SIAM J. IMAGING SCIENCES
Vol. 13, No. 3, pp. 1595–1629

Multiplicative Noise Removal: Nonlocal Low-Rank Model and Its Proximal Alternating Reweighted Minimization Algorithm[*]

Xiaoxia Liu[†], Jian Lu[‡], Lixin Shen[§], Chen Xu[¶], and Yuesheng Xu[‖]

Abstract. The goal of this paper is to develop a novel numerical method for efficient multiplicative noise removal. The nonlocal self-similarity of natural images implies that the matrices formed by their nonlocal similar patches are low-rank. By exploiting this low-rank prior with application to multiplicative noise removal, we propose a nonlocal low-rank model for this task and develop a proximal alternating reweighted minimization (PARM) algorithm to solve the optimization problem resulting from the model. Specifically, we utilize a generalized nonconvex surrogate of the rank function to regularize the patch matrices and develop a new nonlocal low-rank model, which is a nonconvex nonsmooth optimization problem having a patchwise data fidelity and a generalized nonlocal low-rank regularization term. To solve this optimization problem, we propose the PARM algorithm, which has a proximal alternating scheme with a reweighted approximation of its subproblem. A theoretical analysis of the proposed PARM algorithm is conducted to guarantee its global convergence to a critical point. Numerical experiments demonstrate that the proposed method for multiplicative noise removal significantly outperforms existing methods, such as the benchmark SAR-BM3D method, in terms of the visual quality of the denoised images, and of the peak-signal-to-noise ratio (PSNR) and the structural similarity index measure (SSIM) values.

Key words. multiplicative noise removal, nonlocal low-rank regularization, image restoration

AMS subject classifications. 68U10, 94A08, 90C26, 15A03, 46N10, 65F22

DOI. 10.1137/20M1313167

1. Introduction.

We consider the problem of multiplicative noise removal. To effectively restore images degraded by multiplicative noise, we develop a method which consists of an optimization model and an iterative algorithm to solve the minimization problem. Based

[*] Received by the editors January 31, 2020; accepted for publication (in revised form) May 18, 2020; published electronically September 15, 2020.
https://doi.org/10.1137/20M1313167

Funding: The research of the second author is partially supported by the Natural Science Foundation of China under grants 61972265 and 11871348, by the Natural Science Foundation of Guangdong Province of China under grant 2020B1515310008, and by the Educational Commission of Guangdong Province of China under grant 2019KZDZX1007. The research of the third author is partially supported by the National Science Foundation under grant DMS-1913039. The research of the fourth author is partially supported by the Natural Science Foundation of China under grant 61872429. The research of the fifth author is partially supported by the National Science Foundation under grant DMS-1912958, and by the Natural Science Foundation of China under grant 11771464.

[†] Shenzhen Key Laboratory of Advanced Machine Learning and Applications, College of Mathematics and Statistics, Shenzhen University, Shenzhen, Guangdong, 518060, P.R. China (xliu@szu.edu.cn).

[‡] Corresponding author. Shenzhen Key Laboratory of Advanced Machine Learning and Applications, College of Mathematics and Statistics, Shenzhen University, Shenzhen, Guangdong, 518060, P.R. China (jianlu@szu.edu.cn).

[§] Department of Mathematics, Syracuse University, Syracuse, NY 13244 (lshen03@syr.edu).

[¶] Shenzhen Key Laboratory of Advanced Machine Learning and Applications, College of Mathematics and Statistics, Shenzhen University, Shenzhen, Guangdong, 518060, P.R. China (chenxuszu@sina.com).

[‖] Department of Mathematics and Statistics, Old Dominion University, Norfolk, VA 23529 (y1xu@odu.edu).

on the nonlocal self-similarity of natural images, we propose a nonlocal low-rank model for multiplicative noise removal. The resulting model is a nonconvex nonsmooth minimization problem. We develop a proximal alternating reweighted minimization (PARM) algorithm with a convergence guarantee to efficiently solve the problem.

Multiplicative noise (i.e., speckle noise) widely occurs in coherent imaging systems due to the interference of coherent waves scattered from distributed targets. For example, images obtained from synthetic aperture radar (SAR) [27], ultrasound imaging [35], and laser imaging [30] are naturally contaminated with multiplicative noise. Removing multiplicative noise from such images is inevitable in many areas of applications.

