Inverse Problems and Imaging

doi:10.3934/ipi.2025021

EFFICIENT ALGORITHMS FOR MIXED NOISE REMOVAL VIA NONLOCAL LOW-RANK REGULARIZATION

Ashley Prater-Bennette*[1], Minghao Rostami[1,2], Lixin Shen[3], Erin E. Tripp[4] and Jiangyu Yu[3]

[1]Air Force Research Laboratory, Rome, NY 13441, USA

[2]SUNY at Binghamton, Binghamton, NY 13902, USA

[3]Syracuse University, Syracuse, NY 13244, USA

[4]Hamilton College, Clinton, NY 13323, USA

(Communicated by Weihong Guo)

Abstract. This paper addresses an image denoising problem where observed images are contaminated by a mix of Gaussian and impulse noise. The proposed approach involves solving an optimization problem with an objective function that combines a convex content-dependent fidelity term and a nonlocal low-rank regularization term. Both terms are constructed using patch matrices formed from similar patches of the image. Based on the unique properties of these terms, we propose two different strategies to efficiently solve the problem. We also provide convergence analysis for both methods. Our numerical experiments demonstrate that the proposed two-phase approach performs well in terms of three quantitative metrics: peak signal-to-noise ratio (PSNR), structural similarity (SSIM), and feature similarity (FSIM), as well as in the visual quality of the restored images.

1. Introduction.

Mathematically, a noisy image affected by both additive Gaussian white noise and impulse noise can be expressed as:

$$y = \mathcal{N}_{imp}(x + z), \tag{1}$$

where x is the true image in $\mathbb{R}^{n_0\times m_0}$, y is the observed image, z is the additive Gaussian white noise, $\mathcal{N}_{imp} : \mathbb{R}^{n_0\times m_0} \to \mathbb{R}^{n_0\times m_0}$ characterizes impulse noise. The impulse noise assigns specific pixels of $x + z$ within its dynamic range $[0, 255]$ as outliers. Those pixels are considered anomalies in the observed data y, and the probability of outliers is referred to as the noise level of impulse noise in y. Outliers are modeled either as salt-and-pepper or random-valued impulse noise. In the case of salt-and-pepper noise, the outliers take on the values 0 or 255, while other pixels remain unchanged. For random-valued impulse noise, the outliers assume uniformly

2020 *Mathematics Subject Classification.* Primary: 68U10, 94A08, 90C26; Secondary: 15A03, 46N10, 65F22.

Key words and phrases. Impulse noise removal, low-rank regularization, proximal difference-of-convex algorithm, forward-backward splitting algorithm.

The work of Rostami was supported in part by the National Science Foundation (NSF) under grants DMS-1818833 and DMS-2146191 (CAREER). The work of L. Shen was supported in part by the National Science Foundation under grant DMS-2208385.

*Corresponding author: Lixin Shen.

distributed random numbers in the interval $[0, 255]$, and the other pixels remain unaltered.

Due to the challenges associated with modeling the distributions of mixed additive Gaussian white and impulse noise, numerous existing approaches adopt a two-phase strategy. In this framework, the initial phase is dedicated to detecting and labeling the locations of outliers based on the nature of the impulse noise, while the subsequent phase focuses on concurrently eliminating noise at the unlabeled locations and approximating pixels at the labeled locations. The first phase of this two-phase approach typically employs a median type filter to pinpoint the locations of outliers, although the specifics of the second phase vary across different methodologies.

For instance, in [6, 7, 5], the second phase primarily involves denoising the noisy image from outlier-free data and restoring pixels in the positions labeled as outliers using a variational model. In [19], the second phase denoises the noisy image by solving an optimization problem with an objective function that includes a content-dependent fidelity term and a regularization term formed by the ℓ_1 norm of tight framelet coefficients of the underlying image. Meanwhile, in [26], the second phase recovers the image through dictionary learning on the outlier-free pixels. An alternative approach, as seen in [17], foregoes explicit impulse noise detection. Instead, it encodes each noise-corrupted patch over a pre-learned dictionary to simultaneously remove mixed noise using a soft impulse pixel detection strategy. In [15], a method based on Laplacian scale mixture modeling and nonlocal low-rank regularization is proposed.

Recently, a nonlocal low-rank regularized two-phase (NLR-TP) approach was introduced for removing mixed impulse and additive Gaussian white noise [27]. Similar to the methodology in [6, 7], the approach employs a nonlinear filter - either the adaptive median filter (AMF) [16] or the adaptive center-weighted median filter (ACWMF) [10]-to detect impulse noise locations in the initial phase. In the second phase, the algorithm restores the image using only the unlabeled observed data by solving an optimization problem. The objective function consists of a convex content-dependent fidelity term and a nonconvex regularization term, both derived from patch matrices formed by similar patches. The regularization term incorporates the log-sum function and the singular values of patch matrices. To solve the optimization problem, a majorization-minimization approach is employed. At each iteration, the non-quadratic convex fidelity term is approximated by its second-order Taylor expansion, while the nonconvex regularization term is approximated by its first-order Taylor expansion, both based on the previous iteration.

Building on the NLR-TP framework, this paper proposes distinct algorithms for solving the optimization problem by leveraging the structural properties of both the convex content-dependent fidelity term and the nonlocal low-rank regularization term. Specifically, the convex fidelity term is differentiable with a Lipschitz continuous gradient map, and the proximity operator of the regularization term has an explicit form using the proximity operator of the log-sum function [23]. With these properties, the forward-backward splitting algorithm [2] can be employed for solving the optimization problem. Upon closer examination of the log-sum function within the regularization term, two different approaches for solving the optimization problem emerge. In the first approach, the log-sum function is decomposed as the difference of a scaled absolute function with a differentiable convex function that is the difference of the log-sum function with the scaled absolute function.

In the second approach, the log-sum function is expressed as the difference of a convex function, which is the sum of the log-sum function with a scaled quadratic function. Both ways of decomposing the log-sum function allow for the adoption of the proximal difference-of-convex algorithm with extrapolation [25] for efficiently implementing the optimization problem. While both decomposition approaches are mathematically suitable for the problem, numerically, the first approach proves effective, whereas the second approach does not. This intriguing observation will be further explained in the paper. Additionally, numerical demonstrations showcase that the proposed algorithms for removing mixed noise outperform those in [27].

The contributions of this paper are summarized in the following two aspects. First, this paper introduces novel algorithms for solving the optimization problem within the NLR-TP framework, specifically addressing mixed Gaussian and impulse noise. The differentiability of the convex fidelity term, along with the explicit form of the proximity operator for the regularization term, enables the utilization of the forward-backward splitting algorithm. Second, we provide an in-depth analysis of the log-sum function within the regularization term reveals two decomposition approaches. Both decomposition strategies permit the application of the proximal difference-of-convex algorithm with extrapolation for efficient optimization problem implementation.

The subsequent sections of this paper are organized as follows. In Section 2, we provide a concise review of the NLR-TP approach, with particular attention to the formulation of the optimization problem employed in this approach. The properties of this optimization problem are discussed in Section 3. Two different schemes for solving this optimization problem are proposed in Section 4, and the complete procedure for solving the proposed optimization problem for mixed noise removal is given in Section 5. Numerical experiments of the proposed algorithm are shown in Section 6, and we draw our conclusions in Section 7.

2. **Problem formulation.** In this section, we articulate the optimization problem for the removal of mixed Gaussian and impulse noise, building upon the formulation employed in the nonlocal low-rank regularized two-phase (NLR-TP) approach introduced in [27]. The NLR-TP approach encompasses two phases: (i) the detection of outlier candidate locations and (ii) the resolution of an optimization problem featuring a content-driven fidelity term and a nonlocal low-rank regularization term.

Let us assume that the original image x in model (1) is indexed by $\Omega = \{1,2,\ldots,n_0\}\times\{1,2,\ldots,m_0\}$, and the outlier candidate set $\mathcal{Z}$ is defined as follows:

$$\mathcal{Z} = \begin{cases} \{(i,j)\in\Omega : y_{ij} \neq x_{ij} + z_{ij} \text{ and } y_{ij} \in \{0,255\}\}, & \text{for salt-and-pepper noise;} \\ \{(i,j)\in\Omega : y_{ij} \neq x_{ij} + z_{ij}\}, & \text{for random-valued impulse noise.} \end{cases}$$

The impulse noise level is characterized by the ratio of the cardinalities of $\mathcal{Z}$ and Ω.

In the first phase of the NLR-TP, the set $\mathcal{Z}$ is identified using the AMF for salt-and-pepper noise and the ACWMF for random-valued impulse noise.

In the second phase of the NLR-TP, the objective is to approximate the ideal image x using the outlier-free data on $\Omega\setminus\mathcal{Z}$. To achieve this, we represent the image x in a patch format. For an image x of size $n_0 \times m_0$, a block-matching algorithm [11] is employed to find patches $\{x_{\ell_k}\}_{k=1}^{m-1}$ that are similar to a square patch of size $\sqrt{n}\times\sqrt{n}$ centered at position (i,j). These patches form an $n\times m$ matrix X_ℓ, termed the ℓ-th patch matrix of the image x:

$$X_\ell := \begin{bmatrix} x_\ell & x_{\ell 1} & \ldots & x_{\ell m-1} \end{bmatrix},$$

where x_ℓ and $x_{\ell k}$ are the vectorizations of themselves. We reasonably assume that X_ℓ is low-rank.

Now, let $X \in \mathbb{R}^{n\times m}$ be a patch matrix of the image x. We form accordingly the matrices Y and Z from the corresponding locations of y and z in model (1), respectively. Then,

$$Y = \mathcal{N}_{imp}(X+Z), \tag{2}$$

$\mathcal{N}_{imp} : \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ represents impulse noise which assigns some pixels of $X+Z$ in its dynamic range $[0, 255]$. Here, we did not distinguish the operator $\mathcal{N}_{imp}$ used in between (1) and (2). Moreover, we also use Ω and $\mathcal{Z}$ to represent the index set of X and the outlier candidate set, respectively.

To set up our denosing model, let us review some definitions and establish notation. Throughout, we consider real matrices and vectors, and, for $x \in \mathbb{R}^d$, use $\langle\cdot,\cdot\rangle$ and $\|\cdot\|$ to refer to the usual inner product and its induced norm. For $p \ge 1$, let $\|M\|_p = (\sum_{i=1}^n \sum_{j=1}^m |M_{ij}|^p)^{1/p}$ denote the ℓ_p-norm of the matrix M of size $n\times m$ by viewing it as a long vector in $\mathbb{R}^{nm}$, where M_{ij} is the (i,j)th entry of M. In particular, $\|M\|_2$ is the Frobenious norm of M. For two matrices A and B of the same dimension, the Hadamard product $A\odot B$ is a matrix of the same dimension as operants, with elements given by $(A\odot B)_{ij} = A_{ij}B_{ij}$, $(i,j)\in\Omega$. For the sets Ω and $\mathcal{Z}$ associated with model (2), we define an $n\times m$ binary matrix D as follows: $D_{ij} = 1$ if $(i,j)\in\Omega\setminus\mathcal{Z}$ and 0 otherwise. We define $m\wedge n := \min\{m,n\}$.

For $A, B \in \mathbb{R}^{n\times m}$, $\langle A, B\rangle := \sum_{i=1}^n \sum_{j=1}^m A_{ij}B_{ij}$. For any vector $x \in \mathbb{R}^{n\wedge m}$, let $\mathrm{Diag}(x)$ denote the $n\times m$ matrix with $(\mathrm{Diag}(x))_{ii} = x_i$ for all i, and $(\mathrm{Diag}(x))_{ij} = 0$ for $i\neq j$. For any $X\in\mathbb{R}^{n\times m}$, we define $\sigma(X) := (\sigma_1(X), \sigma_2(X), \ldots, \sigma_{n\wedge m}(X))^\top$, where $\sigma_1(X) \ge \sigma_2(X) \ge \ldots \ge \sigma_{m\wedge n}(X)$ are the ordered singular values of X. Denote by $\mathcal{O}(X)$ the set of all pairs (U,V):

$$\mathcal{O}(X) := \left\{(U,V)\in\mathbb{R}^{n\times m\wedge n}\times\mathbb{R}^{m\times m\wedge n} : U^\top U = I, V^\top V = I, X = U\mathrm{Diag}(\sigma(X))V^\top\right\}.$$

That is, for any pair $(U,V)\in\mathcal{O}(X)$, $U\mathrm{Diag}(\sigma(X))V^\top$ is a singular value decomposition of X.

