Numer. Math. Theor. Meth. Appl.
doi: 10.4208/nmtma.OA-2022-0190

Vol. **xx**, No. x, pp. 1-22
x 2023

Low Rank and Total Variation Based Two-Phase Method for Image Deblurring with Salt-and-Pepper Impulse Noise

Yuchao Tang[1,2,*], Shirong Deng[2] and Tieyong Zeng[3]

[1] *School of Mathematics and Information Science, Guangzhou University, Guangzhou 510006, China*
[2] *Department of Mathematics, Nanchang University, Nanchang 330031, China*
[3] *Department of Mathematics, The Chinese University of Hong Kong, Shatin, Hong Kong, SAR, China*

Received 1 December 2022; Accepted (in revised version) 5 July 2023

Abstract. Although there are many effective methods for removing impulse noise in image restoration, there is still much room for improvement. In this paper, we propose a new two-phase method for solving such a problem, which combines the nuclear norm and the total variation regularization with box constraint. The popular alternating direction method of multipliers and the proximal alternating direction method of multipliers are employed to solve this problem. Compared with other algorithms, the obtained algorithm has an explicit solution at each step. Numerical experiments demonstrate that the proposed method performs better than the state-of-the-art methods in terms of both subjective and objective evaluations.

AMS subject classifications: 65K05, 90C25, 94A08
Key words: Image deblurring, impulse noise, total variation, nuclear norm.

---

1. Introduction

Impulse noise removal is a challenging problem in image restoration. In general, the image restoration problem, which is subject to blurring and impulse noise, can be expressed as

$$f = N_{imp}(y), \quad y = Kx, \tag{1.1}$$

where N_{imp} denotes impulse noise, K is a linear blurring operator, $f \in R^{m\times n}$ is the observed image, and $x \in R^{m\times n}$ is the unknown true image. Two types of impulse noise are widely studied: salt-and-pepper (SP) impulse noise and random-valued (RV) impulse noise. Let the dynamic range of y belong to $[y_{\min}, y_{\max}]$, i.e., $y_{\min} \le y_{ij} \le y_{\max}$, for all $1 \le i \le m$, $1 \le j \le n$.

---

*Corresponding author. *Email address:* `hhaaoo1331@163.com` (Y. Tang)

Salt-and-pepper impulse noise. Observed image f satisfies

$$f_{ij} = \begin{cases} y_{\min}, & \text{with probability } s/2, \\ y_{\max}, & \text{with probability } s/2, \\ y_{ij}, & \text{with probability } 1-s, \end{cases} \tag{1.2}$$

where s denotes the level of the salt-and-pepper noise.

Random-valued impulse noise. Observed image f satisfies

$$f_{ij} = \begin{cases} d_{ij}, & \text{with probability } r, \\ y_{ij}, & \text{with probability } 1-r, \end{cases} \tag{1.3}$$

where d_{ij} are uniformly distributed random numbers in $[y_{\min}, y_{\max}]$ and r denotes the level of the random valued noise.

The most popular model for image deblurring with impulse noise is the so-called L1-TV, which is defined by

$$\min_{x} \ \|Kx - f\|_1 + \lambda\varphi(Lx), \tag{1.4}$$

where $\lambda > 0$ is the regularization parameter, $L : R^{m\times n} \to R^{m\times 2n}$ is the first-order difference matrix, and $\varphi : R^{m\times 2n} \to R$ is a convex function. If $\varphi(\cdot) = \|\cdot\|_2$ or $\varphi(\cdot) = \|\cdot\|_1$, $\varphi(Lx)$ denotes the isotropic total variation (ITV) and the anisotropic total variation (ATV), respectively. The first term in (1.4) is usually called data fidelity term, and the second term is called regularization term. Compared with the classical L2-data fidelity term, the L1-data fidelity term is robust for removing outliers. The L1-TV model (1.4) is difficult to solve because of the nondifferentiable of both the L1-norm data fidelity term and the TV term. In the last two decades, many efficient iterative algorithms have been proposed to solve (1.4). These include the primal-dual interior point algorithm [21], alternating minimization algorithm [23], alternating direction method of multiplies [12, 34], and the primal-dual Chambolle-Pock algorithm [6].

Although the L1-TV model (1.4) is effective in removing impulse noise, it does not take into account whether a pixel is contaminated by noise or not. The performance of L1-TV is usually unsatisfactory when the noise level is high, as demonstrated in studies such as [13, 36]. To address this issue, two-phase methods have gained popularity. In the first phase, techniques such as the adaptive median (AM) filter or adaptive center-weighted median (ACWM) filter are used to identify image pixels affected by salt-and-pepper impulse noise or random-valued impulse noise. In the second phase, filter methods or detail-preserving regularization methods based on the identified noise-free pixels are utilized to recover the image. Chan *et al.* [9–11] first proposed a two-phase method for removing random-valued impulse noise and salt-and-pepper impulse noise, respectively. They considered solving a variational minimization problem in the second phase, see also [4, 7, 17]. On the other hand, Chen and Yang [16] introduced a two-stage method for removing impulse noise, which used an iterative and adaptive median-based filter in the second phase. The filter methods in the second phase have been further improved in [15, 26–28] and many others. Since these works mentioned above are only designed for pure impulse noise removal, Cai *et al.* [5] proposed a two-phase method for image deblurring with impulse noise. In contrast, Ma *et al.* [29] introduced a general model that includes one-phase and two-phase methods. They also considered the box constraint $[0, 1]$ on the pixel values of the image. Moreover, Ma *et al.* [30] proposed a path-based two-phase method that incorporated the sparse representation prior and the total variation regularization.

Existing two-phase methods are based on total variation regularization for restoring blurred images with impulse noise [5, 8, 29, 30], which require the empirical selection of regularization parameters. To overcome this drawback, Sciacchitano *et al.* [31] proposed a parameter-free model as follows:

$$\begin{aligned} &\min_{x} \ \varphi(Lx) \\ &\text{s.t.} \ (Kx)_{ij} = f_{ij}, \quad (i,j) \in U, \end{aligned} \tag{1.5}$$

where φ, L, K, and f are the same as (1.4), and U denotes the location of noise-free pixels. The semismooth Newton method was introduced to solve the reduced convex minimization problem for $K = I$. For the general case (i.e., $K \neq I$), the primal-dual Chambolle-Pock algorithm was applied to deal with the equality constraint. It is worth mentioning that the model (1.5) does not consider the box constraint. In the following, we refer to (1.5) as the ExTV method. In contrast to the two-phase methods, there are many existing nonconvex models to improve the performance of the L1-TV model (1.4), such as Nonconvex TV [37,39], TVSCAD [22], TVLog [38], $\ell_0 TV$ [25,35], and Nonconvex-nonconvex [18], etc. In this paper, we mainly focus on extending the two-phase methods of the ExTV (1.5).

Low-rank prior has been exploited in many imaging applications, such as image super-resolution [14, 32], dynamic MRI [24], and functional MRI [33], etc. Although most natural images do not have strict low-rankness, they usually have approximately low-rankness. Therefore, by minimizing the nuclear norm, it prefers matrices that have a small number of singular values with large magnitudes. The main purpose of this paper is to propose a new two-phase model for image deblurring under impulse noise, which combines the nuclear norm and total variation regularization with box constraint. The new model is

$$\begin{aligned} &\min_{x} \ \varphi(Lx) + \mu\|x\|_* \\ &\text{s.t.} \ x \in C, \\ &\quad Kx \in \widetilde{C}, \end{aligned} \tag{1.6}$$

where $\mu > 0$ is the regularization parameter, $\|x\|_*$ denotes the nuclear norm, $C = \{x \in R^{m\times n} | 0 \leq x_{ij} \leq 1\}$, and $\widetilde{C} = \{y \in R^{m\times n} | y_{ij} = f_{ij}, (i,j) \in U\}$. When $C = R^{m\times n}$ and $\mu = 0$, the model (1.6) reduces to the parameter-free model (1.5). Therefore, the model (1.6) generalizes the model (1.5). When the TV term $\varphi(Lx)$ is missing, (1.6) reduces to the following constrained nuclear norm regularization model:

$$\begin{aligned} &\min_x \|x\|_* \\ &\text{s.t. } x \in C, \\ &\quad Kx \in \widetilde{C}. \end{aligned} \tag{1.7}$$

We employ the alternating direction method of multipliers (ADMM) and the proximal ADMM to solve (1.6). Since the ADMM needs to solve a subproblem, we use two different approaches. In detail, for periodic boundary conditions, we use the fast Fourier transform (FFT). For other boundary conditions, we add a proximal term to the subproblem to get a closed-form solution. Numerical experiments are conducted to demonstrate the performance of the proposed method, especially compared with the ExTV method (1.5) for image deblurring with impulse noise.

We summarize the contributions of this paper

1. We propose a new model (1.6) for image deblurring under salt-and-pepper impulse noise, which could be seen as a generalization of the ExTV method (1.5). In particular, when $\mu = 0$, a constrained ExTV is obtained. The proposed method can be used for other type of impulse noise after combining with proper noise detectors.

2. We employ the ADMM and the proximal ADMM to solve the proposed model. Each subproblem of the proposed algorithm has a closed-form solution. Additionally, we provide a first-order iterative algorithm to solve (1.5). Compared with the algorithms in [31], our algorithm does not require smoothing of the total variation for denoising or calculation of the matrix inverse for deblurring.

The rest of this paper is organized as follows. In Section 2, we briefly review some concepts and the key feature of the ADMM and the proximal ADMM. In Section 3, we present the main algorithm for solving the proposed model (1.6). In Section 4, we present extensive experiments to demonstrate the effectiveness and efficiency of the proposed method. Finally, we draw some conclusions.

