512) by Gaussian blur with 70% RV noise. (a) Original. (b) TVL1, PSNR = 19.1797. (c) TVSCAD, PSNR = 26.0516. (d) Gaussian blur (lev = 0.7). (e) TVLog, PSNR = 26.4081. (f) LogTVSCAD, PSNR = 27.9344.



(a) (b) (c)

(d) (e) (f)

Figure 15: The recovery results (Peppers 512) by Motion blur with 70% RV noise. (a) Original. (b) TVL1, PSNR = 19.1338. (c) TVSCAD, PSNR = 25.9661. (d) Motion blur (lev = 0.7). (e) TVLog, PSNR = 26.1602. (f) LogTVSCAD, PSNR = 26.8849.



(a) (b) (c)

(d) (e) (f)

Figure 16: The recovery results (Peppers 512) by Average blur with 70% RV noise. (a) Original. (b) TVL1, PSNR = 19.1807. (c) TVSCAD, PSNR = 25.9819. (d) Average blur (lev = 0.7). (e) TVLog, PSNR = 26.4268. (f) LogTVSCAD, PSNR = 27.6354.

Table 1: Performance comparison of different methods for SP and RV.

Image	Level	Method	Motion			Average			Gaussian		
			CPU	SNR	SSIM	CPU	SNR	SSIM	CPU	SNR	SSIM
House	0.6 (SP)	TVL1	5.1875	9.3729	0.7272	4.9688	10.8485	0.7484	5.4845	5.3156	0.6282
		TVSCAD	46.8756	15.9418	0.8442	47.1563	16.7165	0.8489	51.8125	16.6146	0.8465
		TVLog	47.0468	15.9401	0.8443	46.7344	16.6897	0.8487	43.4843	16.5666	0.8451
		LogTVSCAD	46.2187	**15.9501**	**0.8443**	46.5625	**16.7233**	**0.8490**	48.1562	**16.6208**	**0.8466**
House	0.9 (SP)	TVL1	5.2344	3.1335	0.6315	5.2031	5.2470	0.6346	5.4845	5.3156	0.6282
		TVSCAD	47.6563	10.8527	0.7644	46.1406	13.2556	0.7943	48.9375	13.360	0.7935
		TVLog	46.9375	9.9173	0.7551	45.2187	11.3900	0.7709	45.9063	10.6551	0.7661
		LogTVSCAD	47.3437	**11.7085**	**0.7655**	44.7812	**13.4710**	**0.7972**	44.6406	**13.5232**	**0.7958**
House	0.7 (RV)	TVL1	5.2969	6.7084	0.6603	5.6406	6.9738	0.6760	5.1093	7.2092	0.6770
		TVSCAD	51.906	11.4798	0.7671	50.3906	13.0353	0.7946	44.0625	13.2004	0.8120
		TVLog	49.75	12.2186	0.7971	50.8593	14.2516	0.8192	44.0469	13.2004	0.8232
		LogTVSCAD	50.2188	**13.4190**	**0.8026**	51.0937	**14.8019**	**0.8210**	44.3906	**15.3713**	**0.8287**
Peppers	0.9 (SP)	TVL1	30.3750	6.9405	0.6093	31.5937	5.5903	0.6282	30.6093	6.6347	0.6255
		TVSCAD	146.6250	12.8617	0.7461	146.1719	11.2206	0.7616	148.3593	12.3083	0.7656
		TVLog	145.5781	13.0023	0.7621	146.8437	12.4857	0.7755	148.0625	12.2156	0.7723
		LogTVSCAD	147.9843	**13.2269**	**0.7674**	148.5468	**12.9662**	**0.7756**	148.9062	**12.7609**	**0.7746**
Peppers	0.7 (RV)	TVL1	29.9843	6.6032	0.6290	30.0312	6.6501	0.6433	30.4218	6.6491	0.6453
		TVSCAD	284.875	13.4355	0.7837	285.7343	13.4512	0.7904	286.5781	13.5209	0.7924
		TVLog	297.7812	13.6295	0.8039	304.6406	13.8962	0.7925	288.9062	13.8775	0.7921
		LogTVSCAD	286.3437	**14.3542**	**0.8043**	290.0937	**15.1048**	**0.8130**	287.875	**15.4038**	**0.8136**

The bold values indicate that the SNR and SSIM results obtained by LogTVSCAD are better than the other three methods.