Our denoising model is formulated as follows:

$$\min\left\{J(X) = F(X) + \lambda G(X) : X \in \mathbb{R}^{n\times m}\right\}, \tag{3}$$

where $F(X)$ is the fidelity term, $G(X)$ is the regularization term, and λ is a positive parameter that balances the two. We will introduce both F and G in the following.

To define the fidelity term, we introduce the function $\varphi_\eta : \mathbb{R}\to\mathbb{R}$, as proposed in [19]:

$$\varphi_\eta(t) := \frac{\eta t^2}{\eta + |t|} \tag{4}$$

where η is a given parameter that controls the shape of φ_η. This function behaves like the function t^2 when t is small, making it suitable for Gaussian noise, and like $\eta|t|$ when t is large, making it effective for impulse noise. Importantly, φ_η is convex, differentiable, and has a Lipschitz continuous gradient with a Lipschitz constant of 2, i.e.,

$$|\varphi'_\eta(s) - \varphi'_\eta(t)| \le 2|s-t|, \quad \forall s,t. \tag{5}$$

With the function, we define the fidelity term as

$$F(X) = \|\varphi_\eta(D\odot(X-Y))\|_1.$$

Here, $\varphi_\eta(D \odot (X - Y))$ is viewed as an $n \times m$ matrix whose (i, j)th entry is given by

$$[\varphi_\eta(D \odot (X - Y))]_{ij} = \varphi_\eta(D_{ij}(X_{ij} - Y_{ij})).$$

From an application perspective, the squared loss is well-suited for Gaussian noise, while the ℓ_1-loss is more robust for impulse noise. Since real-world images often suffer from a mixture of both noise types, we require a loss function that combines the strengths of both. From a mathematical perspective, squared loss is highly sensitive to outliers, while the ℓ_1-loss is robust but non-differentiable at zero. The function φ_η offers a smooth transition between these two behaviors while maintaining a 2-Lipschitz continuous gradient, facilitating the use of proximal gradient algorithms for optimization. The parameter η controls the trade-off between smoothness and robustness, and we have observed in numerical experiments that proper tuning of η significantly improves performance.

Additionally, the gradient of the fidelity term will be frequently used in our computations:

$$\nabla F(X) = D \odot \varphi'_\eta(D \odot (X - Y)).$$

For the regularization term, we employ low-rank regularization and define it as

$$G_\epsilon(X) = \sum_{i=1}^{m \wedge n} g_\epsilon(\sigma_i(X)),$$

where for a given positive number ϵ, the function g_ϵ is defined as

$$g_\epsilon(t) = \log\left(\frac{|t|}{\epsilon} + 1\right). \tag{6}$$

This function is widely used in compressive sensing [8] as a bridge between the ℓ_0 quasi norm and the ℓ_1 norm. It also serves as a nonconvex surrogate for the matrix rank function in low-rank regularization problems [12]. Compared to standard nuclear norm regularization, the log-sum function better approximates the properties of the rank function. The nuclear norm tends to overly shrink singular values, which can degrade low-rank recovery performance. In contrast, the log-sum function is smoother and avoids imposing excessive penalties on large singular values, thereby preserving more structural information in the data. Although other nonconvex regularization methods, such as the Schatten-p norm and ℓ_p norm, also promote rank sparsity, the log-sum function has a closed-form proximal operator [23]. This property makes it computationally more efficient and convenient for algorithmic implementation.

With these preparations, the optimization problem in the second phase of the NLR-TP is formulated as

$$\min\left\{J(X) = \|\varphi_\eta(D \odot (X - Y))\|_1 + \lambda \sum_{i=1}^{m \wedge n} g_\epsilon(\sigma_i(X)) : X \in \mathbb{R}^{n \times m}\right\}. \tag{7}$$

In the next section, we will analyze the properties of the objective function of problem (7).

3. Properties.

This section consists of two subsections. The first subsection discusses the Kurdyka-Lojasiewicz property of the objective function of (7), while the other provides the proximity operator associated with the regularization term of (7).

3.1. **The Kurdyka-Łojasiewicz property of the objective function.** In this subsection we will show that the objective function of (7) is a Kurdyka-Łojasiewicz function [1, 2].

Given a function $f : \mathbb{R}^d \to (-\infty, +\infty]$, the domain of f is $\mathrm{dom} f = \{x \in \mathbb{R}^d : f(x) < +\infty\}$. The subdifferential of f is defined as

$$\partial f(x) = \left\{\eta : \lim\inf_{u \to x} \frac{f(u) - f(x) - \langle \eta, u - x\rangle}{\|u - x\|} \geq 0\right\}.$$

When f is convex, this reduces to the usual convex subdifferential, and when f is differentiable, $\partial f(x) = \{\nabla f(x)\}$.

The definition of a Kurdyka-Łojasiewicz function is as follows.

Definition 3.1 (Kurdyka-Łojasiewicz Property)**.** Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lower semicontinuous.

(a) The function f is said to have the Kurdyka-Łojasiewicz (KL) property at $\tilde{x} \in \mathrm{dom}\partial f := \{x \in \mathbb{R}^d : \partial f(x) \neq \emptyset\}$ if there exist $\beta \in (0, +\infty]$, a neighborhood U of $\tilde{x}$ and a continuous function $\psi : [0, \beta) \to [0, \infty)$ such that
   (i) $\psi(0) = 0$;
   (ii) ψ is continuously differentiable on $(0, \beta)$, and continuous at 0;
   (iii) for all $s \in (0, \beta)$, $\psi'(s) > 0$;
   (iv) for all $x \in U \cap \{x \in \mathbb{R}^d : f(\tilde{x}) < f(x) < f(\tilde{x}) + \beta\}$, the following Kurdyka-Łojasiewicz inequality holds

$$\psi'(f(x) - f(\tilde{x}))\mathrm{dist}(0, \partial f(x)) \geq 1.$$

(b) The function f is called a KL function if f has the KL property at each point of $\mathrm{dom}\partial f$.

Semialgebraic and subanalytic functions are typical KL functions [1, 2]. We first recall the definition of semialgebraic functions.

Definition 3.2 (Semialgebraic)**.**

(a) A subset S of $\mathbb{R}^d$ is a real semialgebraic set if there exists a finite number of real polynomial functions $P_{ij}, Q_{ij} : \mathbb{R}^d \to \mathbb{R}$ such that

$$S = \cup_{j=1}^{p} \cap_{i=1}^{q} \{x \in \mathbb{R}^d : P_{ij}(x) = 0, Q_{ij}(x) < 0\}.$$

(b) A function $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is called semialgebraic if its graph $\{(x, \xi) \in \mathbb{R}^{d+1} : f(x) = \xi\}$ is a semialgebraic subset of $\mathbb{R}^{d+1}$.

Lemma 3.3. *Let $\sigma_k : \mathbb{R}^{n\times m} \to \mathbb{R} : X \mapsto \sigma_k(X)$ be the kth singular value of $X \in \mathbb{R}^{n\times m}$, where $k = 1, 2, \ldots n \wedge m$. Then each σ_k is semialgebraic.*

Proof. We consider sequentially the following optimization problems

$$\max_{(p,q)\in C_k} \langle Xp, q\rangle \qquad (\mathrm{P_k})$$

for $k = 1, \ldots, n \wedge m$, where

$$C_k = \left\{(p, q) \in \mathbb{R}^m \times \mathbb{R}^n : \begin{array}{l} \|p\|^2 - 1 = 0, \ \|q\|^2 - 1 = 0, \\ \langle v_i, p\rangle = 0, \ \langle u_i, q\rangle = 0, \ i = 1, \ldots, k - 1 \end{array}\right\}$$

with (v_i, u_i) being an optimal solution to problem $(\mathrm{P_i})$, $i < k$. It is well-known (see, e.g., [14]) that $\sigma_k(X)$ is the optimal value of problem $(\mathrm{P_k})$, i.e., there exist $u_k \in \mathbb{R}^n$

and $v_k \in \mathbb{R}^m$ such that

$$\sigma_k(X) = \max_{(p,q)\in C_k} \langle Xp, q\rangle = \langle Xv_k, u_k\rangle.$$

Clearly, C_k is a semialgebraic set in $\mathbb{R}^m \times \mathbb{R}^n$ and $J(X,p,q) := \langle Xp, q\rangle$ is a semialgebraic function on $\mathbb{R}^{n\times m} \times (\mathbb{R}^m \times \mathbb{R}^n)$ by the definitions of semialgebraic set and semialgebraic function. Hence, σ_k is a semialgebraic function in $\mathbb{R}^{n\times m}$. □

The class of semialgebraic sets is a special subclass of subanalytic sets. We recall the definitions of subanalytic set and subanalytic function.

Definition 3.4 (semianalytic)**.**

(a) A subset A of $\mathbb{R}^n$ is called semianalytic if each point of $\mathbb{R}^n$ admits a neighborhood V for which $V \cap A$ assumes the following form:

$$\cup_{i=1}^{p} \cap_{j=1}^{q} \{x \in V : f_{ij}(x) = 0, g_{ij}(x) > 0\},$$

where the functions $f_{ij}, g_{ij} : V \to \mathbb{R}$ are real-analytic for all $1 \le i \le p$, $1 \le j \le q$.

(b) The set A is called subanalytic if each point of $\mathbb{R}^n$ admits a neighborhood V such that

$$A \cap V = \{x \in \mathbb{R}^n : \exists y \in R^m \text{such that } (x,y) \in B\},$$

where B is a bounded semianalytic subset of $\mathbb{R}^n \times \mathbb{R}^m$ for some $m \ge 1$.

(c) A function $f : \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ is called subanalytic if its graph is a subanalytic subset of $\mathbb{R}^n \times \mathbb{R}$.

Lemma 3.5. *The function g_ϵ given in (6) has the KL property at each point in $\mathbb{R}$.*

Proof. Note that $\text{dom}\partial g_\epsilon = \mathbb{R}$. Since g_ϵ is differentiable on $\mathbb{R} \setminus \{0\}$, then g_ϵ has the KL property at each point of $\mathbb{R} \setminus \{0\}$. At the origin, the function g_ϵ is not differentiable. For any $\beta > 0$ we have

$$\{t \in \mathbb{R} : g_\epsilon(0) < g_\epsilon(t) < g_\epsilon(0) + \beta\} = (-\epsilon(e^\beta - 1), 0) \cup (0, \epsilon(e^\beta - 1)).$$

Let us choose $\psi(t) = ct$ with $c \ge \epsilon e^\beta$. Then,

$$\psi'(g_\epsilon(t) - g_\epsilon(0))\text{dist}(0, g'_\epsilon(t)) = \frac{c}{\epsilon + |t|} \ge 1$$

for all $t \in (-\epsilon(e^\beta - 1), 0) \cup (0, \epsilon(e^\beta - 1))$. This implied that g_ϵ has the KL property at the origin, hence has the KL property at each point in $\mathbb{R}$. □

Lemma 3.6. *The function g_ϵ given in (6) is subanalytic in $\mathbb{R}$.*

Proof. Let us look at the graph of g_ϵ.

$$\begin{aligned} \text{graph}(g_\epsilon) &= \{(t,y) \in \mathbb{R} \times \mathbb{R} : \log\left(\frac{|t|}{\epsilon} + 1\right) = y,\ y \ge 0\} \\ &= \{(t,y) \in \mathbb{R} \times \mathbb{R} : \epsilon + |t| - \epsilon e^y = 0,\ y \ge 0\}. \end{aligned}$$

The last set can be written as the union of the sets $\{(0,0)\}$, $\{(t,y) \in \mathbb{R} \times \mathbb{R} : \epsilon - t - \epsilon e^y = 0,\ y > 0,\ t < 0\}$, and $\{(t,y) \in \mathbb{R} \times \mathbb{R} : \epsilon + t - \epsilon e^y = 0,\ y > 0,\ t > 0\}$, which are semianalytic. Therefore, g_ϵ is subanalytic in $\mathbb{R}$. □

Immediately, we can show that the fidelity term of problem (7) is semialgebraic.