Methods employed for multiplicative noise removal in the literature include the total variation (TV) regularization based models, patch based methods, and nonlocal low-rank based methods. TV regularization has been widely used to preserve edges in the restored images. In a TV regularization based model, the objective function is the sum of a data fidelity term and a TV regularization term. The data fidelity term measures the closeness between the desired image and the observed noisy image, while the TV regularization term measures the total variation of a desired image or an image in its transformed domain. The AA model [3] used the Bayesian maximum a posteriori probability (MAP) estimation to derive the data fidelity term in terms of the desired image. However, this data fidelity term is nonconvex, and the resulting optimization problem is challenging to solve. To overcome this challenge, the DZ model [10] modified the data fidelity term by adding a quadratic term. As a consequence, the objective function of the DZ model becomes convex under some mild conditions. The I-DIV model [33] used the so-called I-divergence, which was typically designed for dealing with Poisson noise [12, 31], as the data fidelity term. By performing the logarithmic transformation, the SO model [32], the HNM model [18], and the Exp model [25] led to convex, even strictly convex, data fidelity terms. The mV model [39] and the TwL-mV model [20] used convex or strongly convex data fidelity terms via the mth root transformation. The TV regularization based models have good performance in denoising. However, they tend to oversmooth image textures and generate unexpected artifacts. To reduce the staircase artifacts of traditional TV regularization based models, the total generalized variation method [11] and the nonlocal TV method [26] were also investigated for multiplicative noise removal.

The patch based methods make use of the redundancy of image patches to yield a restored image with fine details. Sparse representations of image patches have been studied in the patch based methods for multiplicative noise removal. In the learned dictionary method [17], an optimal overcomplete dictionary was learned from the patches of the logarithmic transformed noisy image, and then an image was restored via a variational model based on the learned dictionary and a TV regularization. The SAR-BM3D method [28] is another remarkable approach relying on a sparse representation, which takes advantage of the nonlocal self-similarity of natural images [5]. Nonlocal similar patches, collected as 3D groups, were identified based on a probabilistic similarity measure for multiplicative noise and then were denoised by jointly nonlocal filtering and a local linear minimum-mean-square-error shrinkage in a wavelet domain. We remark that these methods constrain the sparsity priors in either a fixed dictionary or a fixed wavelet domain, which limits their capability in multiplicative noise removal.

Recently, the nonlocal low-rank based methods were extensively exploited in image processing. It is recognized that natural images are of nonlocal self-similarity. Matrices formed by nonlocal similar patches are low-rank, and hence the desired image can be restored by low-rank estimations of nonlocal similar patch matrices. To regularize the rank of the matrices formed by nonlocal similar patches, different approximations of the rank function, including the weighted nuclear norm and the log-det function, were adopted; see, e.g., [9, 14, 16, 19, 36].

Existing studies have shown impressive empirical performance of nonlocal low-rank based methods. However, theoretical analysis of the existing methods is missing, and there is little work on applications of nonlocal low-rank based methods to multiplicative noise removal. To address this issue, we propose developing a new nonlocal low-rank based method that is theoretically and practically suitable for multiplicative noise removal. The proposed method includes a novel nonlocal low-rank model and an efficient iterative algorithm to solve the proposed model with a convergence guarantee. We explore the underlying low-rank prior of the patch matrices and propose a nonlocal low-rank model for multiplicative noise removal. The resulting optimization problem is nonconvex and nonsmooth, which made it challenging to design efficient and theoretically convergence-guaranteed algorithms to solve. In fact, the well-known alternating direction method of multipliers (ADMM) algorithm is not applicable to this optimization problem, and the alternating minimization (AM) algorithm and the augmented Lagrange multiplier (ALM) algorithm may not converge [4, 38]. To address this challenge in developing an efficient convergent algorithm, we propose a proximal alternating minimization scheme with a reweighted approximation of its subproblem and further use the Kurdyka–Łojasiewicz (KL) theory [2, 4] to prove its global convergence to a critical point. The experiments demonstrate that the proposed nonlocal low-rank based method is well suited to multiplicative noise removal.