4. Convergence Analysis

In this section, we analyze the convergence of Algorithm LogTVSCAD. First, we present the following lemma, which is the basis for proving the global convergence.

Lemma 1. *For given parameter $\eta > 0$, the sequence $\{\mathbf{u}_k\}$ generated by Algorithm LogTVSCAD satisfies $F(\mathbf{u}_k) - F(\mathbf{u}_{k+1}) \geq \eta \|\mathbf{u}_k - \mathbf{u}_{k+1}\|$.*

Proof. By the definitions of $\mathbf{j}(\mathbf{u})$ and $\mathbf{h}(\mathbf{u})$ (28), we obtain

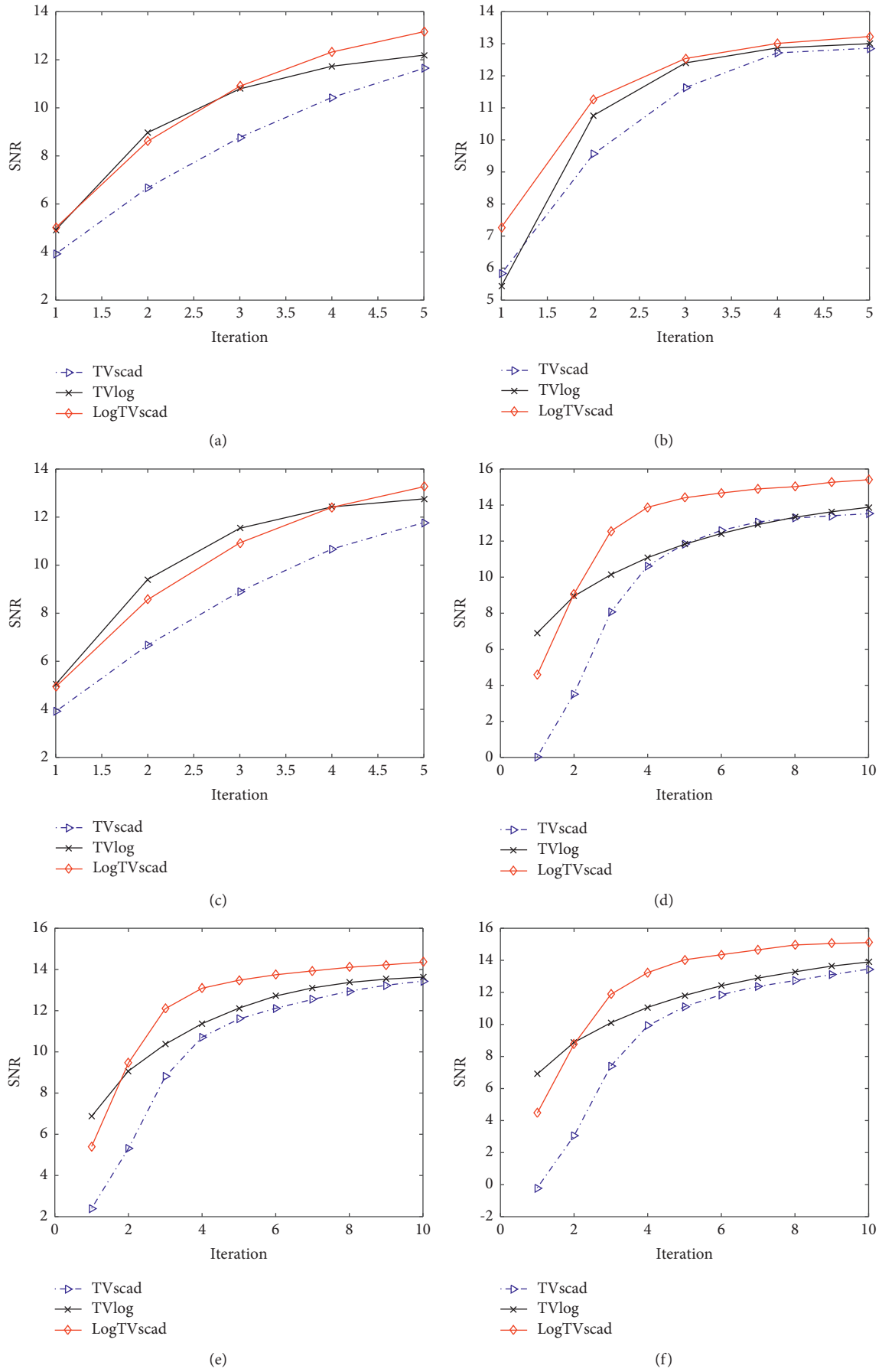
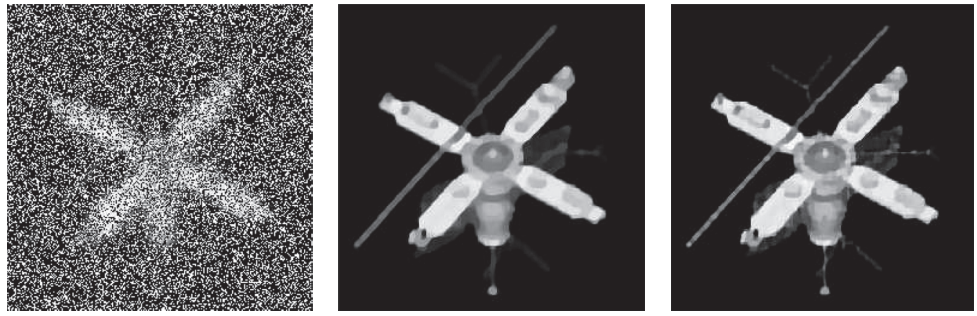