Lemma 3.7. *Let us denote $f(X)$ the fidelity term of problem (7), i,e., $f(X) = \|\phi_\eta(D \odot (X - Y))\|_1$. Then f is a semialgebraic function on $\mathbb{R}^{n\times m}$.*

Proof. For $X \in \mathbb{R}^{n\times m}$, we have

$$f(X) = \sum_{i=1}^{n}\sum_{j=1}^{m}\phi_\eta(D_{ij}(X_{ij}-Y_{ij})) = \sum_{D_{ij}=1}\phi_\eta(X_{ij}-Y_{ij}).$$

From a general result concerning the summation of semialgebraic functions, it suffices to show that for a given $y \in \mathbb{R}$, $\phi_\eta(\cdot - y)$ is semialgebraic. Its graph can be written as

$$\{(t,\xi)\in\mathbb{R}^2 : \phi_\eta(t-y)=\xi\} = \{(t,\xi)\in\mathbb{R}^2 : \eta(t-y)^2 = \xi(\eta+|t-y|)\}.$$

Considering three cases of $t=y$, $t>y$, and $t<y$, the set on the right of the above equation is the union of the sets $\{(y,0)\}$, $\{(t,\xi)\in\mathbb{R}^2 : \eta(t-y)^2-\xi(\eta+t-y)=0, y-t<0\}$, and $\{(t,\xi)\in\mathbb{R}^2 : \eta(t-y)^2-\xi(\eta-t+y)=0, t-y<0\}$ accordingly. Each of the last three sets is semialgebraic by the definition of semialgebraic. Hence the graph of f is semialgebraic. This completes the proof. □

The main result in this subsection is summarized in the following theorem.

Theorem 3.8. *The objective function of the optimization problem* (7) *is a KL function.*

Proof. By Lemma 3.6 and Lemma 3.3, the low-rank regularization term $\sum_{i=1}^{m\wedge n} g_\epsilon(\sigma_i(X))$ in problem (7) as a function of X is subanalytic. Furthermore, by Lemma 3.7, the objective function of problem (7) is subanalytic. Therefore, the sum of them is subanalytic, and thus a KL function. As a result, the objective function of the optimization problem (7) is a KL function. □

3.2. The proximity operators associated with the regularization term.

In this subsection, we will present the proximity operators of the functions associated with the regularization term of (7), which are essential in the development of algorithms for solving problem (7).

We begin by introducing our notation. Let $\mathbb{H}$ be a Hilbert space equipped with an inner product $\langle\cdot,\cdot\rangle_{\mathbb{H}}$ and the induced norm $\|\cdot\|_{\mathbb{H}}$. A function p defined on $\mathbb{H}$ with values in $\mathbb{R}\cup\{+\infty\}$ is proper if its domain $\mathrm{dom}(p)=\{x\in\mathbb{H} : p(x)<+\infty\}$ is nonempty, and p is lower semicontinuous if its epigraph is a closed set. The set of proper and lower semicontinuous functions on $\mathbb{H}$ to $\mathbb{R}\cup\{+\infty\}$ is denoted by $\Gamma(\mathbb{H})$. The set of proper, convex, and lower semicontinuous functions on $\mathbb{H}$ to $\mathbb{R}\cup\{+\infty\}$ is denoted by $\Gamma_0(\mathbb{H})$.

The proximity operator was introduced by Moreau in [22]. For a function $p\in\Gamma(\mathbb{H})$, the proximity operator of p at $z\in\mathbb{H}$ with index α is defined by

$$\mathrm{prox}_{\alpha p}(z) := \arg\min\left\{p(w)+\frac{1}{2\alpha}\|w-z\|_{\mathbb{H}}^2 : w\in\mathbb{H}\right\}.$$

The proximity operator of p is a set-valued operator from $\mathbb{H}\to 2^{\mathbb{H}}$, the power set of $\mathbb{H}$.

Our goal here is to present the proximity operators of the functions associated with the regularization term of (7) that involves singular values of a matrix. As the singular values of a matrix are descending, we expect that they are still descending after applying a proximity operator to them. We need the following result.

Lemma 3.9. *Let $f:\mathbb{R}\to(-\infty,\infty]$ be a proper closed function, and assume that for a given $\lambda>0$ the following condition is satisfied: the function $u\mapsto f(u)+\frac{1}{2\lambda}(u-x)^2$ is coercive for any $x\in\mathrm{dom}(f)$ the domain of f. Then, for any $x,y\in\mathrm{dom}(f)$ with*

$x < y$, we have $u \leq v$ for any $u \in \mathrm{prox}_{\lambda f}(x)$ and $v \in \mathrm{prox}_{\lambda f}(y)$. Furthermore, assume that $0 \in \mathrm{prox}_{\lambda f}(0)$, then for any $u \in \mathrm{prox}_{\lambda f}(x)$ with $x \in \mathrm{dom}(f)$, if both x and u are nonzero, then they must the same signs.

Proof. First, for any $x \in \mathrm{dom}(f)$, it follows by [3, Theorem 6.4] that the set $\mathrm{prox}_{\lambda f}(x)$ is nonempty. Next, by the definition of proximity operator, one has for any $x, y \in \mathrm{dom}(f)$ with $x < y$

$$\begin{aligned} f(u) + \frac{1}{2\lambda}(u-x)^2 &\leq f(v) + \frac{1}{2\lambda}(v-x)^2 \\ f(v) + \frac{1}{2\lambda}(v-y)^2 &\leq f(u) + \frac{1}{2\lambda}(u-y)^2 \end{aligned}$$

for any $u \in \mathrm{prox}_{\lambda f}(x)$ and $v \in \mathrm{prox}_{\lambda f}(y)$. Adding the above two inequalities leads to $(u-v)(x-y) \geq 0$, that is, $u \leq v$.

Finally, under the assumption of $0 \in \mathrm{prox}_{\lambda f}(0)$, it follows immediately from the above results that x and u have the same signs if they are nonzero. □

The primary outcome of this subsection is presented below, accompanied by its proof, which closely follows the methodology outlined in [9, Theorem 2.3]. For the sake of completeness, we provide the proof in the subsequent text.

Theorem 3.10. *Let $f : \mathbb{R} \to (-\infty, \infty]$ be a proper closed function, and assume that for a given $\lambda > 0$ the following condition is satisfied: the function $u \mapsto f(u) + \frac{1}{2\lambda}(u-x)^2$ is coercive for any $x \in \mathrm{dom}(f)$. Let us further assume that $0 \in \mathrm{prox}_{\lambda f}(0)$. Define*

$$F : \mathbb{R}^{n\times m} \ni X \mapsto \sum_{i=1}^{n\wedge m} f(\sigma_i(X)). \tag{8}$$

Then, if $X^\star \in \mathrm{prox}_{\lambda F}(X)$ for $X \in \mathbb{R}^{n\times m}$, then there exist $(U, V) \in \mathcal{O}(X)$ and a vector $d \in \mathbb{R}^{n\wedge m}$ with $d_i \in \mathrm{prox}_{\lambda f}(\sigma_i(X))$ such that

$$X^\star = U\mathrm{Diag}(d)V^\top. \tag{9}$$

Here, if $\sigma_k(X) = \sigma_{k+1}(X)$ for some $1 \leq k \leq n \wedge m$, we always choose $d_k = d_{k+1}$.

Proof. It follows from Lemma 3.9 that $d_i \geq d_{i+1}$ for any $d_i \in \mathrm{prox}_{\lambda f}(\sigma_i(X))$ and $d_{i+1} \in \mathrm{prox}_{\lambda f}(\sigma_{i+1}(X))$ for all $i = 1, 2, \ldots, (n \wedge m) - 1$. Hence $d \in \mathbb{R}^{m\wedge n}_{\downarrow} := \{x \in \mathbb{R}^{m\wedge n} : x_1 \geq x_2 \geq \ldots \geq x_{m\wedge n} \geq 0\}$.

Next, for $X \in \mathbb{R}^{n\times m}$, if $X^\star \in \mathrm{prox}_{\lambda F}(X)$, we show that it can be rewritten in the form of (9). By the definition of the proximity operator of $\mathrm{prox}_{\lambda F}(X)$, we consider the optimization problem

$$\min\left\{\frac{1}{2\lambda}\|W - X\|_2^2 + F(W) : W \in \mathbb{R}^{n\times m}\right\}$$

which, by using (8) and introducing an auxiliary variable $d = \sigma(W)$, can be equivalently written as

$$\min_{d\in\mathbb{R}^{m\wedge n}_{\downarrow}}\left\{\min_{W\in\mathbb{R}^{n\times m},\sigma(W)=d}\left\{\frac{1}{2\lambda}\|W - X\|_2^2 + \sum_{i=1}^{m\wedge n} f(d_i)\right\}\right\}.$$

Note that $\|W - X\|_2^2 = \mathrm{tr}(W^\top W) - 2\mathrm{tr}(W^\top X) + \mathrm{tr}(X^\top X)$, $\mathrm{tr}(W^\top W) = \sum_{i=1}^{n\wedge m} d_i^2$, and $\mathrm{tr}(X^\top X) = \sum_{i=1}^{n\wedge m} \sigma_i(X)^2$. Due to von Neumann's trace inequality (see [21]), we have

$$\|W - X\|_2^2 \geq \sum_{i=1}^{n\wedge m} d_i^2 - 2\sigma(X)^\top d + \sum_{i=1}^{m\wedge n} \sigma_i(X)^2.$$

Equality holds when W admits the singular value decomposition $W = U\mathrm{Diag}(d)V^\top$, where $(U, V) \in \mathcal{O}(X)$. Under this situation, the above optimization problem reduces to

$$\min_{d\in\mathbb{R}_\downarrow^{m\wedge n}} \left\{ \frac{1}{2\lambda}\|d - \sigma(X)\|_2^2 + \sum_{i=1}^{m\wedge n} f(d_i) \right\}.$$

Clearly, if a vector d is a solution to the about problem, then $d_i \in \mathrm{prox}_{\lambda f}(\sigma_i(X))$. This completes the proof. □

As a consequences of Theorem 3.10, we present the results for f being the absolute function and g_ϵ, respectively.

Corollary 3.11. *Let F be given by (8) with $f = |\cdot|$. For any $X \in \mathbb{R}^{m\wedge n}$ and $\lambda > 0$, if $X^\star \in \mathrm{prox}_{\lambda F}(X)$, then $X^\star = U\mathrm{Diag}(d)V^\top$ for some $(U, V) \in \mathcal{O}(X)$ and $d_i = \max\{0, \sigma_i(X) - \lambda\}$, $i = 1, 2, \ldots, n \wedge m$.*

Proof. It can be checked directly that for any $\lambda > 0$ and $x \in \mathbb{R}$, the function $u \mapsto |u| + \frac{1}{2\lambda}(u - x)^2$ is coercive, $\mathrm{prox}_{\lambda|\cdot|}(x) = sgn(x)\max\{0, |x| - \lambda\}$. The result follows from Theorem 3.10. □

We remark that $F(X)$ in Corollary 3.11 is the nuclear norm of X and the corresponding $\mathrm{prox}_{\lambda F}$ is called the singular value shrinkage operator in [4].

To close this subsection, we present the proximity operators of g_ϵ and the regularization term of (7).