The main contributions of this work are as follows:

- We propose a nonlocal low-rank model for multiplicative noise removal. This model is formulated in the log-transformed domain of images. The objective function of the model as the sum of a fidelity term and a regularization term is nonconvex and nonsmooth. Its fidelity term is adapted from the corresponding one in the Exp model [25] to patches and is strictly convex under certain conditions. Its regularization term is the application of the composition of the rank operator with the patch extraction operator to the underlying image. Due to the difficulties caused by the composition and the rank function in solving this model, we propose to split this composition by introducing an auxiliary variable and to approximate the rank function using a smooth concave function.
- We develop a proximal alternating reweighted minimization (PARM) algorithm for solving the proposed nonlocal low-rank model. The key to the PARM algorithm is dealing with the concave function that is used to approximate the rank function in the model. We propose approximating this concave function by its affine approximation (i.e., the reweighted approximation) in each iteration of the PARM algorithm. This approach could be useful for a wide range of nonlocal low-rank models.
- We provide a theoretical analysis of the PARM algorithm which guarantees its global convergence to a critical point, in contrast to the practically used algorithms, such as those in [9, 37], which lack convergence analysis.
- We give a detailed description of the implementation of the PARM algorithm, including

parameter settings, patch sizes, and search windows. We also test the proposed method for various images at different noise levels. Furthermore, we conduct a performance comparison of the proposed method with several existing methods for multiplicative noise removal, with respect to the visual quality of the denoised images, and the PSNR (peak-signal-to-noise ratio) and SSIM (structural similarity index measure) values.

This paper is organized into six sections. In section 2, we present the nonlocal low-rank model for multiplicative noise removal. The proposed PARM algorithm for solving the resulting nonconvex nonsmooth optimization problem is presented in section 3. Section 4 is devoted to the convergence analysis of the proposed algorithm. In section 5, we demonstrate the efficiency of the new method numerically by experimental results. Section 6 concludes this paper.

2. Nonlocal low-rank model for multiplicative noise removal.

In this section we propose a nonlocal low-rank model for multiplicative noise removal by exploiting low-rank priors of the nonlocal similar patch matrices extracted from the underlying images.

Throughout this paper, matrices are bold uppercase, vectors are bold lowercase, and scalars or entries are not bold. Given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \sum_{i=1}^{d} \langle x_i, y_i \rangle$ is the standard inner product, and $\|\boldsymbol{x}\|_2 := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle}$ is the standard ℓ_2 norm. Let $\mathbb{S}_+^d$ denote the set of symmetric positive definite matrices of size $d \times d$, and let $\boldsymbol{I}_d$ denote the identity matrix of size $d \times d$. Given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ and $\boldsymbol{H} \in \mathbb{S}_+^d$, $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_{\boldsymbol{H}} := \langle \boldsymbol{x}, \boldsymbol{H}\boldsymbol{y} \rangle$ is the $\boldsymbol{H}$-weighted inner product, and $\|\boldsymbol{x}\|_{\boldsymbol{H}} := \sqrt{\langle \boldsymbol{x}, \boldsymbol{x} \rangle_{\boldsymbol{H}}}$ is the $\boldsymbol{H}$-weighted ℓ_2 norm. Given $\boldsymbol{X}, \boldsymbol{Y} \in \mathbb{R}^{m \times n}$, $\langle \boldsymbol{X}, \boldsymbol{Y} \rangle_F := \operatorname{tr}(\boldsymbol{X}^\top \boldsymbol{Y})$ is the Frobenius inner product, and $\|\boldsymbol{X}\|_F := \sqrt{\langle \boldsymbol{X}, \boldsymbol{X} \rangle_F}$ is the Frobenius norm.

Multiplicative noise removal in this paper refers to reducing multiplicative noise in an L-look image obtained by the multilook averaging technique. An L-look image $\boldsymbol{v} \in \mathbb{R}^N$ in the intensity format degraded by multiplicative noise can be modeled as

$$\boldsymbol{v} = \boldsymbol{u}\boldsymbol{\eta},$$

where $\boldsymbol{u} \in \mathbb{R}^N$ is the desired image to be restored, $\boldsymbol{\eta} \in \mathbb{R}^N$ is the multiplicative noise, and the multiplication operation is performed componentwise. The multiplicative noise in each pixel follows a Gamma distribution [13], whose probability distribution function is defined as

$$p(\eta_i) = \frac{L^L \eta_i^{L-1}}{\Gamma(L)} e^{-L\eta_i}, \quad i = 1, 2, \dots, N,$$

which has mean 1 and variance $1/L$. A list of TV regularization based models for multiplicative noise removal is presented in Table 1.