2. Preliminaries

In this section, we briefly review some notations and definitions, which will be used throughout this paper. Let X be a finite dimensional real vector space, which equipped with inner product $\langle \cdot, \cdot \rangle$ and associated norm $\|\cdot\|$. Let $M \in R^{N\times N}$ be a self-adjoint and positive matrix, the scale norm $\|\cdot\|_M$ is defined by $\|x\|_M = \sqrt{\langle x, Mx\rangle}, x \in R^N$. The set of extended-real valued, lower-semicontinuous, proper, and convex functions on X is denoted by $\Gamma_0(X)$. The sign function, denoted by $\operatorname{sgn}(x)$. It returns 1 if x is positive, -1 if x is negative, and 0 if $x = 0$. Mathematically, it can be defined as

$$\operatorname{sgn}(x) = \begin{cases} -1, & \text{if} \quad x < 0, \\ 0, & \text{if} \quad x = 0, \\ 1, & \text{if} \quad x > 0. \end{cases}$$

Let C be a nonempty closed convex set of X. The indicator function of the set C is defined by

$$\delta_C(x) = \begin{cases} 0, & \text{if} \quad x \in C, \\ +\infty, & \text{otherwise}. \end{cases}$$

Let $u \in R^{m\times n}$ be a given discrete image, we recall the definition of the total variation is

$$TV(u) = \varphi(Lu) = \begin{cases} \|Lu\|_2 = \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} \sqrt{(\nabla_x u)_{ij}^2 + (\nabla_y u)_{ij}^2}, & \text{(ITV)} \\ \|Lu\|_1 = \sum\limits_{i=1}^{m}\sum\limits_{j=1}^{n} \big(|\nabla_x u|_{ij} + |\nabla_y u|_{ij}\big), & \text{(ATV)} \end{cases}$$

where

$$\|y\|_2 = \sum_{i=1}^{m}\sum_{j=1}^{n} \sqrt{\left(y_{ij}^1\right)^2 + \left(y_{ij}^2\right)^2}, \quad y = (y^1, y^2), \quad y^1, y^2 \in R^{m\times n},$$

the discrete gradient operator $L = \begin{pmatrix} \nabla_x & \nabla_y \end{pmatrix}$, where ∇_x and ∇_y denote the horizonal and vertical first order differences, respectively.

The proximal operator plays a virtual role in studying many first-order convex minimization algorithms, see, e.g., [1].

Definition 2.1. *Let g be a proper lower semicontinuous convex function, the proximal operator of g with index $\lambda > 0$ is defined by*

$$prox_{\lambda g}(y) = \arg\min_x \left\{ \frac{1}{2\lambda}\|x - y\|^2 + g(x) \right\}.$$

The proximal operator is a generalization of the classical orthogonal projection P_C with $g(x) = \delta_C(x)$. That is

$$prox_{\delta_C}(y) = P_C(y) = \arg\min_{x\in C} \|x - y\|.$$

Let $x \in R^n$ and $\lambda > 0$, the proximal operator of the ℓ_1-norm $\|x\|_1$ is the so-called soft-thresholding operator, which is defined by

$$prox_{\lambda\|\cdot\|_1} = Soft(y, \lambda) = \big(\max(|y_i| - \lambda, 0) * \operatorname{sgn}(y_i)\big), \quad i = 1, \dots, n.$$

The nuclear norm of $X \in R^{m\times n}$ is defined to be $\|X\|_* = \sum_{i=1}^{r} \sigma_i(X)$, where $\sigma_i(X)$ is the i-th singular values of X and $r = \min(m, n)$. The nuclear norm $\|X\|_*$ is a convex envelope of the rank of matrix X, which has been widely used in low-rank matrix recovery problems. It is well-known that the proximal operator of the nuclear norm has a closed-form solution, see, e.g., [3].

Lemma 2.1. *Let $Y \in R^{m\times n}$, the proximal operator of $\lambda\|x\|_*$ with $\lambda > 0$ is*

$$prox_{\lambda\|\cdot\|_*}(Y) = USoft(\Sigma, \lambda)V^T,$$

where $Y = U\Sigma V^T$ is a singular value decomposition of Y, and $Soft(\Sigma, \lambda)$ is the soft-thresholding operator.

The ADMM and the proximal ADMM are popular methods for solving the following constrained convex minimization problem:

$$\begin{aligned} &\min_{x,y}\ f(x)+g(y) \\ &\text{s.t.}\ \ Ax+By=b, \end{aligned} \tag{2.1}$$

where X, X_1, X_2 are real Hilbert spaces, $b \in X$, $A : X_1 \to X$ and $B : X_2 \to X$ are nonzero bounded linear operators, $f \in \Gamma_0(X_1)$ and $g \in \Gamma_0(X_2)$.

The iteration scheme of ADMM is read as

$$\begin{cases} x^{k+1} = \arg\min\limits_{x} \left\{ f(x) + \langle \lambda^k, Ax\rangle + \dfrac{\rho}{2}\|Ax+By^k-b\|^2 \right\}, \\ y^{k+1} = \arg\min\limits_{y} \left\{ g(y) + \langle \lambda^k, By\rangle + \dfrac{\rho}{2}\|Ax^{k+1}+By-b\|^2 \right\}, \\ \lambda^{k+1} = \lambda^k + \rho\left(Ax^{k+1}+By^{k+1}-b\right), \end{cases} \tag{2.2}$$

where $\rho > 0$. To get a closed-form solution of $\{x^{k+1}\}$ and $\{y^{k+1}\}$, the proximal ADMM has also received much attention, which is defined by

$$\begin{cases} x^{k+1} = \arg\min\limits_{x} \left\{ f(x) + \langle \lambda^k, Ax\rangle + \dfrac{\rho}{2}\|Ax+By^k-b\|^2 + \dfrac{1}{2}\|x-x^k\|^2_{M_1} \right\}, \\ y^{k+1} = \arg\min\limits_{y} \left\{ g(y) + \langle \lambda^k, By\rangle + \dfrac{\rho}{2}\|Ax^{k+1}+By-b\|^2 + \dfrac{1}{2}\|y-y^k\|^2_{M_2} \right\}, \\ \lambda^{k+1} = \lambda^k + \rho\big(Ax^{k+1}+By^{k+1}-b\big), \end{cases} \tag{2.3}$$

where M_1 and M_2 are self-adjoint, positive matrices. The convergence of the ADMM (2.2) and the proximal ADMM (2.3) can be found in [2,19,20] and references therein.

3. Main algorithm

In this section, we present the main algorithm for solving (1.6). Specifically, we employ the ADMM (2.2) and the proximal ADMM (2.3) to solve (1.6). First, we introduce several auxiliary variables: Let $Lx = y$, $x = z$, $x = w$, and $Kx = q$. Then, we can reformulate (1.6) as the following constrained minimization problem:

$$\begin{aligned} &\min_{x,y,z,w,q}\ \varphi(y) + \delta_C(z) + \mu\|w\|_* + \delta_{\widetilde{C}}(q) \\ &\text{s.t.}\ \ Lx = y, \quad x = z, \quad x = w, \quad Kx = q. \end{aligned} \tag{3.1}$$

To give the detail of the ADMM, we define the corresponding augmented Lagrangian function as

$$\begin{aligned} &L(x,y,z,w,q,\lambda_1,\lambda_2,\lambda_3,\lambda_4) \\ &= \varphi(y) + \langle \lambda_1, Lx-y\rangle + \frac{\rho}{2}\|Lx-y\|^2 \end{aligned}$$

$$
\begin{aligned}
&+ \delta_C(z) + \langle \lambda_2, x - z \rangle + \frac{\rho}{2}\|x - z\|^2 \\
&+ \mu\|w\|_* + \langle \lambda_3, x - w \rangle + \frac{\rho}{2}\|x - w\|^2 \\
&+ \delta_{\widetilde{C}}(q) + \langle \lambda_4, Kx - q \rangle + \frac{\rho}{2}\|Kx - q\|^2,
\end{aligned}
$$

where $\rho > 0$ is the penalty parameter and λ_1, λ_2, λ_3, and λ_4 are the Lagrangian multipliers. Then, the ADMM for solving (3.1) is given by

$$
\begin{cases}
x^{k+1} = \arg\min\limits_x L\left(x, y^k, z^k, w^k, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\right), \\
y^{k+1} = \arg\min\limits_y L\left(x^{k+1}, y, z^k, w^k, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\right), \\
z^{k+1} = \arg\min\limits_z L\left(x^{k+1}, y^{k+1}, z, w^k, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\right), \\
w^{k+1} = \arg\min\limits_w L\left(x^{k+1}, y^{k+1}, z^{k+1}, w, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\right), \\
q^{k+1} = \arg\min\limits_q L\left(x^{k+1}, y^{k+1}, z^{k+1}, w^{k+1}, q, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\right), \\
\lambda_1^{k+1} = \lambda_1^k + \rho\left(Lx^{k+1} - y^{k+1}\right), \\
\lambda_2^{k+1} = \lambda_2^k + \rho\left(x^{k+1} - z^{k+1}\right), \\
\lambda_3^{k+1} = \lambda_3^k + \rho\left(x^{k+1} - w^{k+1}\right), \\
\lambda_4^{k+1} = \lambda_4^k + \rho\left(Kx^{k+1} - q^{k+1}\right).
\end{cases}
\tag{3.2}
$$

In the following, we present how to solve the subproblems of (3.2).