For any given pair of positive parameters (λ, ϵ), let us define $r_{(\lambda,\epsilon)} : \mathbb{R} \to \mathbb{R}$ as follows:

$$r_{(\lambda,\epsilon)}(t) := \frac{1}{2}(t - \epsilon) + \sqrt{\frac{1}{4}(t + \epsilon)^2 - \lambda}. \tag{10}$$

where $t \geq \max\{2\sqrt{\lambda} - \epsilon, 0\}$.Then, it was given in [23] that

$$\mathrm{prox}_{\lambda g_\epsilon}(t) = \begin{cases} \{0\}, & \text{if } |t| < t_*; \\ \{0, \mathrm{sgn}(t)r_{(\lambda,\epsilon)}(t_*)\}, & \text{if } |t| = t_*; \\ \{\mathrm{sgn}(t)r_{(\lambda,\epsilon)}(|t|)\}, & \text{if } |t| > t_*, \end{cases} \tag{11}$$

where $t_* = \frac{\lambda}{\epsilon}$ if $\sqrt{\lambda} \leq \epsilon$ and t^* is the only root of the function

$$w \mapsto \frac{1}{2\lambda}(r_{(\lambda,\epsilon)}(w) - w)^2 + g_\epsilon(r_{(\lambda,\epsilon)}(w)) - \frac{1}{2\lambda}w^2$$

for $w \in [2\sqrt{\lambda} - \epsilon, \frac{\lambda}{\epsilon})$ if $\sqrt{\lambda} > \epsilon$. This root can be found by the bisection method.

With this result, we have the following corollary.

Corollary 3.12. *Let $G_\epsilon : \mathbb{R}^{n\times m} \to \mathbb{R}$ be given as follows:*

$$G_\epsilon(X) := \sum_{i=1}^{m\wedge n} g_\epsilon(\sigma_i(X)), \tag{12}$$

where g_ϵ is given in (6). For any $X \in \mathbb{R}^{m\wedge n}$ and $\lambda > 0$, if $X^\star \in \mathrm{prox}_{\lambda F}(X)$, then $X^\star = U\mathrm{Diag}(d)V^\top$ for some $(U, V) \in \mathcal{O}(X)$ with $d_i \in \mathrm{prox}_{\lambda g_\epsilon}(\sigma_i(X))$ given by (11).

Proof. By identifying $f = g_\epsilon$ in Theorem 3.10, the rest of the proof is similar to the one in Corollary 3.11. □

4. **Two schemes.**

Two schemes for solving (7) are developed in the last two subsections. Scheme 1 is based on the forward-backward splitting approach in [2]. Scheme 2 is based on decomposition of the regularization term in (7) by exploiting the properties of g_ϵ given in (17) and (18).

4.1. **Scheme 1.**

For this scheme, we use the forward-backward splitting approach to optimize the objective function.

$$X^{(k+1)} \in \operatorname{prox}_{\alpha\lambda G_\epsilon}\left(X^{(k)} - \alpha\nabla F(X^{(k)})\right) \tag{13}$$

The gradient of the fidelity term $\nabla F(X^{(k)}) = D \odot \varphi'_\eta(D \odot (X - Y))$ is Lipschitz continuous with constant 2. Hence, Scheme 1 reads as follows: given an initial $X^{(0)} \in \mathbb{R}^{n\times m}$ and $\alpha < 1/2$, iterate

$$X^{(k+1)} \in \operatorname{prox}_{\alpha\lambda G_\epsilon}\left(X^{(k)} - \alpha D \odot \varphi'_\eta(D \odot (X^{(k)} - Y))\right), \tag{14}$$

for $k = 0, 1, \ldots$. The update $X^{(k+1)}$ can be computed efficiently due to the availability of the operator $\operatorname{prox}_{\alpha\lambda G_\epsilon}$, which can be found in Corollary 3.12.

The convergence analysis of the sequence generalized by Scheme 1 is as follows.

Theorem 4.1. *The sequence $(X^{(k)})_{k\in\mathbb{N}}$ converges to some critical point of the objective function of (7) and $\sum_k \|X^{(k+1)} - X^{(k)}\|_2 < \infty$.*

Proof. We already know $J(X) = \|\phi_\eta(D \odot (X - Y))\|_1 + \lambda G_\epsilon(X)$ the objective function of (7) is a KL function. The results of this theorem follow immediately from [2, Theorem 5.1] if the sequence $(X^{(k)})_{k\in\mathbb{N}}$ is bounded.

Write $W^{(k+1)} := D \odot (X^{(k+1)} - Y)$ and $W^{(k)} := D \odot (X^{(k)} - Y)$. Using the definition of the proximity operator for (14), we have

$$\alpha\lambda G_\epsilon(X^{(k+1)}) + \frac{1}{2}\|X^{(k+1)} - X^{(k)} + \alpha D \odot \varphi'_\eta(W^{(k)})\|_2^2 \le \alpha\lambda G_\epsilon(X^{(k)}) + \frac{1}{2}\|\alpha D \odot \varphi'_\eta(W^{(k)})\|_2^2,$$

which is equivalent to

$$\lambda G_\epsilon(X^{(k+1)}) + \frac{1}{2\alpha}\|X^{(k+1)} - X^{(k)}\|_2^2 + \langle D \odot \phi'_\eta(W^{(k)}), X^{(k+1)} - X^{(k)}\rangle \le \lambda G_\epsilon(X^{(k)}). \tag{15}$$

On the other hand, since the fidelity $\|\phi_\eta(D \odot (\cdot - Y))\|_1$ is convex and its gradient is Lipschitz continuous with constant 2 due to (5), we get

$$\|\phi_\eta(W^{(k+1)})\|_1 \le \|\phi_\eta(W^{(k)})\|_1 + \langle D \odot \phi'_\eta(W^{(k)}), X^{(k+1)} - X^{(k)}\rangle + \|X^{(k+1)} - X^{(k)}\|_2^2 \tag{16}$$

by using the descent lemma for $\|\phi_\eta(D \odot (\cdot - Y))\|_1$ at $X^{(k)}$.

Adding two inequalities (15) and (16) leads to

$$J(X^{(k+1)}) + \frac{1 - 2\alpha}{2\alpha}\|X^{(k+1)} - X^{(k)}\|_2^2 \le J(X^{(k)}).$$

That is, the sequence of the objective function values $(J(X^{(k)}))_{k\in\mathbb{N}})$ is decreasing. Due to the coerciveness of the objective function J, the sequence $(X^{(k)})_{k\in\mathbb{N}}$ must be bounded. This completes the proof. □

4.2. **Scheme 2.** The idea of Scheme 2 is based on the observation: the function g_ϵ can be viewed as the difference of two convex functions in two different ways:

$$g_\epsilon(t) = \frac{|t|}{\epsilon} - \left(\frac{|t|}{\epsilon} - \log\left(\frac{|t|}{\epsilon} + 1\right)\right) \tag{17}$$

and

$$g_\epsilon(t) = \left(\log\left(\frac{|t|}{\epsilon} + 1\right) + \frac{\rho}{2}t^2\right) - \frac{\rho}{2}t^2, \tag{18}$$

where $\rho \geq 1/\epsilon^2$. We can directly verify that $\frac{|t|}{\epsilon} - \log\left(\frac{|t|}{\epsilon}+1\right)$ is differentiable and convex while $\log\left(\frac{|t|}{\epsilon}+1\right) + \frac{\rho}{2}t^2$ is convex. One difference between the decompositions in (17) and (18) is that there is an additional parameter ρ used in (18).

Motivated from (17), Scheme 2 relies on this decomposition on the regularization term in (7) as follows:

$$G_\epsilon(X) = Q(X) - P(X), \tag{19}$$

where

$$Q(X) := \sum_{i=1}^{n\wedge m} \frac{\sigma_i(X)}{\epsilon} \quad \text{and} \quad P(X) = \sum_{i=1}^{n\wedge m}\left(\frac{\sigma_i(X)}{\epsilon} - g_\epsilon(\sigma_i(X))\right). \tag{20}$$

The following lemma gives the properties of P and Q defined in (20).

Lemma 4.2. *Let both $P : \mathbb{R}^{n\times m} \to \mathbb{R}$ and $Q : \mathbb{R}^{n\times m} \to \mathbb{R}$ be defined in* (20)*. Then, both P and Q are convex on $\mathbb{R}^{n\times m}$. Furthermore, define $h : \mathbb{R}^{n\wedge m} \to \mathbb{R}$ as follows:*

$$h(x) = \sum_{i=1}^{n\wedge m}\left(\frac{|x_i|}{\epsilon} - g_\epsilon(x_i)\right).$$

Then, P is differentiable and its gradient is

$$\nabla P(X) = U\mathrm{Diag}(\nabla h(\sigma(X)))V^\top,$$

where for any $(U, V) \in \mathcal{O}(X)$.

Proof. The function $Q(X)$ is the constant $1/\epsilon$ multiplying the nuclear norm of X, hence Q is convex. Note that $P(X) = h(\sigma(X))$. It can be checked directly that h is convex on $\mathbb{R}^{n\wedge m}$ and is absolutely symmetric, i.e.,

$$h(x_1, x_2, \ldots, x_{n\wedge m}) = h(|x_{\pi(1)}|, |x_{\pi(2)}|, \ldots, |x_{\pi(n\wedge m)}|),$$

for any permutation π. By [18, Proposition 6.1], P is convex on $\mathbb{R}^{n\times m}$. The final result on the gradient of P follows from [18, Proposition 6.2]. □

Using the decomposition (19), problem (7) can be rewritten as

$$\min\{J(X) = F(X) + \lambda Q(X) - \lambda P(X) : X \in \mathbb{R}^{n\times m}\}. \tag{21}$$

By Lemma 4.2, the objective function $J(X)$ of problem (21) is the sum of $\|\varphi_\eta(D \odot (X - Y))\|_1$ a smooth convex function with Lipschitz gradient, $\lambda Q(X)$ a proper convex function and $-\lambda P(X)$ a differentiable concave function. Due to this structure, the proximal difference-of-convex algorithm with extrapolation in [25] can be adopted for solving problem (21). This leads to our Scheme 2 as follows: set $\alpha < 1/2$, $\{\beta_k\} \subset [0, 1)$ with $\sup_k \beta_k < 1$, $X^{(-1)} = X^{(0)} \in \mathbb{R}^{n\wedge m}$, iterate

$$\begin{cases} W^{(k)} &= X^{(k)} + \beta_k(X^{(k)} - X^{(k-1)}), \\ X^{(k+1)} &= \mathrm{prox}_{\alpha\lambda Q}\left(W^{(k)} - \alpha\left(\nabla F(W^{(k)}) - \lambda\nabla P(X^{(k)})\right)\right), \end{cases} \tag{22}$$

where $(U^{(k)}, V^{(k)}) \in \mathcal{O}(X^{(k)})$.

From Corollary 3.11, $\mathrm{prox}_{\alpha\lambda Q}$ in Scheme 2 is the singular value shrinkage operator with threshold $\frac{\alpha\lambda}{\epsilon}$. Write

$$\widetilde{X}^{(k+1)} := W^{(k)} - \alpha\left(\nabla F(W^{(k)}) - \lambda \nabla P(X^{(k)})\right). \tag{23}$$

We get

$$X^{(k+1)} = U^{(k+1)} \mathrm{Diag}(d^{k+1}) (V^{(k+1)})^\top, \tag{24}$$

where $(U^{(k+1)}, V^{(k+1)}) \in \mathcal{O}(\widetilde{X}^{(k+1)})$, and for all $i = 1, 2, \dots, (n \wedge m)$

$$d_i^{k+1} = \begin{cases} 0, & \text{if } \sigma_i(\widetilde{X}^{(k+1)}) \leq \frac{\alpha\lambda}{\epsilon}; \\ \sigma_i(\widetilde{X}^{(k+1)}) - \frac{\alpha\lambda}{\epsilon}, & \text{otherwise.} \end{cases} \tag{25}$$

For the sequence $(X^{(k)})_{k\in\mathbb{N}}$ generated by Scheme 2, by Theorem 4.1 in [25] this sequence is bounded, $\lim_{k\to\infty} \|X^{(k+1)} - X^{(k)}\|_2 = 0$, and any accumulation point of $(X^{(k)})_{k\in\mathbb{N}}$ is a stationary point of the objective function of (7), i.e,

$$0 \in \nabla F(X^{(k)}) + \lambda(\partial Q(X^{(\star)}) - \nabla P(X^{(\star)})),$$

where $X^{(\star)}$ is an accumulation point of $(X^{(k)})_{k\in\mathbb{N}}$. Furthermore, by Theorem 4.2 in [25], we have $\sum_k \|X^{(k+1)} - X^{(k)}\|_2 < \infty$.