In the following, we present our nonlocal low-rank model for multiplicative noise removal. According to the nonlocal self-similarity of natural images, for an image patch, we can find enough nonlocal similar patches across the image or within a local window [5]. This redundancy of patches is important for the effectiveness of the nonlocal low-rank regularization. We begin with collecting similar patches using block matching [8, 28] and formulating patch matrices. Suppose that $\hat{\boldsymbol{u}} \in \mathbb{R}^N$ is an estimated clean image in the intensity format and that J is the number of nonlocal similar patch groups to be collected. For the reference patch $\hat{\boldsymbol{u}}_j \in \mathbb{R}^{m_j}$ with size $\sqrt{m_j} \times \sqrt{m_j}$ in the jth patch group, we search within a local window for a

Table 1
TV regularization based models for multiplicative noise removal.[1]

Name	Model Φ	Transform.	Properties of Φ
AA [3]	$\min\limits_{\boldsymbol{x}\in\mathbb{R}_+^N} \langle \log \boldsymbol{x} + \frac{\boldsymbol{v}}{\boldsymbol{x}}, \mathbb{1}\rangle + \lambda \lVert\boldsymbol{x}\rVert_{TV}$	$\boldsymbol{x}=\boldsymbol{u}$	nonconvex
DZ [10]	$\min\limits_{\boldsymbol{x}\in\mathbb{R}_+^N} \langle \log \boldsymbol{x} + \frac{\boldsymbol{v}}{\boldsymbol{x}}, \mathbb{1}\rangle + \rho \lVert \sqrt{\frac{\boldsymbol{x}}{\boldsymbol{v}}} - \mathbb{1}\rVert_2^2 + \lambda \lVert\boldsymbol{x}\rVert_{TV}$	$\boldsymbol{x}=\boldsymbol{u}$	strictly convex if $\rho \geq \frac{2\sqrt{6}}{9}$
I-DIV [33]	$\min\limits_{\boldsymbol{x}\in\mathbb{R}_+^N} \langle \boldsymbol{x} - \boldsymbol{v}\log \boldsymbol{x}, \mathbb{1}\rangle + \lambda \lVert\boldsymbol{x}\rVert_{TV}$	$\boldsymbol{x}=\boldsymbol{u}$	convex
SO [32]	$\min\limits_{\boldsymbol{x}\in\mathbb{R}^N} \langle \boldsymbol{x} + \frac{\boldsymbol{v}}{e^{\boldsymbol{x}}}, \mathbb{1}\rangle + \lambda \lVert\boldsymbol{x}\rVert_{TV}$	$\boldsymbol{x}=\log\boldsymbol{u}$	strictly convex
HNM [18]	$\min\limits_{\boldsymbol{x}\in\mathbb{R}^N,\boldsymbol{w}\in\mathbb{R}^N} \langle \boldsymbol{x} + \frac{\boldsymbol{v}}{e^{\boldsymbol{x}}}, \mathbb{1}\rangle + \rho\lVert\boldsymbol{x}-\boldsymbol{w}\rVert_2^2 + \lambda \lVert\boldsymbol{w}\rVert_{TV}$	$\boldsymbol{x}=\log\boldsymbol{u}$	convex
Exp [25]	$\min\limits_{\boldsymbol{x}\in\mathbb{R}^N} \langle \boldsymbol{x} + \frac{\boldsymbol{v}}{e^{\boldsymbol{x}}}, \mathbb{1}\rangle + \rho \lVert \sqrt{\frac{e^{\boldsymbol{x}}}{\boldsymbol{v}}} - \gamma\mathbb{1}\rVert_2^2 + \lambda \lVert\boldsymbol{x}\rVert_{TV}$	$\boldsymbol{x}=\log\boldsymbol{u}$	strictly convex if $\rho\gamma^4 \leq \frac{4096}{27}$
mV [39]	$\min\limits_{\boldsymbol{x}\in\sqrt[m]{U}} \langle m\log \boldsymbol{x} + \frac{\boldsymbol{v}}{\boldsymbol{x}^m}, \mathbb{1}\rangle + \lambda \lVert\boldsymbol{x}\rVert_{TV}$	$\boldsymbol{x}=\sqrt[m]{\boldsymbol{u}}$	convex if m is sufficiently large
TwL-mV [20]	$\min\limits_{a>0,\boldsymbol{x}\in\sqrt[m]{U}} \frac{1}{s}\langle a, \boldsymbol{x}^s\rangle - \frac{1}{s}\langle m\log a - \frac{s\boldsymbol{v}}{\boldsymbol{x}^m}, \mathbb{1}\rangle + \lambda \lVert\boldsymbol{x}\rVert_{TV}$	$\boldsymbol{x}=\sqrt[m]{\boldsymbol{u}}$	strongly convex with respect to $\boldsymbol{x}$