(1) For the subproblem $\{x^{k+1}\}$, we consider two approaches to solve it. First, we assume the periodic boundary condition. According to the first-order optimality condition of $\{x^{k+1}\}$, we have

$$
\begin{aligned}
&L^T\lambda_1^k + \rho L^T\big(Lx^{k+1} - y^k\big) + \lambda_2^k + \rho\big(x^{k+1} - z^k\big) \\
&\quad + \lambda_3^k + \rho\big(x^{k+1} - w^k\big) + K^T\lambda_4^k + \rho K^T\big(Kx^{k+1} - q^k\big) = 0,
\end{aligned}
$$

which can be rewritten as

$$
\begin{aligned}
&\left(\rho L^T L + 2\rho I + \rho K^T K\right) x^{k+1} \\
&= L^T\big(\rho y^k - \lambda_1^k\big) + \rho z^k - \lambda_2^k + \rho w^k - \lambda_3^k + K^T\big(\rho q^k - \lambda_4^k\big).
\end{aligned}
\tag{3.3}
$$

Under the assumption of periodic boundary condition, the matrix $\rho L^T L + 2\rho I + \rho K^T K$ has a block circulant matrix with circulant blocks (BCCB) structure. Therefore, the Eq. (3.3) can be effectively solved via fast Fourier transform (FFT), i.e.,

$$
x^{k+1} = \mathcal{F}^{-1}\left(\frac{\mathcal{F}(L^T(\rho y^k - \lambda_1^k) + \rho z^k - \lambda_2^k + \rho w^k - \lambda_3^k + K^T(\rho q^k - \lambda_4^k))}{\mathcal{F}(\rho L^T L + 2\rho I + \rho K^T K)}\right), \tag{3.4}
$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ represent the Fourier transform and the inverse Fourier transform, respectively.

For the other boundary conditions, such as the zero boundary condition, reflexive boundary condition, and anti-reflexive boundary condition. We solve $\{x^{k+1}\}$ in the ADMM scheme (3.2) by borrowing the idea of the proximal ADMM (2.3) and adding a proximal term $\|x-x^k\|_{M_1}^2/2$, where $M_1 = I/\lambda - \rho L^T L - 2\rho I - \rho K^T K$ such that $\lambda < 1/(\rho\|L\|^2 + 2\rho + \rho\|K\|^2)$. That is

$$x^{k+1} = \arg\min_x L\big(x, y^k, z^k, w^k, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\big) + \frac{1}{2}\|x-x^k\|_{M_1}^2. \tag{3.5}$$

By the first-order optimality condition of (3.5) and after simple calculation, we get

$$\begin{aligned} x^{k+1} = x^k - \lambda\big(&L^T\lambda_1^k + \rho L^T(Lx^k - y^k) + \lambda_2^k + \rho(x^k - z^k) \\ &+ \lambda_3^k + \rho(x^k - w^k) + K^T\lambda_4^k + \rho K^T(Kx^k - q^k)\big). \end{aligned} \tag{3.6}$$

(2) For the subproblem $\{y^{k+1}\}$, we have

$$\begin{aligned} y^{k+1} &= \arg\min_y L\big(x^{k+1}, y, z^k, w^k, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\big) \\ &= \arg\min_y \left\{\varphi(y) + \frac{\rho}{2}\left\|Lx^{k+1} - y + \frac{1}{\rho}\lambda_1^k\right\|\right\} \\ &= prox_{\frac{1}{\rho}\varphi}\left(Lx^{k+1} + \frac{1}{\rho}\lambda_1^k\right). \end{aligned} \tag{3.7}$$

If $\varphi = \|\cdot\|_1$ or $\|\cdot\|_2$, the proximal operator of $prox_{\frac{1}{\rho}\varphi}$ has a closed-form solution. In detail, when $\varphi = \|\cdot\|_2$, we have

$$(y^{k+1})_{ij} = \begin{pmatrix} (y_x^{k+1})_{ij} \\ (y_y^{k+1})_{ij} \end{pmatrix} = \begin{pmatrix} \max(\Gamma_{ij} - 1/\rho, 0) * \dfrac{(\nabla_x x^{k+1} + \lambda_{1,x}^k/\rho)_{ij}}{\Gamma_{ij}} \\ \max(\Gamma_{ij} - 1/\rho, 0) * \dfrac{(\nabla_y x^{k+1} + \lambda_{1,y}^k/\rho)_{ij}}{\Gamma_{ij}} \end{pmatrix},$$

where

$$\Gamma_{ij} = \sqrt{\left(\nabla_x x^{k+1} + \frac{1}{\rho}\lambda_{1,x}^k\right)_{ij}^2 + \left(\nabla_y x^{k+1} + \frac{1}{\rho}\lambda_{1,y}^k\right)_{ij}^2}, \quad i = 1,\dots,m, \quad j = 1,\dots,n.$$

When $\varphi = \|\cdot\|_1$, we have

$$\begin{aligned} (y^{k+1})_{ij} &= \begin{pmatrix} (y_x^{k+1})_{ij} \\ (y_y^{k+1})_{ij} \end{pmatrix} \\ &= \begin{pmatrix} \max\left(\left|\left(\nabla_x x^{k+1} + \frac{1}{\rho}\lambda_{1,x}^k\right)_{ij}\right| - \frac{1}{\rho}, 0\right) * \operatorname{sgn}\left(\left(\nabla_x x^{k+1} + \frac{1}{\rho}\lambda_{1,x}^k\right)_{ij}\right) \\ \max\left(\left|\left(\nabla_y x^{k+1} + \frac{1}{\rho}\lambda_{1,y}^k\right)_{ij}\right| - \frac{1}{\rho}, 0\right) * \operatorname{sgn}\left(\left(\nabla_y x^{k+1} + \frac{1}{\rho}\lambda_{1,y}^k\right)_{ij}\right) \end{pmatrix}. \end{aligned}$$

(3) For the subproblem $\{z^{k+1}\}$, we have

$$\begin{aligned} z^{k+1} &= \arg\min_z L\big(x^{k+1}, y^{k+1}, z, w^k, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\big) \\ &= \arg\min_z \left\{ \delta_C(z) + \frac{\rho}{2} \left\| x^{k+1} - z + \frac{1}{\rho}\lambda_2^k \right\| \right\} \\ &= P_C\left(x^{k+1} + \frac{1}{\rho}\lambda_2^k \right), \end{aligned} \tag{3.8}$$

where P_C denotes the orthogonal projection onto the closed convex set C. Consider the definition of $C = \{x \in R^{m\times n} \mid 0 \le x_{ij} \le 1, i = 1, \dots, m, j = 1, \dots, n\}$, we have

$$z_{ij}^{k+1} = \begin{cases} 0, & \text{if} \quad \left(x^{k+1} + \frac{1}{\rho}\lambda_2^k \right)_{ij} < 0, \\ \left(x^{k+1} + \frac{1}{\rho}\lambda_2^k \right)_{ij}, & \text{if} \quad 0 \le \left(x^{k+1} + \frac{1}{\rho}\lambda_2^k \right)_{ij} \le 1, \\ 1, & \text{if} \quad \left(x^{k+1} + \frac{1}{\rho}\lambda_2^k \right)_{ij} > 1. \end{cases} \tag{3.9}$$

(4) For the subproblem $\{w^{k+1}\}$, we have

$$\begin{aligned} w^{k+1} &= \arg\min_w L\big(x^{k+1}, y^{k+1}, z^{k+1}, w, q^k, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\big) \\ &= \arg\min_w \left\{ \mu\|w\|_* + \frac{\rho}{2} \left\| x^{k+1} - w + \frac{1}{\rho}\lambda_3^k \right\| \right\} \\ &= prox_{\frac{\mu}{\rho}\|\cdot\|_*}\left(x^{k+1} + \frac{1}{\rho}\lambda_3^k \right). \end{aligned} \tag{3.10}$$

(5) For the subproblem $\{q^{k+1}\}$, we have

$$\begin{aligned} q^{k+1} &= \arg\min_q L\big(x^{k+1}, y^{k+1}, z^{k+1}, w^{k+1}, q, \lambda_1^k, \lambda_2^k, \lambda_3^k, \lambda_4^k\big) \\ &= \arg\min_q \left\{ \delta_{\widetilde{C}}(q) + \frac{\rho}{2} \left\| Kx^{k+1} - q + \frac{1}{\rho}\lambda_4^k \right\| \right\} \\ &= P_{\widetilde{C}}\left(Kx^{k+1} + \frac{1}{\rho}\lambda_4^k \right), \end{aligned} \tag{3.11}$$

where $P_{\widetilde{C}}$ denotes the orthogonal projection onto the closed convex set $\widetilde{C}$. Taking into account the definition of $\widetilde{C}$, we have

$$q_{ij}^{k+1} = \begin{cases} f_{ij}, & \text{if} \quad (i,j) \in U, \\ \left(Kx^{k+1} + \frac{1}{\rho}\lambda_4^k \right)_{ij}, & \text{if} \quad (i,j) \in I \backslash U. \end{cases} \tag{3.12}$$

In summary, the detailed ADMM for solving (1.6) is summarized in Algorithm 3.1.

Algorithm 3.1 The ADMM and the proximal ADMM for solving (1.6)

Input: For arbitrarily x^0, y^0, z^0, w^0, and q^0. Choose $\rho > 0$.
1: Update x^{k+1} by (3.4) or (3.6).
2: Update y^{k+1} by (3.7).
3: Update z^{k+1} by (3.8).
4: Update w^{k+1} by (3.10).
5: Update q^{k+1} by (3.11).
6: Update the multipliers by

$$\begin{aligned}
\lambda_1^{k+1} &= \lambda_1^k + \rho(Lx^{k+1} - y^{k+1}),\\
\lambda_2^{k+1} &= \lambda_2^k + \rho(x^{k+1} - z^{k+1}),\\
\lambda_3^{k+1} &= \lambda_3^k + \rho(x^{k+1} - w^{k+1}),\\
\lambda_4^{k+1} &= \lambda_4^k + \rho(Kx^{k+1} - q^{k+1}).
\end{aligned}$$

Stop when a given stopping criterion is met.
Output: x^{k+1}.