We introduce an alternative decomposition of G_ϵ based on (18). The pair (P, Q) used in (19) is defined as follows:

$$Q(X) := \sum_{i=1}^{n\wedge m} \left(g_\epsilon(\sigma_i(X)) + \frac{\rho}{2}\sigma_i^2(X)\right) \quad \text{and} \quad P(X) := \frac{\rho}{2}\|\sigma(X)\|_2^2 \tag{26}$$

with any $\rho \geq \frac{1}{\epsilon^2}$. Clearly, $P(X)$ is the square of the Schatten-2 norm of X multiplying the constant $\frac{\rho}{2}$, which is also the square of the Frobenius norm of X multiplying the same constant, yielding $\nabla P(X) = \rho X$. Viewing $g_\epsilon(\cdot) + \frac{\rho}{2}(\cdot)^2$, which is convex by (18), as the function f in Theorem 3.10, one immediately sees that $Q(X) = \sum_{i=1}^{n\wedge m} f(\sigma_i(X))$ and the function f satisfies all assumptions on it. With this pair of (P, Q), scheme 2 becomes

$$\begin{cases} W^{(k)} &= X^{(k)} + \beta_k(X^{(k)} - X^{(k-1)}) \\ X^{(k+1)} &= \mathrm{prox}_{\alpha\lambda Q}\left(W^{(k)} - \alpha\left(\nabla F(W^{(k)}) - \lambda\rho X^{(k)}\right)\right) \end{cases} \tag{27}$$

Write

$$\widetilde{X}^{(k+1)} := W^{(k)} - \alpha\left(\nabla F(W^{(k)}) - \lambda\rho X^{(k)}\right). \tag{28}$$

By Theorem 3.10, we have

$$X^{(k+1)} = U^{(k+1)} \mathrm{Diag}(d^{k+1}) (V^{(k+1)})^\top, \tag{29}$$

where $(U^{(k+1)}, V^{(k+1)}) \in \mathcal{O}(\widetilde{X}^{(k+1)})$, and for all $i = 1, 2, \dots, (n \wedge m)$

$$d_i^{k+1} = \mathrm{prox}_{\alpha\lambda f}(\sigma_i(\widetilde{X}^{(k+1)})) = \mathrm{prox}_{\frac{\alpha\lambda}{1+\alpha\lambda\rho} g_\epsilon}\left(\frac{\sigma_i(\widetilde{X}^{(k+1)})}{1+\alpha\lambda\rho}\right).$$

Note that $\frac{\alpha\lambda}{1+\alpha\lambda\rho} < \frac{1}{\rho} \leq \epsilon^2$. By (11) and (10) with some manipulations, we know that

$$d_i^{k+1} = \begin{cases} 0, & \text{if } \sigma_i(\widetilde{X}^{(k+1)}) \leq \frac{\alpha\lambda}{\epsilon}; \\ \frac{(\sigma_i(\widetilde{X}^{(k+1)}) - \epsilon(1+\alpha\lambda\rho)) + \sqrt{(\sigma_i(\widetilde{X}^{(k+1)}) + \epsilon(1+\alpha\lambda\rho))^2 - 4\alpha\lambda(1+\alpha\lambda\rho)}}{2(1+\alpha\lambda\rho)}, & \text{otherwise.} \end{cases} \tag{30}$$

At first glance, both iterative schemes (22) and (27) should theoretically work for optimization problem (7). However, our numerical experiments for mixed noise removal showed that scheme (22) works very well, but scheme (27) does not. Specifically, all iterates $X^{(k)}$ generated by scheme (27) remain nearly indistinguishable from the initial guess $X^{(0)}$. A similar observation has been reported in [25]. However, a plausible explanation for this phenomenon remains unknown.

5. Our algorithms for mixed noise removal.

In this section, we present our algorithm for model (1). As it was mentioned earlier, the general framework for model (1) presented here is similar to the NLR-TP in [27]. The main difference is how to solve the optimization problem (7). With the preparations in the previous sections, we are ready to present our two-phase algorithm for model (1). Phase I is to identify the set $\mathcal{Z}$ the locations of outlier candidates in the noisy image y. According to the types of impulse noise in y, we use AMF [16] for salt-and-pepper noise and use ACWMF [10] for random-valued impulse noise to estimate $\mathcal{Z}$. In Phase II, for a local patch, say y_ℓ, of y, we search for its nonlocal patches across the image to form the corresponding patch matrix Y_ℓ, which is Y in model (2). Accordingly, we have X_ℓ and Z_ℓ that can be viewed as X and Z, respectively, in model (2). Therefore, we can estimate X_ℓ from Y_ℓ through either Scheme 1 or Scheme 2. By rendering all patches, we aggregate all estimated X_ℓ together, yielding an reconstructed image x. As suggested in [13], we can run the Phase II several more rounds to further enhance the denoised image. A complete procedure for model (1) with K rounds of Phase II is summarized in Algorithm 1.

We manage three parameters, namely η, ϵ, and λ in (7). Our strategy for adjusting these parameters involves varying η based on the iteration and adjusting ϵ and λ according to noise levels in observed images. Specifically, at the kth iteration of Scheme 1 or Scheme 2 for patch ℓ, we set η as follows:

$$\eta = \frac{\sqrt{2}}{|\Omega \setminus \mathcal{Z}|} \|D \odot (X_\ell^{(k)} - Y)\|_1, \tag{31}$$

where $|\Omega \setminus \mathcal{Z}|$ is the cardinality of the set $\Omega \setminus \mathcal{Z}$. As for the parameter α common to both Scheme 1 and Scheme 2, we consistently choose $\alpha = 0.499$, ensuring it is less than $1/2$.

In each iteration of Phase II in our algorithm, we generate a denoised image by combining all the estimated patch matrices. For the i-th iteration, the pixel value at a specific point in the denoised image $x^{(i)}$ is obtained as the weighted average of the pixel values at the corresponding position across all estimated patches covering this pixel. While various weighted averaging strategies can be employed, we opt for the straightforward approach of using equal weights for all patches. This choice is made based on our observations from numerical experiments, where we have not identified significant differences with other weighting strategies.

6. Numerical experiments.

The superior performance of NLR-TP against state-of-the-art denoising methods, including WESNR [17], WJSR[20] and LSM-NLR[15] for mixed noise removal, has been reported in [27]. Therefore, in this section, we present numerical experiments to illustrate the effectiveness of our proposed Algorithm 1 by comparing it with NLR-TP. To distinguish the usage of Scheme 1 and Scheme 2, Algorithm 1-S1 and Algorithm 1-S2 refer to Algorithm 1.

Algorithm 1 Mixed Noise Removal for (1)

```
 1: Input: Degraded image $y$; parameters
 2: Set parameters $\delta > 0$, patch size, patch number, outer iteration number $K$,inner
    iteration number $M$;
 3: Initialize $x^{(0)} = y$ and $y^{(0)} = y$;
 4: Phase I: Apply AMF (for salt-and-pepper noise) or ACWMF (for random-valued
    noise) to estimate the set $\mathcal{Z}$ and obtain the binary matrix $D$;
 5: Phase II:
 6: for $i$ from 1 to $K$ do
 7:     $y^{(i)} = x^{(i-1)} + \delta(y - x^{(i-1)})$;
 8:     for each patch $y_\ell$ of $y^{(i)}$ do
 9:         Form the patch matrix $Y_\ell$;
10:         for $t$ from 1 to $M$ do
11:             Set parameters $\eta$,$\epsilon$,$\lambda$,$\alpha < 1/2$, $\{\beta_k\} \subset [0, 1)$ with $\sup_k \beta_k < 1$;
12:             1. Scheme 1:
                $X^{(k+1)} \in \mathrm{prox}_{\alpha\lambda G_\epsilon}\left(X^{(k)} - \alpha D \odot \varphi'_\eta(D \odot (X^{(k)} - Y))\right);$
13:             2. Scheme 2:
                $W^{(k)} = X^{(k)} + \beta_k(X^{(k)} - X^{(k-1)}),$
                $X^{(k+1)} = \mathrm{prox}_{\alpha\lambda Q}\left(W^{(k)} - \alpha\left(\nabla F(W^{(k)}) - \lambda\nabla P(X^{(k)})\right)\right),$
14:         end for
15:     end for
16:     Aggregate $X_\ell$ to form the denoised image $x^{(i)}$;
17: end for
```

To compare our proposed algorithm and NLR-TP, we select four testing images: the "Barbara", "House" and "Boat" images of size 512 × 512, as shown in Figure 1(a), 1(b), and 1(c), respectively. The "Barbara" image is widely used due to its rich details, including edges and smooth regions. The "House" image features well-defined edges and extensive cartoon-like regions. The "Boat" image includes elements such as clouds and water surfaces, which lack distinct texture features. Additionally, we include a 512 × 512 color image of "Baboon" in Figure 1(d) to demonstrate the applicability of our algorithm to color images.

For testing our algorithm, we added mixed noise to the images, comprising Gaussian noise and another type of impulse noise, either salt-and-pepper or random-value. Details on the noise types will be discussed in the following subsections.

The effectiveness of algorithms in removing mixed noise are assessed by three metrics, namely, the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM) [24] and the feature similarity (FSIM) [28]. The PSNR mainly measures the intensity similarity between an reconstructed image and its reference image while the other two mainly measure the perceptual image quality of an reconstructed image. Normally, the higher PSNR, SSIM, and FSIM scores are, the better the quality of the reconstructed images is.

Algorithm 1 consists of two loops. The number of iterations for performing both Scheme 1 and Scheme 2 (inner loop) in line 10 is set to 15 and 25, respectively. Meanwhile, the maximum number of iterations for updating the observed patch

(a) Barbara (b) House (c) Boat (d) Baboon

FIGURE 1. The testing images (a) "Barbara", (b) "House", (c) "Boat", and (c) "Baboon".

matrices (outer loop) from line 6 to line 11 is set to 50 (i.e., $K = 50$). The outer loop terminates once the reconstructed image achieves the highest PSNR value.

In the following sections, we present our numerical results separately based on the types of mixed noise and the image color type.

6.1. **Mixed Salt-and-Pepper noise and Gaussian noise.** In this subsection, $\mathcal{N}_{imp}$ in model (1) represents the salt-and-pepper noise. The noise levels of the salt-and-pepper noise in our experiments are set to be 20, 30, and 50 while the standard variations of the Gaussian noise are set to 10 and 20. As a result, six different mixed noise scenarios are tested for each image.

For this type of noise, we need to set parameters λ, ϵ and block size. We choose $\epsilon = 0.01$ and a block size of 10. Table 1 reports the values of λ used in Algorithm 1-S1 and Algorithm 1-S2 for testing the "Barbara", "House" and "Boat" images across the six cases. In this table, 'sp20+gs10' refers the mixed noise with the level of salt-and-pepper noise 20 and the standard deviation of Gaussian noise 10. Similar notations apply to the other mixed noise cases.

In Table 2, we present the results of three metrics-PSNR, SSIM, and FSIM-for NLR-TP, Algorithm 1-S1 and Algorithm 1-S2. For the image of "Barbara", we observe that Algorithm 1-S1 and Algorithm 1-S2 perform better than NLR-TP in terms of the values of PSNR, in particular in the 'sp50+gs10' noise case. Both Algorithm 1-S1 and Algorithm 1-S2 exhibit comparable performance. Regarding the "House" image, Algorithm 1-S2 achieves the highest performance, followed by Algorithm 1-S1 in terms of PSNR values. For the "Boat" image, although with $\sigma = 20$ our algorithm's numerical performance is inferior to NLRTP (still quite close), at sigma = 10 our algorithm performs better numerically. Figures 2(a), 2(b) and 2(c) provide visualizations of the PSNR values for the "Barbara", "House" and "Boat" images, respectively, at a noise level of $\sigma = 10$.