Figure 17: SNR versus iteration number for the Peppers image. (a) Gaussian, 90% SP noise. (b) Motion, 90% SP noise. (c) Average, 90% SP noise. (d) Gaussian, 70% RV noise. (e) Motion, 70% RV noise. (f) Average, 70% RV noise.

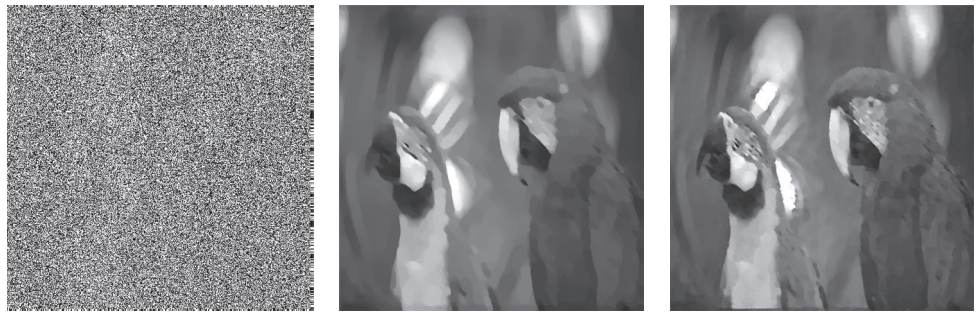
(a) (b) (c)

Figure 18: Original images. (a) Satellite (256). (b) Macaws (512). (c) Boat (1024).



(a) (b) (c)

Figure 19: The recovery results (satellite 256×256) by Average blur with 60% SP noise. (a) Average blur (lev = 0.6). (b) L0TV, PSNR = 24.4399. (c) LogTVSCAD, PSNR = 25.5613.