In the following, we briefly discuss the convergence of the proposed Algorithm 3.1. Let

$$u = \begin{pmatrix} y \\ z \\ w \\ q \end{pmatrix}, \quad A = \begin{pmatrix} L \\ I \\ I \\ K \end{pmatrix}, \quad B = \begin{pmatrix} -I & 0 & 0 & 0 \\ 0 & -I & 0 & 0 \\ 0 & 0 & -I & 0 \\ 0 & 0 & 0 & -I \end{pmatrix}.$$

Define

$$g(u) = \varphi(y) + \delta_C(z) + \mu\|w\|_* + \delta_{\widetilde{C}}(q),$$

then (3.1) can be rewritten as

$$\begin{aligned}
&\min_{x,u} \; g(u)\\
&\text{s.t.} \;\; Ax + Bu = 0.
\end{aligned}$$

Therefore, the convergence of the iterative sequences generated by Algorithm 3.1 follows directly from the classical convergence analysis of the ADMM and the proximal ADMM, respectively.

4. Numerical experiments

In this section, we will demonstrate the performance of the proposed method and compare it to other methods. We refer to the proposed method as LR_CExTV. All experiments were performed on a Laptop with an Intel Core 2 Duo 2.70 GHz and 4GB memory, running on Windows 7 and MATLAB R2014a.

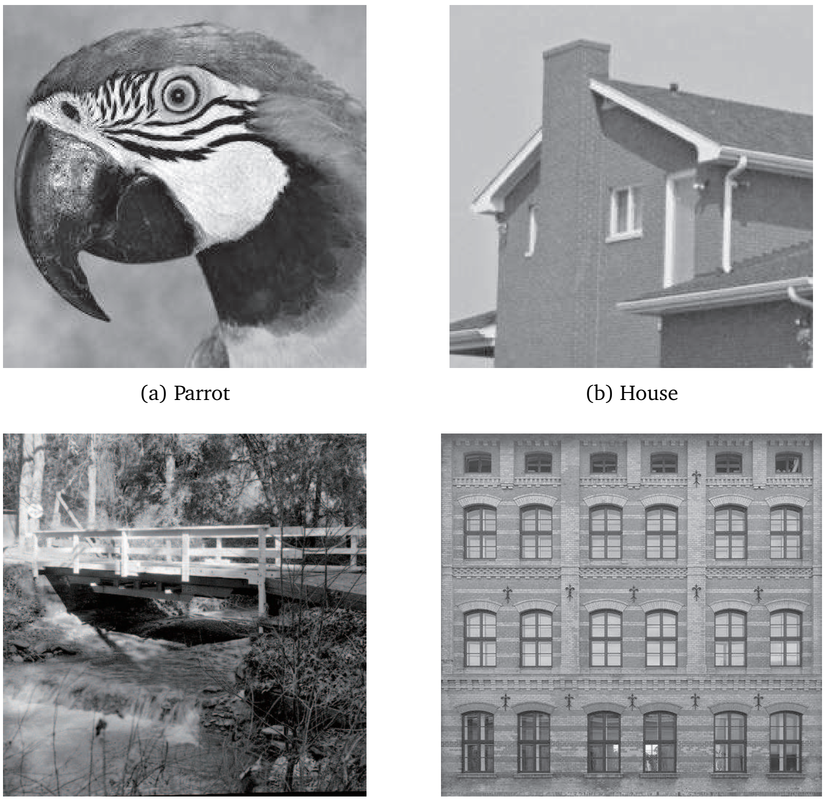
To measure the quality of the restored images, we use the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) index, which are defined by

$$PSNR = 10\log_{10}\frac{P^2}{\sum_{i,j}(x_{ij}-\widetilde{x}_{ij})^2/mn},$$
$$SSIM = \frac{(2\mu_x\mu_{\widetilde{x}}+c_1)(2\sigma_{x\widetilde{x}}+c_2)}{(2\mu_x^2\mu_{\widetilde{x}}^2+c_1)(\sigma_x^2+\sigma_{\widetilde{x}}^2+c_2)},$$

where P is the maximum peak value of the original image $x \in R^{m\times n}$, $\widetilde{x} \in R^{m\times n}$ is the restored image, $c_1 > 0$ and $c_2 > 0$ are small constants, μ_x and $\mu_{\widetilde{x}}$ are the mean values of x and $\widetilde{x}$, respectively; σ_x and $\sigma_{\widetilde{x}}$ are the variances of x and $\widetilde{x}$, respectively; $\sigma_{x\widetilde{x}}$ is the covariance of x and $\widetilde{x}$.

4.1. Experiment setting

Test images. We choose the 256-by-256 gray level image "Parrot", the 256-by-256 gray level image "House", the 512-by-512 gray level image "Bridge", and the 517-by-493 gray level image "Building" as test images, which are shown in Fig. 1.



(a) Parrot (b) House

(c) Bridge (d) Building

Figure 1: Test images.

Parameters. In our experiments, we solve a sequence of the convex minimization problem (1.6) with a varied choice of μ and record the best choice of μ that gives the highest PSNR. We set $\lambda = 0.01$ in the update of the sequences $\{x^{k+1}\}$ of (3.6). For the parameter ρ, we set it to be 14, which performs stably and efficiently throughout the experiments.

Noisy pixels detection. In the first phase, we detect the noisy pixel location and obtain the set of U in our method (1.6). Different from other two-phase methods [5, 8], we do not use an adaptive median filter to detect the noisy pixels. It is enough to set the observed pixels $f_{ij} = 0$ or 1 as salt-and-pepper noise and the rest pixels are viewed as noise-free. This approach is also used in the ℓ_0TV method [35].

Stopping criterion. The stopping criterion is defined by

$$\frac{\|x^{k+1} - x^k\|}{\|x^k\|} \leq \epsilon,$$

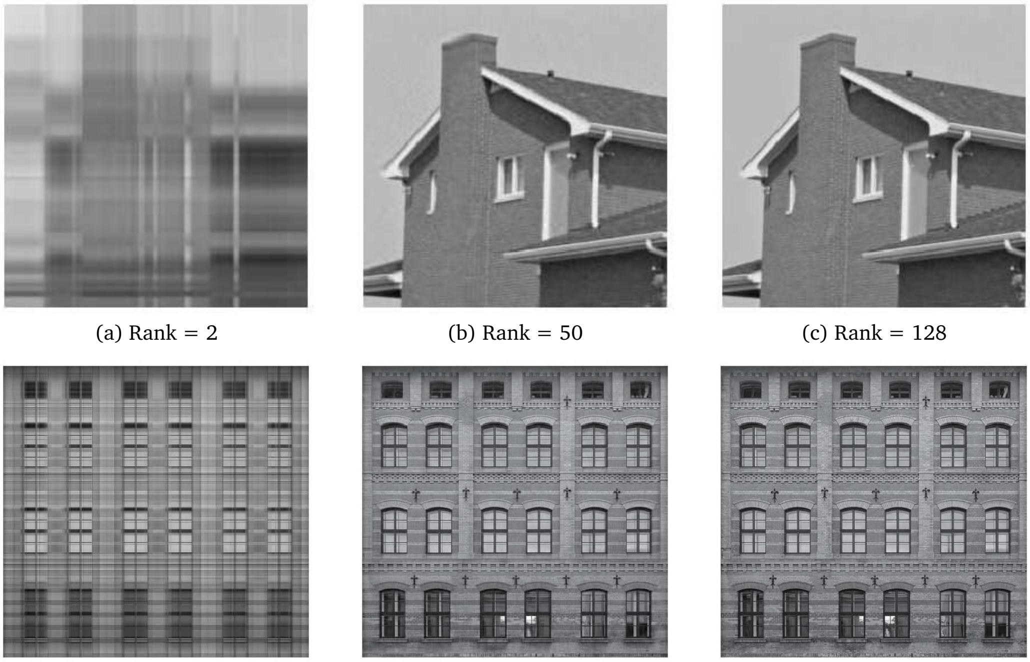
where ϵ is a given small constant. In the following experiments, we set $\epsilon = 10^{-6}$.

4.2. Numerical results and discussions

In the first experiment, we demonstrate the motivation of the proposed model (1.6), particularly with the introduction of nuclear norm regularization. We select “House” and “Building” as the test images and construct approximate images with low rank, which are shown in Fig. 2. We compare the proposed model (1.6) with the ExTV (1.5) and (1.7), referred to as CExLR. We add salt-and-pepper impulse noise at different levels to the corresponding images. The obtained results are presented in Tables 1-2. The hyphen symbol $(-)$ indicates that the maximum number of iterations 4000 was exceeded. The results from the Tables 1-2 show that the model based on nuclear norm regularization is significantly better than the other two models when the rank of the test image is very small. When the rank of the test image increases and the noise level is low, the model based on nuclear norm regularization still outperforms the other two models. When the noise level is high, the proposed model is better than both the nuclear norm regularization model and the total variation regularization model. For full-rank images, the proposed model consistently outperforms the other two models. Considering that using only nuclear norm or total variation regularization cannot fully represent the prior information of natural images, we adopt a combination of nuclear norm and total variation regularization for our proposed model.