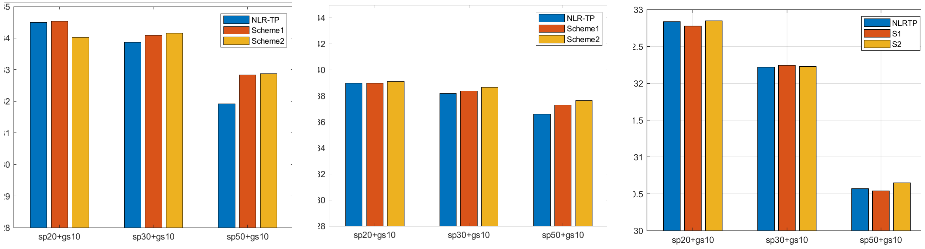
Figures 3, 4 and 5 display the visual quality of the denoised images of "Barbara", "House" and "Boat" for the cases having the standard deviation of Gaussian noise 10 (Boat) or 20 (Barbara,House). Figure 3(a) is the noisy "Barbara" image with Gaussian noise of $\sigma = 20$ and salt-and-pepper noise level is 50. The denoised images by NLR-TP, Algorithm 1-S1 and Algorithm 1-S2 are depicted in Figure 3(b), 3(c), and 3(d), respectively.

For the "Barbara" image, it is evident that all three methods—NLR-TP, Algorithm 1-S1, and Algorithm 1-S2—effectively remove noise and restore the contours of the image. In particular, the texture of the clothing is well preserved. The second row of Figure 3 presents a zoomed-in view of a specific region from the first row images. Close-up views corresponding to NLR-TP, Algorithm 1-S1, and Algorithm 1-S2 all maintain the striped patterns in that region. However, a closer inspection reveals that Algorithm 1-S1 and Algorithm 1-S2 produce less blurring in the top-left corner of the zoomed-in area compared to NLR-TP, indicating superior preservation of fine details.

Similarly, Figures 4 and Figure 5 show the image of the denoised "House" and "Boat". Our method effectively restores image contours while preserving edge sharpness, demonstrating its capability to remove this type of mixed noise. The second row of Figure 4 provides a zoomed-in view of a selected region from the first-row images. Due to the presence of noise, none of the three methods can fully recover the vertical texture on the wall in the zoomed-in area. However, all three approaches successfully preserve the contours and edges of the window, highlighting their effectiveness in maintaining structural details. A similar observation holds for the "Boat" image in Figure 5.

Table 1. Parameters set for mixed salt-and-pepper noise and Gaussian noise.

Algorithm	Algorithm 1-S1				Algorithm 1-S2			
Noise Type	η	λ	ϵ	block size	η	λ	ϵ	block size
Images: Barbara and House								
sp20+gs10	1	4.0e-4	0.01	10	1	2e-4	0.01	10
sp30+gs10	1	3.5e-4	0.01	10	1	2e-4	0.01	10
sp50+gs10	1	3.4e-4	0.01	12	1	1.8e-4	0.01	12
sp20+gs20	1	1.1e-4	0.01	12	1	5e-5	0.01	12
sp30+gs20	1	1e-4	0.01	12	1	5e-5	0.01	12
sp50+gs20	1	9e-5	0.01	12	1	5e-5	0.01	12
Image: Boat								
sp20+gs10	1	5e-4	0.1	10	3	2.3e-3	0.8	10
sp30+gs10	1	8e-4	0.1	10	3	2.1e-3	0.1	10
sp50+gs10	1	1.3e-3	1	15	5	1.5e-3	1	15
sp20+gs20	7	5.5e-3	0.01	10	0.5	1.3e-3	0.5	10
sp30+gs20	7	9e-3	1	10	0.5	1.2e-3	0.5	10
sp50+gs20	9	1.5e-4	0.01	15	5	7e-4	1	15



(a) (b) (c)

Figure 2. The PSNR values of the denoised images from salt and pepper noise by various methods for the images of (a) "Barbara", (b) "House" and (c) "Boat".

TABLE 2. Comparison of NLR-TP and Algorithm 1.

Algorithm	NLR-TP	Algorithm 1-S1	Algorithm 1-S2
Noise Type	(PSNR, SSIM, FSIM)	(PSNR, SSIM, FSIM)	(PSNR, SSIM, FSIM)
The “Barbara” image			
sp20+gs10	(34.50, 0.9731, 0.9806)	(34.53, 0.9735, 0.9816)	(**34.62**, **0.9741**, **0.9821**)
sp30+gs10	(33.87, 0.9700, 0.9783)	(34.09, 0.9714, 0.9801)	(**34.15**, **0.9718**, **0.9804**)
sp50+gs10	(31.92, 0.9563, 0.9705)	(32.83,0.9629, **0.9742**)	(**32.87**, **0.9640**, 0.9737)
sp20+gs20	(31.34, 0.9481, 0.9637)	(31.40,0.9478, **0.9649**)	(**31.44**, **0.9490**, 0.9648)
sp30+gs20	(30.90, 0.9451, 0.9606)	(**31.02**, 0.9441,**0.9649**)	(31.00, **0.9458**, 0.9617)
sp50+gs20	(30.01, 0.9334, 0.9537)	(**30.06**, 0.9312,**0.9551**)	(30.04, **0.9337**, 0.9549)
The “House” image			
sp20+gs10	(38.99, 0.9598, 0.9760)	(38.97, 0.9588, 0.9766)	(**39.12**, **0.9618**, **0.9777**)
sp30+gs10	(38.17, 0.9550, 0.9732)	(38.38, 0.9518, 0.9719)	(**38.67**, **0.9580**, **0.9750**)
sp50+gs10	(36.58, 0.9430, 0.9704)	(37.31, 0.9410, 0.9677)	(**37.65**, **0.9505**, **0.9705**)
sp20+gs20	(35.76, 0.9290, 0.9542)	(35.69, 0.9227, 0.9472)	(**35.77**, **0.9297**, **0.9543**)
sp30+gs20	(35.34, **0.9257**, **0.9528**)	(35.34, 0.9187,0.9438)	(**35.37**, 0.9248, 0.9520)
sp50+gs20	(34.73, **0.9150**, **0.9457**)	(34.73, 0.9114, 0.9397)	(**34.74**, 0.9119, 0.9425)
The “Boat” image			
Bsp20+gs10	(32.84, **0.9565**, 0.9768)	(32.78, **0.9565**, **0.9777**)	(**32.85**, 0.9553, 0.9771)
Bsp30+gs10	(32.22, 0.9497, 0.9724)	(**32.24**, **0.9511**, **0.9738**)	(32.23, 0.9501, 0.9731)
Bsp50+gs10	(30.57, 0.9298, 0.9612)	(30.54, **0.9311**, 0.9614)	(**30.65**, 0.9307, **0.9618**)
Bsp20+gs20	(**29.91**, **0.9099**, 0.9484)	(29.82, 0.9028, **0.9490**)	(29.81, 0.9067, 0.9484)
Bsp30+gs20	(**29.47**, **0.9023**, **0.9439**)	(29.31, 0.8904, 0.9431)	(29.30, 0.8951, 0.9401)
Bsp50+gs20	(**28.31**, 0.8726, 0.9255)	(28.20, **0.8736**, **0.9291**)	(28.26, 0.8736, 0.9294)

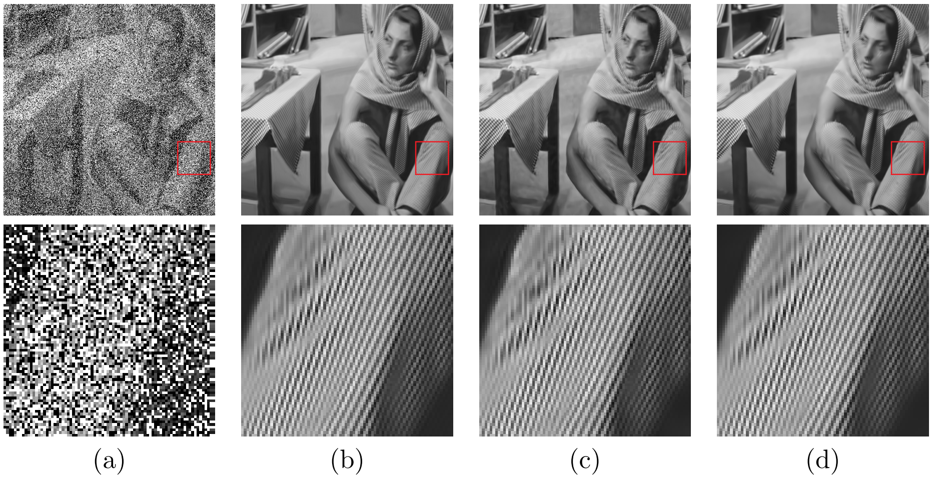


FIGURE 3. (a) The noisy “Barbara” image with Gaussian noise ($\sigma = 20$) and salt-and-pepper noise at level 50. The denoised images are obtained using (b) NLR-TP, (c) Algorithm 1-S1, and (d) Algorithm 1-S2. The second row displays the corresponding zoomed-in regions of the images in the first row.

6.2. **Mixed Random Impulse Noise and Gaussian Noise.** In this subsection, $\mathcal{N}_{imp}$ in model (1) represents the random-valued impluse noise. The noise levels of the impulse noise used in our experiments are set to be 15, 25, and 45 while the standard variations of the Gaussian noise are set to 15 and 25. Therefore, there are six cases of mixed noise to be tested for each test image.

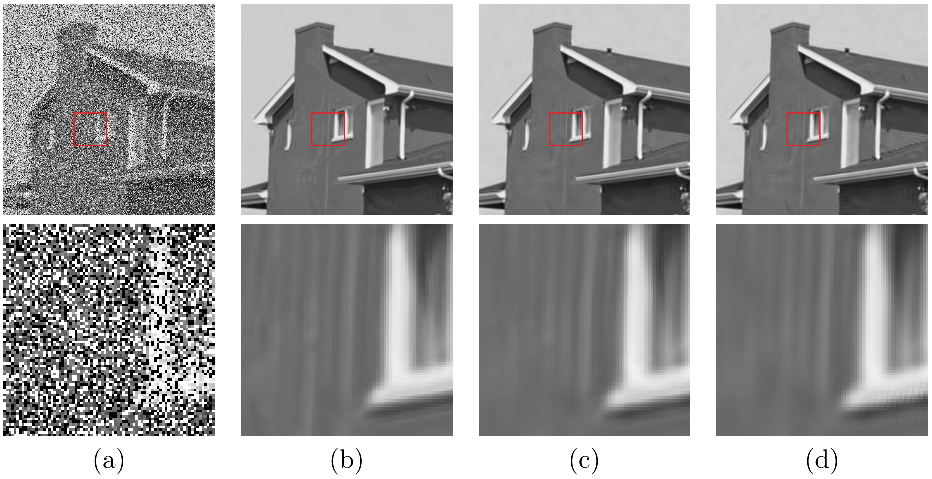


FIGURE 4. (a) The noisy "House" image with Gaussian noise ($\sigma = 20$) and salt-and-pepper noise at level 50. The denoised images are obtained using (b) NLR-TP, (c) Algorithm 1-S1, and (d) Algorithm 1-S2. The second row displays the corresponding zoomed-in regions of the images in the first row.

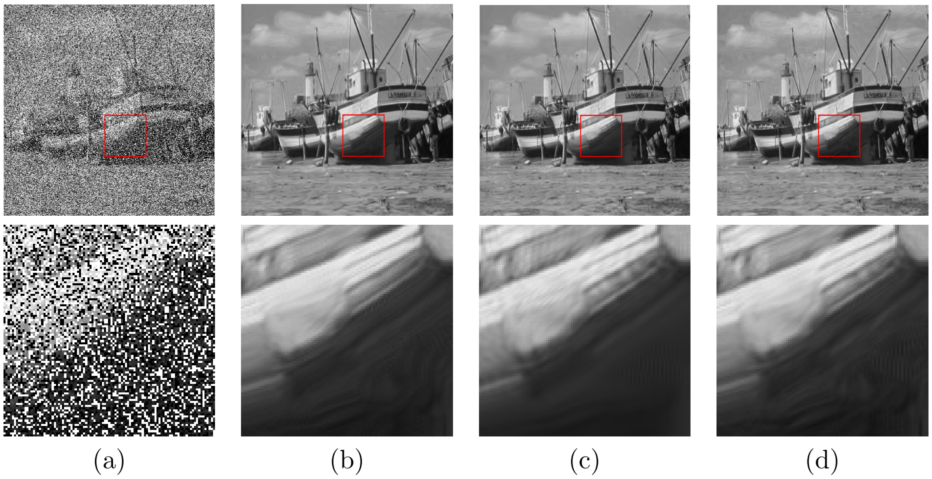


FIGURE 5. (a) The noisy "Boat" image with Gaussian noise ($\sigma = 10$) and salt-and-pepper noise at level 50. The denoised images are obtained using (b) NLR-TP, (c) Algorithm 1-S1, and (d) Algorithm 1-S2. The second row displays the corresponding zoomed-in regions of the images in the first row.