(a) (b) (c)

Figure 20: The recovery results (macaws 512×512) by Motion blur with 90% SP noise. (a) Motion blur (lev = 0.9). (b) L0TV, PSNR = 27.1022. (c) LogTVSCAD, PSNR = 27.5358.

$$\begin{cases} \mathbf{j}(\mathbf{u_k}) - \mathbf{j}(\mathbf{u_{k+1}}) \geq \langle \partial \mathbf{j}(\mathbf{u_{k+1}}), \mathbf{u_k} - \mathbf{u_{k+1}} \rangle, \\ \mathbf{h}(\mathbf{u_{k+1}}) - \mathbf{h}(\mathbf{u_k}) \geq \langle \partial \mathbf{h}(\mathbf{u_k}), \mathbf{u_{k+1}} - \mathbf{u_k} \rangle. \end{cases} \tag{30}$$

Following (20), we have

$$\begin{aligned} &\mu \mathbf{K}^{\mathbf{T}} \nabla \mathbf{G}_{\gamma}(\mathbf{K}\mathbf{u_{k+1}} - \mathbf{f}) + \eta(\mathbf{u_k} - \mathbf{u_{k+1}}) \in \partial \mathbf{j}(\mathbf{u_{k+1}}), \\ &\mathbf{D}^{\mathbf{T}} \nabla \mathbf{G}_{\mathbf{s}}(\mathbf{D}\mathbf{u_k}) + \mu \mathbf{K}^{\mathbf{T}} \nabla \mathbf{G}_{\gamma}(\mathbf{K}\mathbf{u_k} - \mathbf{f}) \in \partial \mathbf{h}(\mathbf{u_k}). \end{aligned} \tag{31}$$

We obtain from (30) that $F(\mathbf{u}_k) - F(\mathbf{u}_{k+1}) \geq \eta \|\mathbf{u}_k - \mathbf{u}_{k+1}\|$. It is easy to see that $\|\mathbf{Du}\|_{1/2}$ and $\|\mathbf{Ku} - \mathbf{f}\|_1$ are semialgebraic from their definitions. We also know from the literature [26, 27] that the penalty functions SCAD and the Log enjoy the KL property. Then, the following result is obtained. □

Lemma 2. *For given parameter $\eta > 0$, the function $\mathbf{F}(\mathbf{u})$ is a* **KL** *function.*

Lemma 3. *Let $\mathbf{d_k} \in \partial \mathbf{F}(\mathbf{u_k})$, for a sufficiently large constant $M > 0$, and then*

$$\|\mathbf{d_{k+1}}\| \leq \mathbf{M} \|\mathbf{u_{k+1}} - \mathbf{u_k}\|^2. \tag{32}$$

Proof. It follows from (20) that

(a) (b) (c)

Figure 21: The recovery results (boat 1024×1024) by Gaussian blur with 30% RV noise. (a) Gaussian blur (lev = 0.3). (b) L0TV, PSNR = 31.4702. (c) LogTVSCAD, PSNR = 34.3214.

Table 2: Performance comparison of L0TV and LogTVSACD methods.

Image	Level	Method	Motion			Average			Gaussian		
			PSNR	SNR	SSIM	PSNR	SNR	SSIM	PSNR	SNR	SSIM
Satellite	0.6(SP)	L0TV	24.4391	13.2826	0.8625	24.4398	13.2835	0.8653	24.4000	13.2435	0.8648
		LogTVSCAD	25.7365	14.5800	0.9063	25.5614	14.4049	0.8987	25.4307	14.2742	0.8951
Macaws	0.9(SP)	L0TV	26.4027	9.7899	0.7066	27.0230	10.4192	0.7155	27.0000	10.3962	0.7157
		LogTVSCAD	27.5733	10.9695	0.7348	27.6667	11.0629	0.7324	27.4857	10.8829	0.7310
Boat	0.3(RV)	L0TV	30.3035	15.5168	0.8190	31.4866	16.6999	0.8408	31.4702	16.6834	0.8408
		LogTVSCAD	33.7020	18.9152	0.8965	34.3836	19.5969	0.8996	34.3214	19.5347	0.8984