In the second experiment, we consider salt-and-pepper denoising without blurring. To show the influence of the regularization parameter, Fig. 3 shows the PSNR against the regularization parameter μ for the test images. It can be observed from Fig. 3 that the PSNR obtained by $\mu > 0$ is always larger than $\mu = 0$. For the choice of $\mu > 0$, the PSNR values nearly keep stagnating.

We now compare the proposed method (1.6) with the ExTV method (1.5) and the CExTV method (i.e., $\mu = 0$ in (1.6)). The obtained results are presented in Table 3. It



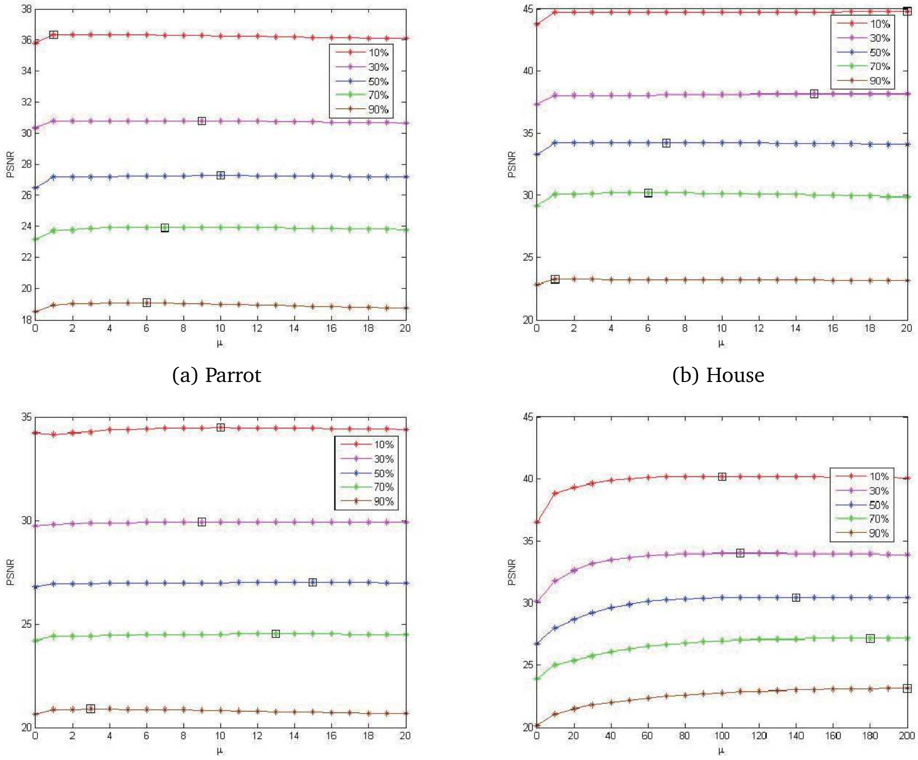
(a) Rank = 2 (b) Rank = 50 (c) Rank = 128

(d) Rank = 2 (e) Rank = 50 (f) Rank = 240

Figure 2: Low rank approximation images of "House" and "Building".

Table 1: Numerical results of different methods for "House" image with fixed rank.

Rank	Noise level	Input PSNR/SSIM	ExTV PSNR/SSIM/Iter	CExLR PSNR/SSIM/Iter	LR_CExTV (1.6) PSNR/SSIM/Iter
2	10%	15.64/0.0977	60.85/0.9996/1190	116.28/1.0000/216	66.49/0.9999/1025
	30%	10.83/0.0232	53.66/0.9980/1110	113.82/1.0000/307	92.77/1.0000/1402
	50%	8.68/0.0120	48.57/0.9946/1134	112.82/1.0000/415	90.12/1.0000/1787
	70%	7.20/0.0068	43.19/0.9842/2528	107.78/1.0000/603	84.50/1.0000/2249
	90%	6.11/0.0044	32.73/0.9167/−	104.31/ 1.0000/896	73.39/1.0000/3727
50	10%	15.42/0.1728	45.24/0.9942/954	109.93/1.0000/197	49.68/0.9969/874
	30%	10.69/0.0590	38.89/0.9765/908	95.01/1.0000/677	60.08/0.9998/2492
	50%	8.47/0.0281	34.54/0.9500/1073	34.35/0.9437/592	37.26/0.9723/1188
	70%	7.03/0.0159	30.57/0.9021/2103	26.08/0.7256/532	31.60/0.9107/898
	90%	5.92/0.0075	23.49/0.7689/−	19.68/0.3572/1456	24.18/0.7786/1081
128	10%	15.48/0.1818	43.63/0.9921/954	49.88/0.9978/489	46.54/0.9748/892
	30%	10.70/0.0595	37.93/0.9703/908	36.54/0.9543/558	38.99/0.9748/828
	50%	8.46/0.0307	33.21/0.9370/1117	30.49/0.8590/1471	34.54/0.9419/842
	70%	7.03/0.0158	29.47/0.8840/2271	23.33/0.6791/1917	30.35/0.8896/1087
	90%	5.91/0.0052	22.64/0.7421/−	19.41/0.3405/1994	23.35/0.7520/1033
Full rank	10%	15.44/0.1809	43.63/0.9917/286	43.02/0.9876/1043	44.76/0.9926/463
	30%	10.70/0.0591	37.29/0.9685/510	35.47/0.9425/718	38.15/0.9711/409
	50%	8.46/0.0309	33.27/0.9360/1043	30.20/0.8483/474	34.21/0.9388/377
	70%	7.00/0.0155	29.17/0.8808/1986	25.16/0.6674/1281	30.17/0.8870/449
	90%	5.90/0.0063	22.79/0.7416/−	19.24/0.3156/−	23.22/0.7495/1772



(a) Parrot (b) House

(c) Bridge (d) Building

Figure 3: The PSNR values with respect to the regularization μ for the test images corrupted by salt-and-pepper noise with different noise levels. The best PSNR is marked by the square ($\square$).

can be seen from Table 3 that the CExTV method is slightly better than the ExTV method without box constraint in most cases. This confirms the advantage of incorporating information about the pixel values. We see that the proposed method outperforms the other two methods. In particular, for the test images of "Building", the proposed method achieves 3 dB higher than the other two methods. The proposed method requires fewer iterations than the other two methods, especially when the noise level is above 30%. For a noise level of 90%, the proposed method significantly reduces the number of iterations compared to the other methods. Fig. 4 presents the computation time of the proposed algorithm for the test images of "Parrot" and "Bridge". Fig. 5 shows the restored images of "Building" from salt-and-pepper noise with noise levels of 70% and 90%. It can be seen from Fig. 5 that the proposed method significantly outperforms the other two methods visually, especially when the noise level is 90%.

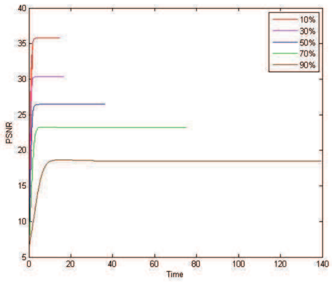
In the third experiment, we report numerical results for restoring blurred images with salt-and-pepper noise. We consider Gaussian blur with a size of 7×7 and standard derivation 5. We use the same way in the first experiment to obtain the noise-free set U. To show the influence of the regularization parameter on the restoration results, Fig. 6 shows the PSNR values to the regularization parameter μ for the test images.

Table 2: Numerical results of different methods for "Building" image with fixed rank.

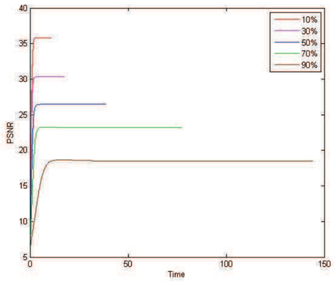
Rank	Noise level	Input PSNR/SSIM	ExTV PSNR/SSIM/Iter	CExLR PSNR/SSIM/Iter	LR_CExTV (1.6) PSNR/SSIM/Iter
2	10%	15.72/0.2434	41.30/0.9920/968	116.91/1.0000/216	72.67/1.0000/998
	30%	10.95/0.0769	34.86/0.9645/948	114.51/1.0000/326	96.08/1.0000/1159
	50%	8.74/0.0368	31.16/0.9193/1227	112.59/1.0000/449	93.61/1.0000/1331
	70%	7.27/0.0180	28.00/0.8334/2437	109.97/1.0000/682	89.68/1.0000/1791
	90%	6.18/0.0073	23.49/0.5921/–	104.57/ 1.0000/1292	42.49/ 0.9919/2744
50	10%	15.64/0.3399	37.89/0.9884/835	76.09/1.0000/203	74.88/1.0000/713
	30%	10.84/0.1231	31.35/0.9491/897	74.05/1.0000/418	71.83/1.0000/1025
	50%	8.60/0.0646	27.64/0.8817/1218	53.79/0.9999/871	50.99/0.9997/2522
	70%	7.15/0.0316	24.59/0.7672/3721	33.36/0.9652/2013	33.40/0.9655/3322
	90%	6.06/0.0108	20.52/0.4797/–	23.71/0.7028/1393	24.49/0.7367/1234
240	10%	15.62/0.3604	36.55/0.9854/805	50.43/0.9993/1260	47.19/0.9984/1392
	30%	10.81/0.1330	30.18/0.9381/870	35.01/0.9703/1944	35.54/0.9751/1281
	50%	8.25/0.0542	26.72/0.8617/1551	30.15/0.9114/1486	30.86/0.9277/1013
	70%	6.79/0.0257	23.93/0.7344/3194	26.66/0.8165/1304	27.39/0.8434/917
	90%	5.70/0.0096	20.06/0.4408/–	22.41/0.6244/1214	23.12/0.6538/919
Full rank	10%	15.60/0.3627	36.45/0.9850/359	39.53/0.9887/1881	40.19/0.9913/829
	30%	10.86/0.1361	30.10/0.9364/756	33.28/0.9546/650	34.00/0.9643/760
	50%	8.63/0.0668	26.74/0.8602/1398	29.70/0.8999/558	30.45/0.9192/783
	70%	7.15/0.0312	23.90/0.7315/3396	26.40/0.8068/492	27.13/0.8349/831
	90%	6.06/0.0109	20.15/0.4434/–	22.39/0.6231/502	23.12/0.6501/790

Table 3: The PSNR (dB), SSIM, and number of iterations (Iter) of different methods for images corrupted by salt-and-pepper noise.