Table 3 shows the parameters used in Algorithm 1-S1 and Algorithm 1-S2 for testing the "Barbara", "House" and "Boat" images at six cases. In the table, 'rv15+gs15' refers the mixed noise with the level of random-valued impluse noise 15 and the standard deviation of Gaussian noise 15. Similar explanation for other mixed noise cases is obvious. For each case, the parameters λ and ϵ in optimization problem (7) together with the block size for patch matrices are provided.

In Table 4, we present the results of three metrics PSNR, SSIM, and FSIM corresponding to NLR-TP, Algorithm 1-S1 and Algorithm 1-S2. For the image of "Barbara", we observe that Algorithm 1-S1 and Algorithm 1-S2 perform significantly better than NLR-TP in terms of the values of PSNR for all tested cases. Algorithm 1-S1 is better than Algorithm 1-S2 except in the case of rv45+gs15. For the "House" image, three algorithms perform comparable for the noise cases rv15+gs15 and rv25+gs15 while both Algorithm 1-S1 and Algorithm 1-S2 perform better than NLR-TP in the other cases in terms of the values of PSNR for all tested cases. Regarding the "Boat" image, both Algorithm 1-S1 and Algorithm 1-S2 perform better than NLR-TP in terms of PSNR in all cases. Specifically, Algorithm 1-S1 performs significantly better than Algorithm 1-S2.

Figure 6(a), 6(b) and 6(c) provide visualizations of the PSNR values for the "Barbara","House" and "Boat" images, respectively, at a noise level of $\sigma = 25$.

We present the visual quality of the denoised images for "Barbara", "House" and "Boat" with Gaussian noise at a standrad deviation $\sigma = 25$ and random-valued noise at level 45. Figure 7(a) shows the noisy "Barbara" image. The denoised images by NLR-TP, Algorithm 1-S1 and Algorithm 1-S2 are depicted in Figure 7(b), 7(c), and 7(d), respectively. For the "Barbara" image, we observe that Figure 7(b), 7(c), and 7(d) are quite clear, and the textures within the image are also distinctly visible. This suggests that our method effectively removes noise. However, it is evident that Figure 7(b) contains more unnatural spots and streaks compared to Figure 7(c) and 7(d), which appear clearer in comparison. In the second row of Figure 7, we shoe a zoomed-in portion of the images from the first row. The zoomed-in part corresponding to NLR-TP appears jagged, likely because it cannot effectively remove random-valued noise. In contrast, the zoomed-in portions of Algorithm 1-S1 and Algorithm 1-S2 are smoother and appear more natural.

Figure 8 shows the denoised "House" image. We observe that Figure 8(c) and 8(d) appear clearer than Figure 8(b). In Figure 8(b), the staircase effect along the edges is noticeable, but it is significantly reduced in Figures 8(c) and 8(d), particularly in Figure 8(d), where the contours and edges are much more natural and distinct. The second row of Figure 8 shows a zoomed-in portion of the images from the first row. The zoomed-in parts corresponding to Algorithm 1-S1 and Algorithm 1-S2 show clearer edges compared to NLR-TP. Notably, the zoomed-in portion from Algorithm 1-S2 even restores the texture on the wall.

Similarly, Figure 9 shows the denoised "Boat" image. It is evident that our method effectively removes the noise in Figure 9(a). In the zoomed-in section of the boat shown in the second row, we can observe that our method effectively preserves both the boat's edge regions and the smooth areas. The details in Figure 9(d), such as the lines, are much clearer.

6.3. **Denoising of color images.** To conclude the numerical section, we evaluate our algorithm on color images. Specifically, we apply mixed Gaussian noise with a standard deviation of $\sigma = 20$ and salt-and-pepper noise at level 30 to the "Baboon" image in Figure 1(d). The numerical values of (PSNR, SSIM, FSIM) for the denoised images are (22.32, 0.6786, 0.8562), (22.38, 0.6897, 0.8618), and (22.33, 0.6878, 0.8615) by NLR-TP, Algorithm 1-S1, and Algorithm 1-S2, respectively. Among these,Algorithm 1-S1 achieves slightly better performance than the other

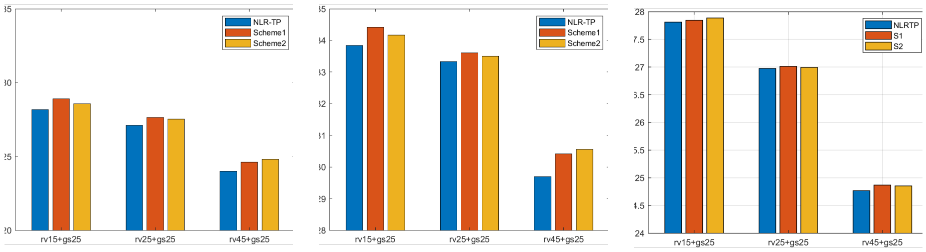
Table 3. Parameter sets for denoising the image with Algorithm 1.

Algorithm	Algorithm 1-S1				Algorithm 1-S2			
Noise Type	η	$\frac{1}{\lambda}$	ϵ	block size	η	$\frac{1}{\lambda}$	ϵ	block size
Image: Barbara Figure 1(a)								
rv15+gs15	1	7.0e-5	1e-2	12	1.7	1.7e-4	0.01	12
rv25+gs15	1	5.0e-5	1e-2	12	1.7	1.8e-4	0.01	12
rv45+gs15	1	3.0e-5	1e-2	12	1.7	1.8e-4	0.01	12
rv15+gs25	1	3.0e-5	1e-2	12	1.7	1.0e-4	0.01	12
rv25+gs25	1	2.2e-5	1e-2	12	1.7	1.0e-4	0.01	12
rv45+gs25	1	1.5e-5	1e-2	12	1.7	1.0e-5	0.01	12
Image: House Figure 1(b)								
rv15+gs15	1	7.0e-5	1e-3	18	1.7	1.6e-4	1e-1	18
rv25+gs15	1	5.0e-5	1e-3	18	1.7	1.2e-4	1e-1	18
rv45+gs15	1	2.0e-5	1e-3	18	1.7	1.2e-4	1e-2	18
rv15+gs25	1	2.2e-5	1e-3	18	1.7	1.0e-4	1e-2	12
rv25+gs25	1	1.8e-5	1e-3	18	1.7	7.0e-5	1e-1	15
rv45+gs25	1	9.0e-6	1e-3	18	1.7	8.0e-5	1e-2	12
Image: Boat Figure 1(c)								
rv15+gs15	1	1e-3	1	12	1.5	7e-4	2.5	10
rv25+gs15	1	5e-4	1	12	1.2	3e-4	2.5	10
rv45+gs15	1	2.0e-4	1	12	1	3e-4	2.5	15
rv15+gs25	1	2.5e-4	1	10	1.5	2.5e-4	5	10
rv25+gs25	1	2.0e-4	1	10	1.5	1.5e-4	5	10
rv45+gs25	1	1.25e-4	1	10	1.5	1e-4	5	15

Table 4. Comparison of NLR-TP and Algorithm 1

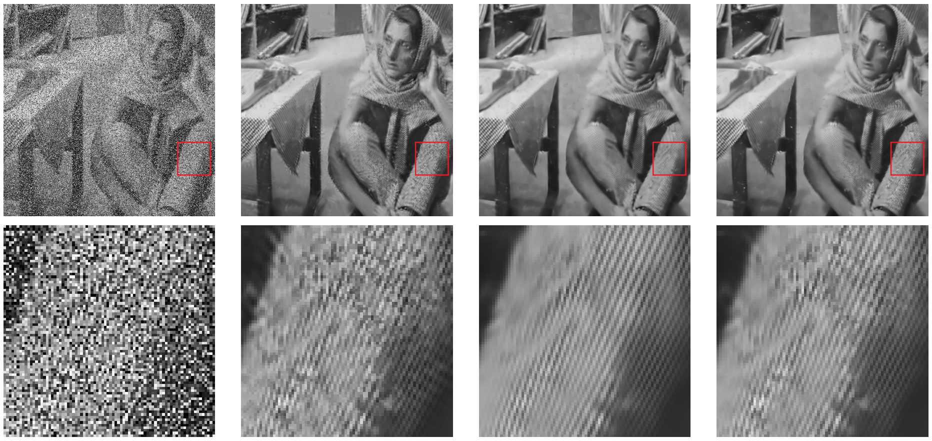
Algorithm	NLR-TP	Algorithm 1-S1	Algorithm 1-S2
Noise Type	(PSNR, SSIM, FSIM)	(PSNR, SSIM, FSIM)	(PSNR, SSIM, FSIM)
The “Barbara” image			
rv15+gs15	(29.78, 0.9393, 0.9600)	(30.41, 0.9437, 0.9633)	(**30.40**, **0.9443**, **0.9631**)
rv25+gs15	(28.57, 0.9261, 0.9543)	(**29.14**, **0.9304**, **0.9581**)	(28.98, 0.9246, 0.9519)
rv45+gs15	(25.16, 0.8564, 0.9252)	(25.33, 0.8567, 0.9302)	(**26.00**, **0.8775**, **0.9322**)
rv15+gs25	(28.16, 0.9074, 0.9415)	(**28.89**, **0.9178**, **0.9468**)	(28.57, 0.9068, 0.9439)
rv25+gs25	(27.10, 0.8839, 0.9324)	(**27.64**, **0.8919**, **0.9375**)	(27.52, 0.8837, 0.9349)
rv45+gs25	(23.99, 0.7946, 0.8905)	(24.61, 0.8075, **0.9028**)	(**24.81**, **0.8151**, 0.8979)
The “House” image			
rv15+gs15	(36.69, 0.9358, 0.9589)	(36.68, 0.9339, 0.9587)	(**36.69**, **0.9358**, **0.9601**)
rv25+gs15	(36.01, 0.9238, 0.9488)	(36.10, 0.9248, 0.9504)	(**36.18**, **0.9301**, **0.9558**)
rv45+gs15	(33.08, 0.9022, 0.9332)	(33.68, 0.9036, **0.9359**)	(**33.74**, **0.9057**, 0.9337)
rv15+gs25	(33.84, **0.9086**, **0.9415**)	(**34.42**, 0.9065, 0.9351)	(34.17, 0.9029, 0.9337)
rv25+gs25	(33.33, 0.8966, **0.9299**)	(**33.61**, **0.8986**, 0.9282)	(33.50, 0.8976, 0.9269)
rv45+gs25	(29.69, 0.8627, 0.9046)	(30.41, **0.8725**, **0.9086**)	(**30.56**, 0.8636, 0.9078)
The “Boat” image			
rv15+gs15	(29.65, 0.9126, 0.9519)	(29.65, 0.9077, 0.9503)	(**29.66**, **0.9063**, **0.9502**)
rv25+gs15	(28.59, 0.8870, 0.9367)	(**28.69**, **0.8932**, **0.9419**)	(28.64, 0.8927, 0.9417)
rv45+gs15	(26.34, 0.8378, 0.9110)	(26.43, 0.8366, 0.9103)	(**28.42**, **0.8379**, **0.9115**)
rv15+gs25	(27.81, **0.8607**, **0.9252**)	(27.85, 0.8554, 0.9228)	(**27.89**, 0.8558, 0.9240)
rv25+gs25	(26.98, 0.8314, 0.9068)	(**27.01**, 0.8326, 0.9097)	(27.99, **0.8341**, **0.9117**)
rv45+gs25	(24.77, **0.7559**, 0.8675)	(**24.87**, 0.7506, 0.8635)	(24.85, 0.7544, **0.8686**)

two. The visual quality of the denoised images is shown in Figure 10, demonstrating that our algorithm is effectively applicable to color images.