$$\mathbf{d}_{\mathbf{k}+1} = \mathbf{D}^{\mathbf{T}}\nabla\mathbf{G}_{\mathbf{s}}\left(\mathbf{D}\mathbf{u}_{\mathbf{k}}\right) + \mu\mathbf{K}^{\mathbf{T}}\nabla\mathbf{G}_{\gamma}\left(\mathbf{K}\mathbf{u}_{\mathbf{k}} - \mathbf{f}\right) - \mu\mathbf{K}^{\mathbf{T}}\nabla\mathbf{G}_{\gamma} \cdot \left(\mathbf{K}\mathbf{u}_{\mathbf{k}+1} - \mathbf{f}\right) - \eta\left(\mathbf{u}_{k} - \mathbf{u}_{\mathbf{k}+1}\right) \in \partial\mathbf{F}\left(\mathbf{u}_{\mathbf{k}+1}\right). \tag{33}$$

According to [35], $\nabla\mathbf{G}_s$ and $\nabla\mathbf{G}_\gamma$ are Lipschitz continuous. Moreover, $\{\mathbf{F}(\mathbf{u}_\mathbf{k})\}$ is sufficiently decreasing from Lemma 1. Thus, we obtain $\|\mathbf{d}_{\mathbf{k}+1}\| \le \mathbf{M}\|\mathbf{u}_{\mathbf{k}+1} - \mathbf{u}_\mathbf{k}\|^2$, where $M > 0$ is a sufficiently large constant. □

Theorem 1. *Let the sequence $\{\mathbf{u}_k\}$ generated by Algorithm LogTVSCAD, and then it converges to a critical point of (5).*

Proof. Since $\{\mathbf{u}_k\}$ is bounded, it follows from Lemma 1 that

$$\sum_{\mathbf{k}=0}^{\infty} \left\|\mathbf{u}_{\mathbf{k}+1} - \mathbf{u}_\mathbf{k}\right\|^2 \le \frac{\mathbf{F}\left(\mathbf{u}_0\right)}{\boldsymbol{\eta}}. \tag{34}$$

Then, $k \longrightarrow \infty$, $(\mathbf{u}_{\mathbf{k}+1} - \mathbf{u}_\mathbf{k}) \longrightarrow 0$. Let $\mathbf{u}^*$ be any accumulation point of the sequence $\{\mathbf{u}_k\}$. There exists a subsequence $\left\{\mathbf{u}_{n_k}\right\}$ such that

$$\mathbf{u}_{n_k} \longrightarrow \mathbf{u}^*, F\left(\mathbf{u}_{n_k}\right) \longrightarrow F\left(\mathbf{u}^*\right), \quad k \longrightarrow \infty. \tag{35}$$

Thus, combining Lemma 2 and 3 and Theorem 2.9 [38], the result is right and $0 \in \partial F(\mathbf{u}^*)$. □

5. Numerical Results

In this section, we evaluate the performance of the proposed LogTVSCAD method through some numerical experiments. We provide the results of classical TVL1 [9] for reference purpose, and we also compare with TVSCAD nonconvex model [26] and TVLog nonconvex model [27]. All of our test experiments are performed on MATLAB R2015a on the PC with Intel(R) Core(TM) 2.2 GHz CPU and 8.0 GB RAM.

To assess the quality of the recovered image, we use the peak signal-to-noise ratio (PSNR) and the signal-to-noise ratio (SNR) as the evaluation indicators, because they are widely used in image processing kinds of literature. The higher PSNR and SNR indicate better quality image; they are defined as follows:

$$\begin{aligned} \mathrm{PSNR} &= 10\lg\left(\frac{P}{\left\|x^k - x\right\|^2}\right), \\ \mathrm{SNR} &= 20\lg\left(\frac{\|x - \overline{x}\|}{\left\|x^k - x\right\|}\right), \end{aligned} \tag{36}$$

where P is the size of the image and $\mathbf{x}$ and $\mathbf{x}^k$ denote the original image and restored image, respectively. $\overline{\mathbf{x}}$ is the mean intensity value of $\mathbf{x}$. The structural similarity (SSIM) is another effective evaluation index for comparing the image quality, and the SSIM value closer to the one showing better structure preservation; for more details, please see [39].