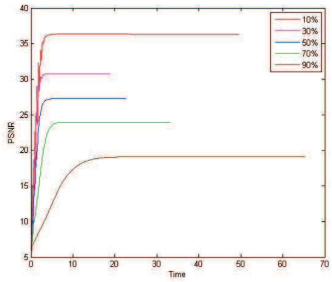
Image	Noise level	Input PSNR/SSIM	ExTV PSNR/SSIM/Iter	CExTV PSNR/SSIM/Iter	LR_CExTV (1.6) PSNR/SSIM/Iter
Parrot	10%	14.92/0.2522	35.75/0.9905/439	35.76/0.9905/316	36.30/0.9902/939
	30%	10.18/0.1037	30.34/0.9636/519	30.35/0.9637/542	30.77/0.9636/353
	50%	7.98/0.0574	26.48/0.9213/1084	26.48/0.9213/1086	27.24/0.9229/390
	70%	6.46/0.0295	23.17/0.8507/2181	23.17/0.8507/2181	23.93/0.8553/561
	90%	5.39/0.0102	18.50/0.6759/–	18.50/ 0.6759/–	19.08/ 0.6821/1072
House	10%	15.44/0.1809	43.63/0.9917/286	43.75/0.9917/301	44.76/0.9926/463
	30%	10.70/0.0591	37.29/0.9685/510	37.32/0.9685/510	38.15/0.9711/409
	50%	8.46/0.0309	33.27/0.9360/1043	33.27/0.9360/1043	34.21/0.9388/377
	70%	7.00/0.0155	29.17/0.8808/1986	29.17/0.8808/1986	30.17/0.8870/449
	90%	5.90/0.0063	22.79/0.7416/–	22.79/0.7416/–	23.22/0.7495/1772
Bridge	10%	15.25/0.3315	34.21/0.9768/2220	34.21/0.9768/2220	34.45/0.9759/394
	30%	10.48/0.1124	29.71/0.9267/2124	29.71/0.9267/2124	29.92/0.9258/400
	50%	8.25/0.0542	26.77/0.8496/2899	26.77/0.8495/2899	26.97/0.8498/430
	70%	6.79/0.0257	24.18/0.7207/–	24.18/0.7207/–	24.49/0.7240/399
	90%	5.70/0.0096	20.65/0.4535/–	20.65/0.4535/–	20.88/0.4582/1179
Building	10%	15.60/0.3627	36.45/0.9850/359	36.49/0.9851/352	40.19/0.9913/829
	30%	10.86/0.1361	30.10/0.9364/756	30.11/0.9366/756	34.00/0.9643/760
	50%	8.63/0.0668	26.74/0.8602/1398	26.74/0.8603/1398	30.45/0.9192/783
	70%	7.15/0.0312	23.90/0.7315/3396	23.90/0.7315/3396	27.13/0.8349/831
	90%	6.06/0.0109	20.15/0.4434/–	20.15/0.4434/–	23.12/0.6501/790



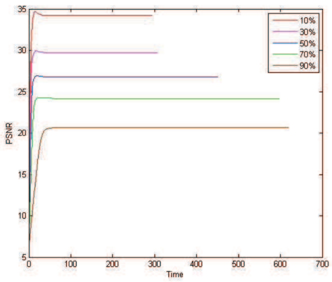
(a) ExTV



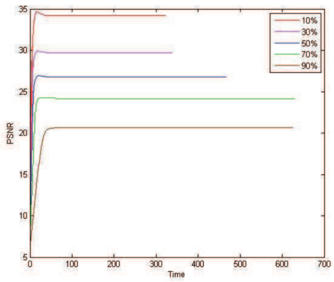
(b) CExTV



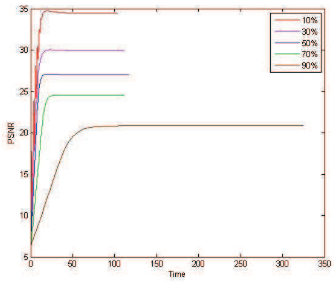
(c) LR_CExTV



(d) ExTV

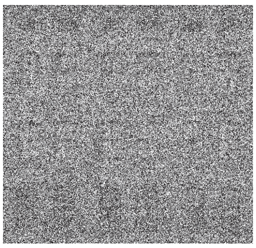


(e) CExTV



(f) LR_CExTV

Figure 4: The PSNR values with respect to the CPU time in seconds are compared for the different methods. (a)-(c) The test image is "Parrot"; (d)-(f) The test image is "Bridge".

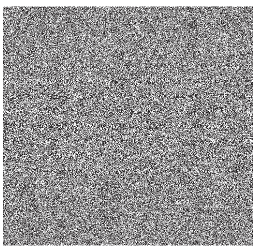


(a) 70%/7.15

(b) ExTV/23.90

(c) CExTV/23.90

(d) LR_CExTV/27.13



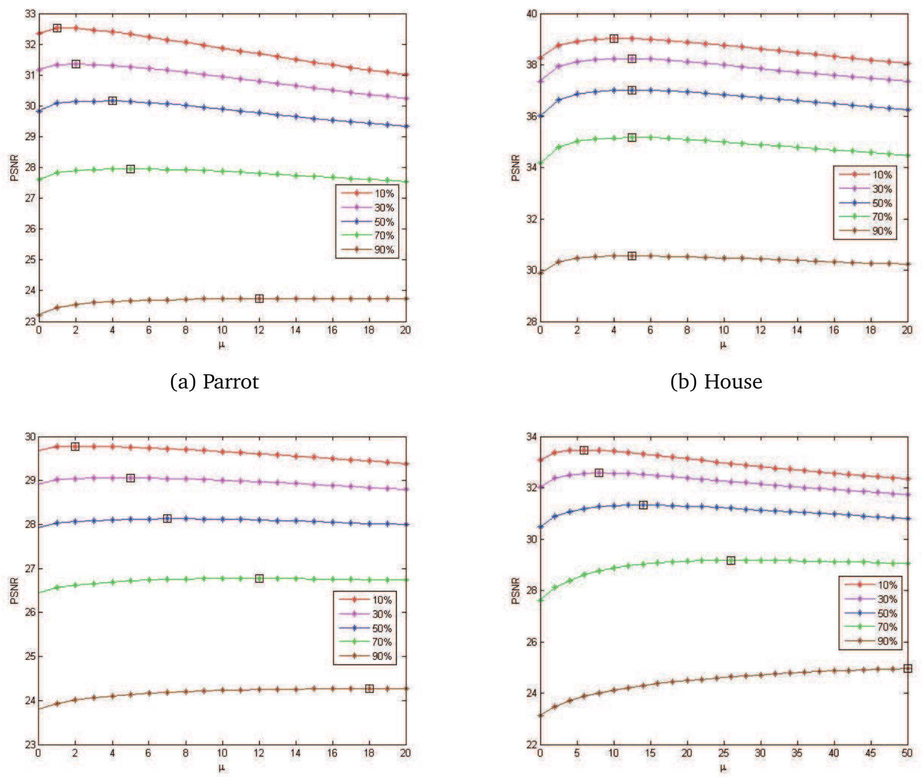
(e) 90%/6.06

(f) ExTV/20.15

(g) CExTV/20.15

(h) LR_CExTV/23.12

Figure 5: Restored images (with PSNR (dB)) of different methods. First column: noisy image "Building" with salt-and-pepper noise at 70% and 90%. Second column: restored images by the ExTV method. Third column: restored images by the CExTV method. Forth column: restored images by the LR_CExTV method.



(a) Parrot (b) House

(c) Bridge (d) Building

Figure 6: The PSNR values with respect to the regularization μ for the test images corrupted by Gaussian blur and salt-and-pepper noise with different noise levels. The best PSNR is marked by the square ($\square$).