(a) (b) (c)

FIGURE 6. The PSNR values of the denoised images from random valued noise by various methods for the images of (a) "Barbara", (b) "House" and (c) "Boat".



(a) (b) (c) (d)

FIGURE 7. (a) The noisy "Barbara" image with Gaussian noise ($\sigma = 25$) and random-valued impluse noise at level 45. The denoised images are obtained using (b) NLR-TP, (c) Algorithm 1-S1, and (d) Algorithm 1-S2. The second row displays the corresponding zoomed-in regions of the images in the first row.

7. Conclusions.

In this paper, we proposed efficient algorithms for solving an optimization problem whose objective function consists of a convex content-dependent fidelity term and a nonlocal low-rank regularization term, both constructed using patch matrices formed from similar patches of an image. We demonstrated that the objective function satisfies the Kurdyka-Łojasiewicz (KL) property and leveraged this property to analyze the convergence of the proposed algorithms. Numerical experiments were conducted, showing that our algorithms outperform NLR-TP. However, our model and algorithms involve several parameters, and selecting their optimal values remains a challenge. Future work will focus on developing effective strategies for parameter selection to further enhance the practicality and robustness of the proposed approach.

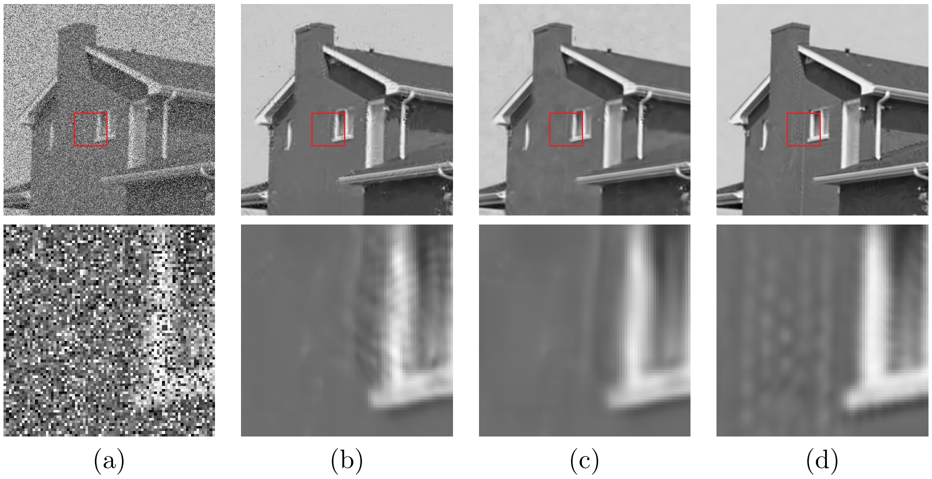


FIGURE 8. (a) The noisy "House" image with Gaussian noise ($\sigma = 25$) and random-valued impluse noise at level 45. The denoised images are obtained using (b) NLR-TP, (c) Algorithm 1-S1, and (d) Algorithm 1-S2. The second row displays the corresponding zoomed-in regions of the images in the first row.

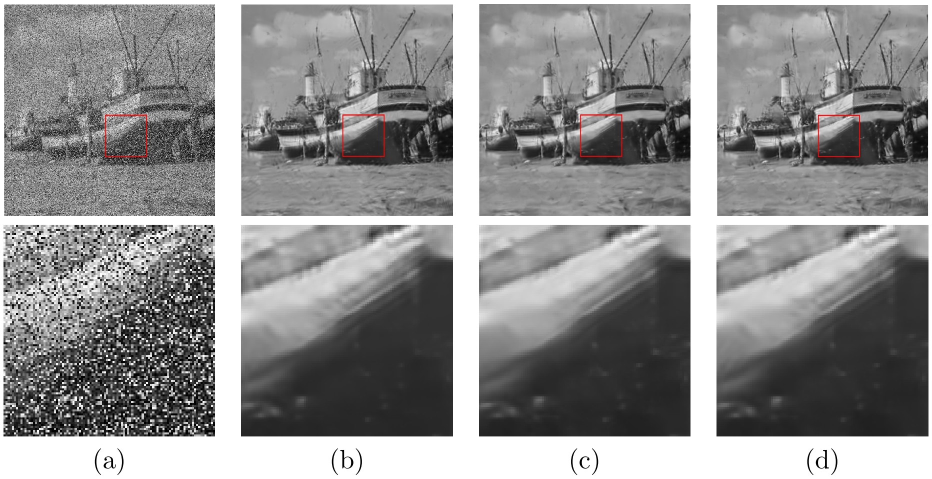


FIGURE 9. (a) The noisy "Boat" image with Gaussian noise ($\sigma = 25$) and random-valued impluse noise at level 45. The denoised images are obtained using (b) NLR-TP, (c) Algorithm 1-S1, and (d) Algorithm 1-S2. The second row displays the corresponding zoomed-in regions of the images in the first row.

Declarations.

- *Disclaimer*: Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of AFRL (Air Force Research Laboratory).

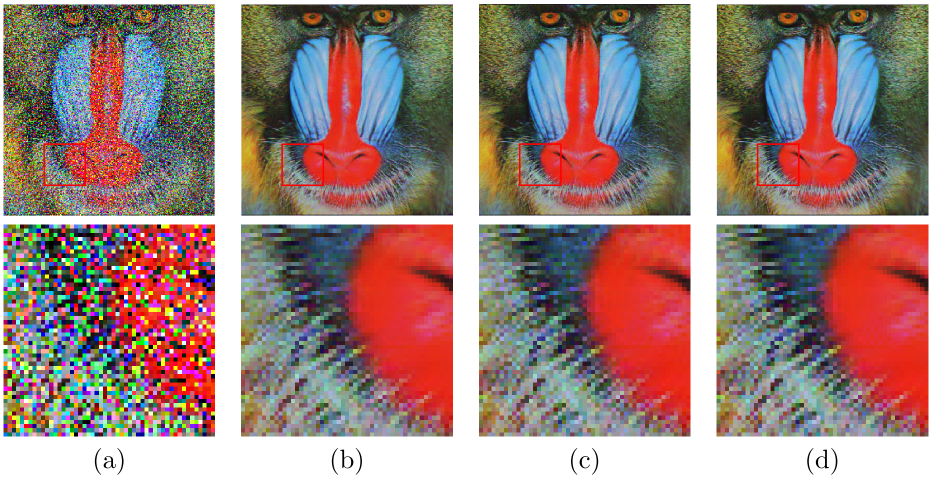


FIGURE 10. (a) The noisy "Baboon" image with Gaussian noise ($\sigma = 20$) and salt-and-pepper noise at level 30. The denoised images are obtained using (b) NLR-TP, (c) Algorithm 1-S1, and (d) Algorithm 1-S2. The second row displays the corresponding zoomed-in regions of the images in the first row.

- *Conflict of interests*: On behalf of all authors, the corresponding author states that there is no conflict of interest.

REFERENCES

[1] H. Attouch, J. Bolte and P. Redont, Proximal alternating alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality, *Mathematics of Operations Research*, **35** (2010), 438-457.

[2] 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*, Ser. A, **137** (2013), 91-129.

[3] A. Beck, *First-order Methods in Optimization*, MOS-SIAM Ser. Optim., 25, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA; Mathematical Optimization Society, Philadelphia, PA, 2017.

[4] J.-F. Cai, E. Candes and Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM Journal on Optimization*, **20** (2010), 1956-1982.

[5] J.-F. Cai, R. Chan, L. Shen and Z. Shen, Simultaneously inpainting in image and transformed domains, *Numerische Mathematik*, **112** (2009), 509-533.

[6] J.-F. Cai, R. H. Chan and M. Nikolova, Two-phase methods for deblurring images corrupted by impulse plus Gaussian noise, *Inverse Problems and Imaging*, **2** (2008), 187-204.

[7] J.-F. Cai, R. H. Chan, L. Shen and Z. Shen, Convergence analysis of tight framelet approach for missing data recovery, *Advances in Computational Mathematics*, **31** (2009), 87-113.

[8] E. Candes, M. B. Wakin and S. Boyd, Enhancing sparsity by reweighted ℓ^1 minimization, *Journal of Fourier Analysis and Applications*, **14** (2008), 877-905.

[9] K. Chen, H. Dong and K.-S. Chan, Reduced rank regression via adaptive nuclear norm penalization, *Biometrika*, **100** (2013), 901-920.

[10] T. Chen and H. R. Wu, Adaptive impulse detection using center-weighted median filters, *IEEE Signal Processing Letters*, **8** (2001), 1-3.

[11] K. Dabov, A. Foi, V. Katkovnik and K. Egiazarian, Image denoising by sparse 3D transform-domain collaborative filtering, *IEEE Transanctions on Image Processing*, **16** (2007), 2080-2095.

[12] M. Fazel, H. Hindi and S. Boyd, Log-det heuristic for matrix rank minimization with applications to Hankel and Euclidean distance matrices, *Proceedings of American Control Conference*, **3** (2003), 2156-2162.

[13] S. Gu, L. Zhang, W. Zuo and X. Feng, Weighted nuclear norm minimization with application to image denoising, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, 2862-2869.

[14] O. Güler, *Foundations of Optimization*, Grad. Texts in Math., 258, Springer, New York, 2010.

[15] T. Huang, W. Dong, X. Xie, G. Shi and X. Bai, Mixed noise removal via laplacian scale mixture modeling and nonlocal low-rank approximation, *IEEE Transactions on Image Processing*, **26** (2017), 3171-3186.

[16] H. Hwang and R. A. Haddad, Adaptive median filters: New algorithms and results, *IEEE Transactions on Image Processing*, **4** (1995), 499-502.

[17] J. Jiang, L. Zhang and J. Yang, Mixed noise removal by weighted encoding with sparse nonlocal regularization, *IEEE Transactions on Image Processing*, **23** (2014), 2651-2662.

[18] A. S. Lewis and H. S. Sendov, Nonsmooth analysis of singular values. Part I: Theory, *Set-Valued Analysis*, **13** (2005), 213-241.

[19] Y.-R. Li, L. Shen, D. Q. Dai and B. W. Suter, Framelet algorithms for de-blurring images corrupted by impulse plus gaussian noise, *IEEE Transactions on Image Processing*, **20** (2011), 1822-1837.

[20] L. Liu, L. Chen, C. L. P. Chen, Y. Y. Tang, et al., Weighted joint sparse representation for removing mixed noise in image, *IEEE Transactions on Cybernetics*, **47** (2016), 600-611.

[21] L. Mirsky, A trace inequality of John von Neumann, *Monatshefte fur Mathematik*, **79** (1975), 303-306.

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

[23] A. Prater-Bennette, L. Shen and E. E. Tripp, The proximity operator of the log-sum penalty, *Journal of Scientific Computing*, **93** (2022), Paper No. 67, 34 pp.

[24] 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*, **13** (2004), 600-612.

[25] B. Wen, X. Chen and T. K. Pong, A proximal difference-of-convex algorithm with extrapolation, *Computational Optimization and Applications*, **69** (2018), 297-324.

[26] Y. Xiao, T. Zeng, J. Yu and M. K. Ng, Restoration of images corrupted by mixed Gaussian-impulse noise via l1-l0 minimization, *Pattern Recognition*, **44** (2011), 1708-1720.

[27] C. Xu, X. Liu, J. Zheng, L. Shen, Q. Jiang and J. Lu, Nonlocal low-rank regularized two-phase approach for mixed noise removal, *Inverse Problems*, **37** (2021), Paper No. 085001, 25 pp.

[28] L. Zhang, L. Zhang, X. Mou and D. Zhang, FSIM: A feature similarity index for image quality assessment, *IEEE Transactions on Image Processing*, **20** (2011), 2378-2386.

Received July 2024; revised March 2025; early access May 2025.