In our experiments, we choose two images of House (256) and Peppers (512) as the test images, because they are widely used in the academic research work of image processing. The three types of blurs tested are generated by the MATLAB function **fspecial**, and they are Gaussian (hsize = 9, std = 10), motion (len = 21, angle = 135^0), and average (hsize = 9) blurs, respectively. Two common types of impulsive noise are tested: salt-and-pepper (SP) and random-valued (RV) noises. It is well known that RV noise is more

difficult to remove than SP noise. In the experiments, we tested with various noise levels: 60%, 90% SP noise, and 70% RV noise.

The selection of parameters in models and algorithms is an open question. If we use the same parameters, the performances of different methods may have opposite results. The best parameters selection of TVL1, TVSCAD, and TVLog adopts the recommendations of the literatures [26, 27]. For the regularization parameter μ, we swept over

$$\mu \in \{1, 5, 10, 15, 20, 25, 30, 50, 100, 200, 400, 450, 500\}. \tag{37}$$

The penalty parameters $\beta_w, \beta_z, \beta_x$ of (24) are chosen from

$$\{\beta_w, \beta_z, \beta_x\} \in \{1, 1.5, 5, 10, 20, 25, 50, 100, 200, 1000, 2000\}. \tag{38}$$

In LogTVSCAD model, we set the parameters $\eta = 0.001$, $s = 0.002 * k$, $\gamma_1 = 0.08/k$, and $\gamma_2 = \max(0.2 * 0.68^{k-1}, 0.002)$ for the SP noise. For RV noise, we set the parameters $s = 0.02 * k$, $\gamma_1 = 0.08/k$ and $\gamma_2 = \max(0.2 * 0.85^{k-1}, 0.02)$, where k denotes the number of iterations. From all the tested cases, we can see that the performance of the three nonconvex models (TVSCAD, TVLog, and LogTVSCAD) outperforms that of the TVL1 model. Figures 2–4 show that the three nonconvex models have approximately the same effect at low noise level. In Figures 5–10, we show the restored results by TVL1, TVSCAD, TVLog, and LogTVSCAD on test images House (256) and Peppers (512) with Gaussian blur, Motion blur, and Average blur, and the SP noise level is 90%, respectively. These experimental results show that our LogTVSCAD is slightly better than the other methods at the high noise level. For the 70% RV noise, the higher PSNR values of Figures 11–16 indicate that the LogTVSCAD method is also better than the other three methods. Further, all the test results including CPU time, SNR, and SSIM are listed in Table 1. Through the table, we note that the LogTVSCAD model achieves higher PSNR and SSIM values in various kinds of cases. The experimental results show that the LogTVSCAD model has a better performance for both low and high noise levels. The results in Table 1 also show that the TVL1 model has superiority in CPU time cost.

Figure 17 shows the convergence curves of the three methods of LogTVSCAD, TVLog, and TVSCAD. For the Peppers image with 90% SP noise, Figures 17(a)–17(c) show the plots of SNR versus iteration number under Gaussian blur, Motion blur, and Average blur, respectively. It is easy to see that the SNR values increase as the number of iterations increases. The curves of LogTVSCAD are slightly higher than those of TVLog and TVSCAD. From Figures 17(d)–17(f), for 70% RV noise, we can see that the proposed LogTVSCAD has better performance than TVSCAD and TVLog. Hence, the proposed new model is effective and desirable.