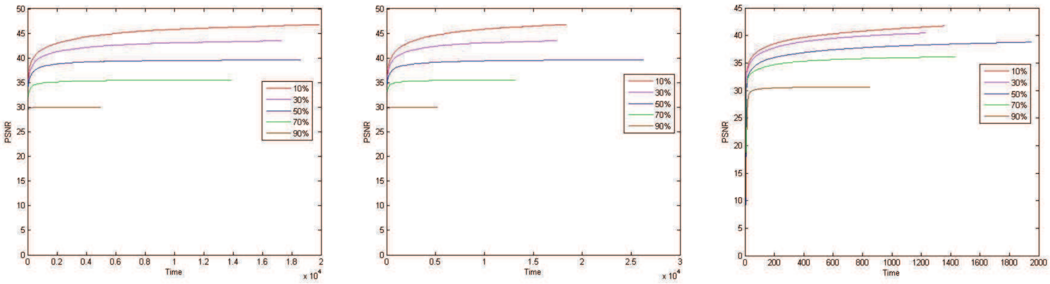
It is well-known that the ADMM usually exhibits slow convergence when reaching high precision solutions. In Table 4, we report numerical results of the compared methods when the maximum iteration number of 4000 is reached. It can be seen from Table 4 that the proposed LR_CExTV method outperforms the other two methods in terms of PSNR and SSIM values. However, the proposed LR_CExTV method requires more CPU time than the other two methods due to its involvement of the proximal operator of the nuclear norm. Furthermore, we present the number of iterations where the stopping criterion reached in Table 5. The symbol " $-$ " means that the maximum number of iterations of 4×10^5 is exceeded. It can be observed from Tables 4 and 5 that the proposed algorithm can quickly converge to a solution with low accuracy. For the noise levels below 50%, when the number of iterations exceeds 4×10^5, the solution quality will improve more and more slowly. Therefore, more iteration numbers are required to achieve higher precision solutions. For case of large noise (e.g., 90%), the quality of the images recovered by the two different stopping criteria is almost the same. This shows that the proposed algorithm could be stopped early for the restoration of high noisy images.

Table 4: The PSNR (dB), SSIM, and CPU time (in seconds) of different methods for images corrupted by Gaussian blur and salt-and-pepper noise when the iteration number is fixed by 4000.

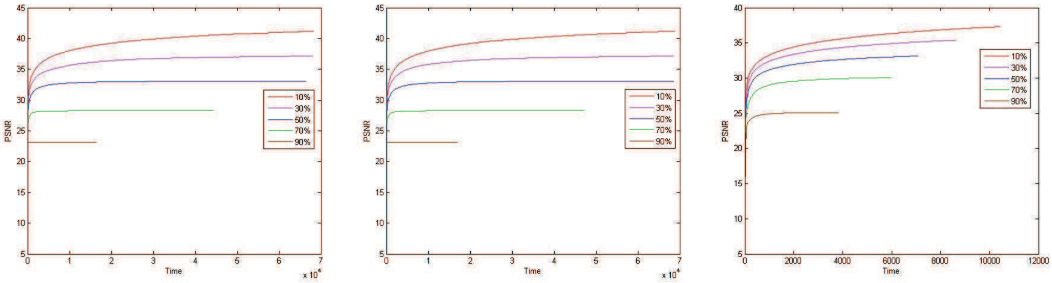
Image	Noise level	Input PSNR/SSIM	ExTV PSNR/SSIM/Time	CExTV PSNR/SSIM/Time	LR_CExTV (1.6) PSNR/SSIM/Time
Parrot	10%	14.05/0.1319	32.31/0.9436/157.3	32.34/0.9436/173.2	32.51/0.9447/271.7
	30%	9.94/0.0484	31.10/0.9338/178.4	31.17/0.9340/178.1	31.34/0.9356/277.6
	50%	7.85/0.0221	29.79/0.9195/181.5	29.83/0.9198/266.2	30.15/0.9221/412.8
	70%	6.42/0.0122	27.39/0.8907/255.0	27.60/0.8915/200.7	27.95/0.8955/301.7
	90%	5.40/0.0072	23.09/0.7965/192.7	23.21/0.7977/198.4	23.75/ 0.8071/302.8
House	10%	15.17/0.1103	38.27/0.9507/152.3	38.27/0.9506/162.6	39.02/0.9540/256.5
	30%	10.64/0.0322	37.36/0.9423/166.1	37.36/0.9424/262.1	38.24 /0.9467/342.1
	50%	8.43/0.0173	36.00/0.9278/189.0	36.00/0.9278/169.2	37.01/0.9332/263.6
	70%	6.97/0.0096	34.17/0.9027/181.4	34.17/0.9027/181.7	35.16/0.9098/339.8
	90%	5.89/0.0050	29.88/0.8433/231.6	29.88/0.8433/204.7	30.55/0.8503/266.0
Bridge	10%	14.51/0.1123	29.66/0.8931/625.0	29.67/0.8936/727.9	29.77/0.8953/1316.1
	30%	10.25/0.0329	28.90/0.8741/685.3	28.90/0.8745/713.1	29.05/0.8769/1186.0
	50%	8.18/0.0178	27.92/0.8435/682.9	27.93/0.8442/709.6	28.12/0.8476/1174.3
	70%	6.75/0.0084	26.43/0.7820/655.2	26.44/0.7829/673.5	26.77/0.7907/1144.1
	90%	5.68/0.0048	23.79/0.6063/654.6	23.80/0.6072/680.1	24.26/0.6299/1148.0
Building	10%	14.64/0.1091	33.06/0.9418/580.4	33.07/0.9418/689.2	33.47/0.9452/1141.6
	30%	10.58/0.0369	32.03/0.9284/755.4	32.03/0.9284/710.4	32.57/0.9342/1111.8
	50%	8.50/0.0182	30.47/0.9028/629.8	30.47/0.9028/683.5	31.33/0.9149/1219.1
	70%	7.10 /0.0106	27.62/0.8354/661.6	27.62/0.8354/952.6	29.17/0.8689/1175.9
	90%	6.05/0.0056	23.12/0.6106/655.0	23.12/0.6106/689.4	24.96/0.6985/1276.6

Table 5: The PSNR (dB), SSIM, and number of iterations (Iter) of different methods for images corrupted by Gaussian blur and salt-and-pepper noise when the stopping criterion $\epsilon = 1 \times 10^{-6}$.

Image	Noise level	Input PSNR/SSIM	ExTV PSNR/SSIM/Iter	CExTV PSNR/SSIM/Iter	LR_CExTV (1.6) PSNR/SSIM/Iter
Parrot	10%	14.05/0.1319	41.52/0.9901/−	41.59/0.9901/−	37.81/0.9796/66912
	30%	9.94/0.0484	35.92/0.9766/−	36.09/0.9768/−	34.64/0.9669/51335
	50%	7.85/0.0221	31.89/0.9521/−	32.06/0.9525/−	31.90/0.9462/34079
	70%	6.42/0.0122	28.06/0.9100/252736	28.26/0.9110/256216	28.47/0.9092/22000
	90%	5.40/0.0072	23.09/0.7994/88476	23.21/0.8006/89310	23.77/ 0.8094/8851
House	10%	15.17/0.1103	46.79/0.9907/−	46.80/0.9907/−	41.72/0.9736/19472
	30%	10.64/0.0322	43.48/0.9828/−	43.48/0.9828/−	40.43/0.9665/18145
	50%	8.43/0.0173	39.60/0.9647/−	39.60/0.9647/−	38.77/0.9537/17845
	70%	6.97/0.0096	35.47/0.9255/237823	35.47/0.9255/237850	36.11/0.9256/17174
	90%	5.89/0.0050	29.96/0.8464/91789	29.96/0.8464/92691	30.61/0.8529/10153
Bridge	10%	14.51/0.1123	36.98/0.9791/−	37.06/0.9796/−	34.80/0.9660/91393
	30%	10.25/0.0329	32.75/0.9493/−	32.84/0.9505/−	31.96/0.9380/60055
	50%	8.18/0.0178	29.59/0.8987/−	29.63/0.9000/−	29.58/0.8952/42575
	70%	6.75/0.0084	26.94/0.8119/279872	26.96/0.8132/282316	27.26/0.8186/25476
	90%	5.68/0.0048	23.85/0.6125/95199	23.86/0.6134/95270	24.29/0.6348/10741
Building	10%	14.64/0.1091	41.19/0.9896/−	41.19/0.9896/−	37.30/0.9755/35846
	30%	10.58/0.0369	37.14/0.9759/−	37.14/0.9759/−	35.39/0.9638/29942
	50%	8.50/0.0182	33.08/0.9448/−	33.08/0.9449/−	33.13/0.9423/24076
	70%	7.10 /0.0106	28.29/0.8593/269037	28.29/0.8594/268189	30.06/0.8925/19606
	90%	6.05/0.0056	23.14/0.6125/97312	23.14/0.6124/96660	25.07/0.7084/12192



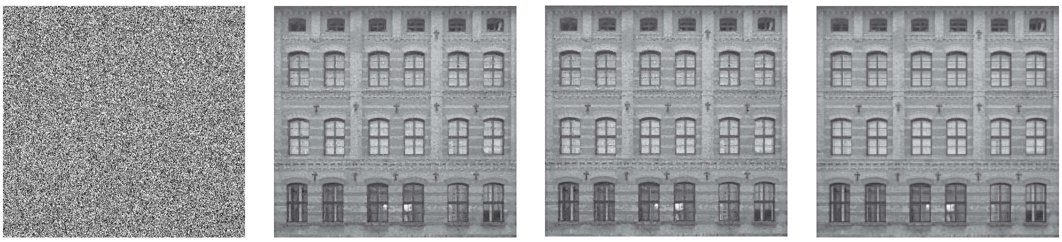
(a) ExTV (b) CExTV (c) LR_CExTV



(d) ExTV (e) CExTV (f) LR_CExTV

Figure 7: The PSNR values with respect to the CPU time in seconds for the compared methods. (a)-(c) The test image is "House"; (d)-(f) The test image is "Building".

(a) 70%/7.10 (b) ExTV/28.29 (c) CExTV/28.29 (d) LR_CExTV/30.06



(e) 90%/6.05 (f) ExTV/23.14 (g) CExTV/23.14 (h) LR_CExTV/25.07

Figure 8: Restored images (with PSNR (dB)) of different methods. First column: noisy image "Building" with Gaussian blur and salt-and-pepper noise 70% and 90%. Second column: restored images by the ExTV method. Third column: restored images by the CExTV method. Forth column: restored images by the LR_CExTV method.