In the end, we compare our method to the recent other nonconvex work (called L0TV) [40]. The authors reformulated the L0TV model as an equivalent biconvex mathematical program with equilibrium constraints and then solved it using the proximal ADMM. In the experiments, we chose three other typical images to test, i.e., satellite (256), macaws (512), and boat (1024), which are shown in Figure 18. The blur types still adopt the motion, average, and Gaussian blurs. In the experiments, we tested 60% SP noise, 90% SP noise, and 70% RV noise. In terms of parameter settings, satellite (256) and macaws (512) images adopt the same settings as before. For boat (1024) image, we consider selecting $\mu = 30$, and other parameters remain unchanged. The parameters selection of L0TV method adopt the recommendations of the literature [40]. Figures 19–21 show some visual comparisons of image restorations. We observe that the L0TV and LogTVSCAD can restore the blur and noise images well. In Figure 21, the PSNR values of both methods are greater than 30, which shows that both methods are effective for low-level noise. The numerical comparison between the two methods is shown in Table 2. It is obvious that the proposed LogTVSCAD also performs well.

6. Conclusions

In this paper, we proposed a new LogTVSCAD model for image restoration with impulsive noise. To solve the nonconvex model, we firstly apply the DCA to reformulate the nonconvex problem and then use the ADMM method to solve the subproblem. The global convergence of the proposed algorithm is proved. The experimental results on recovering images show that the proposed LogTVSCAD model is an effective approach in impulsive noise and is competitive with TVL1, TVSCAD, and TVLog. In future work, we will consider the application of this method in other fields and investigate other nonconvex reconstruction methods.

Data Availability

The data used to support the findings of this study are available from the corresponding authors upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 11901137 and 61967004, in part by the China Postdoctoral Science Foundation under Grant 2020M682959, in part by the Natural Science Foundation of Guangxi Province under Grant 2018GXNSFBA281023, in part by Scientific Research Fund of Hunan Provincial Education Department under Grant 20A273, and in part by Research Fund of Mathematics Discipline of Hunan University of Humanities, Science and Technology under Grant 2020SXJJ01.

References

[1] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numerica*, vol. 25, pp. 161–319, 2016.

[2] R. Gonzalez and R. Woods, *Digital Image Processing*, Pearson, London, UK, 3rd edition, 2008.

[3] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[4] A. Chambolle and P.-L. Lions, "Image recovery via total variation minimization and related problems," *Numerische Mathematik*, vol. 76, no. 2, pp. 167–188, 1997.

[5] T. Chan, S. Esedoglu, F. Park et al., "Total variation image restoration: overview and recent developments," in *Handbook of Mathematical Models in Computer Vision*, N. Paragios, Y. Chen, and O. Faugeras, Eds., pp. 17–31, Springer, Boston, MA, USA, 2006.

[6] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2842–2865, 2009.

[7] Y. Dong, M. Hintermüller, and M. Neri, "An efficient primal-dual method for $L1$TV image restoration," *SIAM Journal on Imaging Sciences*, vol. 2, no. 4, pp. 1168–1189, 2009.

[8] C. Clason, B. Jin, and K. Kunisch, "A duality-based splitting method for $\ell1$-TV image restoration with automatic regularization parameter choice," *SIAM Journal on Scientific Computing*, vol. 32, no. 3, pp. 1484–1505, 2010.

[9] R. H. Chan, M. Tao, and X. Yuan, "Constrained total variation deblurring models and fast algorithms based on alternating direction method of multipliers," *SIAM Journal on Imaging Sciences*, vol. 6, no. 1, pp. 680–697, 2013.

[10] H. Fu, M. K. Ng, M. Nikolova, and J. L. Barlow, "Efficient minimization methods of mixed l2-l1 and l1-l1 norms for image restoration," *SIAM Journal on Scientific Computing*, vol. 27, no. 6, pp. 1881–1902, 2006.

[11] T. Goldstein and S. Osher, "The split bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[12] J.-F. Cai, S. Osher, and Z. Shen, "Split bregman methods and frame based image restoration," *Multiscale Modeling and Simulation*, vol. 8, no. 2, pp. 337–369, 2010.

[13] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, vol. 40, no. 1, pp. 120–145, 2011.

[14] J. Junfeng Yang, Y. Yin Zhang, and W. Wotao Yin, "A fast alternating direction method for TVL1-L2 signal reconstruction from partial fourier data," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 288–297, 2010.

[15] M. Bai, X. Zhang, and Q. Shao, "Adaptive correction procedure for TVL1 image deblurring under impulse noise," *Inverse Problems*, vol. 32, no. 8, Article ID 085004, 2016.

[16] I. Selesnick, "Total variation denoising via the moreau envelope," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 216–220, 2017.

[17] W. Wang and J. Wang, "Enhancing matrix completion using a modified second-order total variation," *Discrete Dynamics in Nature and Society*, vol. 2018, Article ID 2598160, 12 pages, 2018.

[18] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[19] S. Foucart and M. Lai, "Sparsest solutions of underdetermined linear systems via ℓ_q-minimization for $0 < q \leq 1$," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 395–407, 2009.

[20] Z. Zongben Xu, X. Xiangyu Chang, F. Fengmin Xu, and H. Zhang, "$L\{1/2\}$ regularization: a thresholding representation theory and a fast solver," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 7, pp. 1013–1027, 2012.

[21] E. Candes, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted L1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2007.

[22] J. H. Friedman, "Fast sparse regression and classification," *International Journal of Forecasting*, vol. 28, no. 3, pp. 722–738, 2012.

[23] C. Zhang, "Nearly unbiased variable selection under minimax concave penalty," *The Annals of Statistics*, vol. 38, no. 2, pp. 894–942, 2010.

[24] P. Gong, C. Zhang, Z. Lu et al., "A general iterative shrinkage and thresholding algorithm for non-convex regularized optimization problems," in *Proceedings of the 30th International Conference on Machine Learning*, pp. 37–45, Atlanta, GA, USA, June 2013.

[25] X. Zhang, M. Bai, and M. K. Ng, "Nonconvex-TV based image restoration with impulse noise removal," *SIAM Journal on Imaging Sciences*, vol. 10, no. 3, pp. 1627–1667, 2017.

[26] J. Yang, G. Gu, and S. Jiang, "A TVSCAD approach for image deblurring with impulsive noise," *Inverse Problems*, vol. 33, no. 12, Article ID 125008, 2017.

[27] B. Zhang, G. Zhu, and Z. Zhu, "A TV-log nonconvex approach for image deblurring with impulsive noise," *Signal Processing*, vol. 174, Article ID 107631, 2020.

[28] Z.-X. Cui and Q. Fan, "A "Nonconvex+Nonconvex" approach for image restoration with impulse noise removal," *Applied Mathematical Modelling*, vol. 62, pp. 254–271, 2018.

[29] Y. Lou, T. Zeng, S. Osher, and J. Xin, "A weighted difference of anisotropic and isotropic total variation model for image processing," *SIAM Journal on Imaging Sciences*, vol. 8, no. 3, pp. 1798–1823, 2015.

[30] H. A. Le Thi, T. Pham Dinh, H. M. Le, and X. T. Vo, "DC approximation approaches for sparse optimization," *European Journal of Operational Research*, vol. 244, no. 1, pp. 26–46, 2015.

[31] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations & Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2013.

[32] Y. Ji and S. Qu, "Proximal point Algorithms for vector DC programming with applications to probabilistic lot sizing with service levels," *Discrete Dynamics in Nature and Society*, vol. 2017, Article ID 5675183, 2017.

[33] R. Rockafellar and R. Wets, *Variational Analysis*, Springer Science Business Media, Berlin, Germany, 2009.

[34] R. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, NJ, USA, 1997.

[35] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, New York, NY, USA, 2011.

[36] J. Bolte, A. Daniilidis, and A. Lewis, "The lojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 1205–1223, 2007.

[37] H. A. Le Thi and T. Pham Dinh, "DC programming and DCA: thirty years of developments," *Mathematical Programming*, vol. 169, no. 1, pp. 5–68, 2018.

[38] H. Attouch, J. Bolte, and B. F. Svaiter, "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods," *Mathematical Programming*, vol. 137, no. 1-2, pp. 91–129, 2013.

[39] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[40] G. Yuan and B. Ghanem, "L0TV: TV: a sparse optimization method for impulse noise image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 352–364, 2019.