Compared with the ExTV and CExTV, LR_CExTV can save the number of iterations when the same stopping criterion is satisfied. Fig. 7 shows the PSNR versus CPU time in seconds for the compared methods. It can be seen from Fig. 7 that the proposed LR_CExTV method takes less CPU time than the other two methods, especially when the noise level is high. To visually show the restored images, Fig. 8 presents the recovered "Building" images from blurring and salt-and-pepper noise. It can be observed from Fig. 8 that the proposed method outperforms the other two methods in terms of details in recovering image quality.

Remark 4.1. We did not compare the proposed model with other impulse noise models, especially several nonconvex models, such as Nonconvex TV [39], TVSCAD [22], Nonconvex [18], ℓ_0TV [25, 35], and TV-Log [38], among others. The main reason is that these models heavily rely on the selection of regularization parameters and other factors. To replicate the results of this paper, the code is available at `https://github.com/hhaaoo1331/LRCExTV`.

5. Conclusions

In this paper, we proposed a new two-phase method involved with low rank, total variation and box constraint for image deblurring with impulse noise. In the first phase, the noise pixels are detected by prior knowledge of the impulse noise. Then, in the second phase, we employ the ADMM and the proximal ADMM to solve a constrained minimization problem to get a clean image. Compared with existing algorithms, the obtained iterative algorithm has a simple structure, which is easy to be implemented. Numerical experiments show that the solutions with low precision can be obtained quickly. For a high accurate solution, it needs to spend more iteration numbers and computing time. Therefore, how to speed up the proposed algorithm to solve the problem (1.6) is a question worthy of further study.

Acknowledgments

We would like to thank the editor and reviewers for their constructive comments and suggestions, which have greatly improved the quality of this manuscript.

This work was funded by the National Natural Science Foundations of China (Grant Nos. 12061045,12031003,12271117), by the Jiangxi Provincial Natural Science Foundation (Grant No. 20224ACB211004), and by the basic research joint funding project of university and Guangzhou City (Grant No. 202102010434).

References

[1] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, 2017.

[2] S. BOYD, N. PARIKH, E. CHU, B. PELEATO, AND J. ECKSTEIN, *Distrituted optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn. 3 (2010), 1–122.

[3] J. F. CAI, E. J. CANDES, AND Z. SHEN, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim. 20 (2010), 1956–1982.

[4] J. F. CAI, R. H. CHAN, AND C. DI FIORE, *Minimization of a detail-preserving regularization functional for impulse noise removal*, J. Math. Imaging Vis. 29 (2007), 79–91.

[5] J. F. CAI, R. H. CHAN, AND M. NIKOLOVA, *Fast two-phase image deblurring under impulse noise*, J. Math. Imaging Vision 36 (2010), 46.

[6] A. CHAMBOLLE AND T. POCK, *A first-order primal-dual algorithm for convex problems with applications to imaging*, J. Math. Imaging Vision 40(1) (2011), 120–145.

[7] R. H. CHAN AND K. CHEN, *Fast multilevel algorithm for a minimization problem in impulse noise removal*, SIAM J. Sci. Comput. 30(3) (2008), 1474–1489.

[8] R. H. CHAN, Y. Q. DONG, AND M. HINTERMÜLLER, *An efficient two-phase l1-tv method for restoring blurred images with impulse noise*, IEEE Trans. Image Process. 19(7) (2010), 1731–1739.

[9] R. H. CHAN, C. W. HO, AND M. NIKOLOVA, *Convergence of Newton's method for a minimization problem in impulse noise removal*, J. Comput. Math. 22(2) (2004), 168–177.

[10] R. H. CHAN, C. W. HO, AND M. NIKOLOVA, *Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization*, IEEE Trans. Image Process. 14(19) (2005), 1479–1485.

[11] R. H. CHAN, C. HU, AND M. NIKOLOVA, *An iterative procedure for removing random-valued impulse noise*, IEEE Signal Process. Lett. 11(12) (2004), 921–924.

[12] 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(1) (2013), 680–697.

[13] T. F. CHAN AND S. ESEDOGLU, *Aspects of total variation regularized l1 function approximation*, SIAM J. Appl. Math. 65(5) (2005), 1817–1837.

[14] K. CHANG, X. Y. ZHANG, P. DING, B. X. LI, *Data-adaptive low-rank modeling and external gradient prior for single image super-resolution*, Signal Process. 161 (2019), 36–49.

[15] P. Y. CHEN AND C. Y. LIEN, *An efficient edge-preserving algorithm for removal of salt-and-pepper noise*, IEEE Signal Process. Lett. 15 (2008), 833–836.

[16] S. S. CHEN AND X. YANG, *Two-stage method for impulse-noise removal*, Opt. Eng. 46(9) (2007), 097003.

[17] S. S. CHEN, X. YANG, AND G. CAO, *Impulse noise suppression with an augmentation of ordered difference noise detector and an adaptive variational method*, Pattern Recognit. Lett. 30 (2009), 460–467.

[18] Z. X. CUI AND Q. B. FAN, *A "nonconvex+nonconvex" approach for image restoration with impulse noise removal*, Appl. Math. Modell. 62 (2018), 254–271.

[19] E. ESSER, X. ZHANG, AND T. CHAN, *A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science*, SIAM J. Imaging Sci. 3(4) (2010), 1015–1046.

[20] E. X. FANG, B. S. HE, H. LIU, AND X. M. YUAN, *Generalized alternating direction method of multipliers: New theoretical insights and applications*, Math. Program. Comput. 7(2) (2015), 149–187.

[21] H. Y. 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 J. Sci. Comput. 27(6) (2006), 1881–1902.

[22] G. GU, S. JIANG, AND J. YANG, *A tvscad approach for image deblurring with impulsive noise*, Inverse Probl. 33(12) (2017), 125008.

[23] X. X. GUO, F. LI, AND M. K. NG, *A fast $\ell 1$-tv algorithm for image restoration*, SIAM J. Sci. Comput. 31(3) (2009), 2322–2341.

[24] W. Q. HUANG, Z. W. KE, Z. X. CUI, J. CHENG, Z. L. QIU, S. JIA, L. YING, Y. J. ZHU, AND D. LIANG, *Deep low-rank plus sparse network for dynamic mr imaging*, Med. Image Anal. 73 (2021), 102190.

[25] M. M. KANG, M. J. KANG, AND M. Y. JUNG, *Sparse representation based image delburing model under random-valued impulse noise*, Multidimens. Systems Signal Process. 30 (2019), 1063–1092.

[26] S. K. KAYHAN, *An effective 2-stage method for removing impulse noise in images*, J. Visual Commun. Image Represent. 25(2) (2014), 478–486.

[27] S. F. LIANG, S. M. LU, J. Y. CHANG, AND C. T. LIN, *A novel two-stage impulse noise removal technique based on neural newtorks and fuzzy decision*, IEEE Trans. Fuzzy Syst. 16(4) (2008), 863–873.

[28] C. LIN, Y. C. LI, S. L. FENG, AND M. X. HUANG, *A two-stage algorithm for the detection and removal of random-valued impulse noise based on local similarity*, IEEE Access 8 (2020), 222001–222012.

[29] L. Y. MA, M. K. NG, J. YU, AND T. Y. ZENG, *Efficient box-constrained tv-type-l1 algorithms for restoring images with impulse noise*, J. Comput. Math. 31(3) (2013), 249–270.

[30] L. Y. MA, J. YU, AND T. Y. ZENG, *Sparse representation prior and total variation-based image deblurring under impulse noise*, SIAM J. Imaging Sci. 6(4) (2013), 2258–2284.

[31] F. SCIACCHITANO, Y. Q. DONG, AND M. S. ANDERSEN, *Total variation based parameter-free model for impulse noise removal*, Numer. Math. Theor. Meth. Appl. 10(1) (2017), 186–204.

[32] F. SHI, J. CHENG, L. WANG, P. T. YAP, AND D. G. SHEN, *Lrtv: Mr image super-resolution with low-rank and total variation regularization*, IEEE Tran. Med. Imaging 34(12) (2015), 2459–2466.

[33] X. WANG, Y. S. REN, AND W. S. ZHANG, *Depression disorder classification of fmri data using sparse low-rank functional brain network and graph-based features*, Comput. Math. Methods Med. 2017 (2017), 3609821.

[34] C. L. WU, J. Y. ZHANG, AND X. C. TAI, *Augmented Lagrangian method for total variation restoration with non-quadratic fidelity*, Inverse Problems and Imaging 5 (2011), 237–261.

[35] G. Z. YUAN AND B. GHANEM, *$\ell_0$tv: A sparse optimization method for impulse noise image restoration*, IEEE Trans. Pattern Anal. Mach. Intell. 41(2) (2019), 352–364.

[36] J. YUAN, J. SHI, AND X. C. TAI, *A convex and exact approach to discrete constrained tv-l1 image approximation*, East Asian J. Appl. Math. 1(2) (2011), 172 –186.

[37] C. ZENG, C. L. WU, AND R. JIA, *Non-Lipschitz models for image restoration with impulse noise removal*, SIAM J. Imaging Sci. 12(1) (2019), 420–458.

[38] B. X. ZHANG, G. P. ZHU, AND Z. B. ZHU, *A tv-log nonconvex approach for image deblurring with impulsive noise*, Signal Process. 174(12) (2020), 107631.

[39] X. J. ZHANG, M. R. BAI, AND M. K. NG, *Nonconvex-tv based image restoration with impulse noise removal*, SIAM J. Imaging Sci. 10(3) (2017), 1627–1667.