[1] Φ denotes the objective function of the model. $\mathbb{R}_+ = (0,+\infty)$. $U = (0,C]^N$ and $C\in\mathbb{R}_+$. $\lambda>0$, $\rho>0$, $\gamma\geq 1$, and $s\geq 1$. $\mathbb{1}$ denotes the vector whose entries are all ones. The division, multiplication, logarithmic, exponential, and square root operations are componentwise operations.

total of n_j patches that are similar to the reference patch, assuming $m_j \leq n_j$, $j=1,2,\ldots,J$. To fully exploit the statistics of L-look images, we measure the similarity between two patches $\hat{\boldsymbol{u}}_j \in \mathbb{R}^{m_j}$ and $\hat{\boldsymbol{u}}_j' \in \mathbb{R}^{m_j}$ using the block similarity measure introduced in [28],

$$d(\hat{\boldsymbol{u}}_j, \hat{\boldsymbol{u}}_j') = (2L-1)\sum_{i=1}^{m_j} \log\left(\sqrt{\frac{(\hat{u}_j)_i}{(\hat{u}_j')_i}} + \sqrt{\frac{(\hat{u}_j')_i}{(\hat{u}_j)_i}}\right). \tag{1}$$

Following the above, for each group we construct a patch matrix from all of the patches in the given group through an extraction operator. Define $\boldsymbol{R}_{jl} \in \mathbb{R}^{m_j\times N}$ to be a binary matrix (i.e., its entries are either 1 or 0) such that $\boldsymbol{R}_{jl}\hat{\boldsymbol{u}}$ is the lth patch in the jth nonlocal similar patch group of the given estimated image $\hat{\boldsymbol{u}}$, $l=1,2,\ldots,n_j$, $j=1,2,\ldots,J$. Then we define a linear operator $R_j : \mathbb{R}^N \to \mathbb{R}^{m_j\times n_j}$ as

$$R_j(\boldsymbol{x}) := \begin{bmatrix} \boldsymbol{R}_{j1}\boldsymbol{x} & \boldsymbol{R}_{j2}\boldsymbol{x} & \cdots & \boldsymbol{R}_{jn_j}\boldsymbol{x} \end{bmatrix}.$$

Here, $R_j(\boldsymbol{x})$ is called the jth patch matrix of the (transformed) image $\boldsymbol{x}\in\mathbb{R}^N$. After the patch matrix is extracted, the patch matrix can be further processed using, for example, normalization with mean zero, and the corresponding extraction operator R_j can be defined accordingly. Intuitively, the patch matrix $R_j(\boldsymbol{x})$ with similar structures should be a low-rank matrix if $\boldsymbol{x}$ is close to the clean image $\hat{\boldsymbol{u}}$, for example, up to a transformation.

Taking advantage of the low-rank prior of image patch matrices $R_j(\boldsymbol{x})$, the objective function of a patch based nonlocal low-rank model consists of a data fidelity term to restore the desired image and a nonlocal low-rank regularization term as follows:

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) + \sum_{j=1}^{J} \lambda_j \operatorname{rank}(R_j(\boldsymbol{x})), \tag{2}$$

where $\boldsymbol{x} \in \mathbb{R}^N$ is the desired (transformed) image to be restored, $f : \mathbb{R}^N \to (-\infty, +\infty]$ is the data fidelity term that measures the closeness between the observed image and the desired image, $R_j : \mathbb{R}^N \to \mathbb{R}^{m_j \times n_j}$ is the (normalized) extraction of the jth nonlocal similar patch matrix, and $\lambda_j > 0$ is the regularization parameter, $j = 1, 2, \ldots, J$.

To restore images degraded by multiplicative noise, our proposed data fidelity term f for model (2) is formulated based on patchwise data. Let $\boldsymbol{v} \in \mathbb{R}^N$ be the given noisy image, and let $\boldsymbol{x} \in \mathbb{R}^N$ be the unknown clean log-transformed image to be restored. We extend the pixelwise data fidelity term of the Exp model [25] as shown in Table 1 to a patchwise data fidelity term that is a weighted sum of fidelity terms in terms of patch matrices $R_j(\boldsymbol{x})$ as follows:

$$f(\boldsymbol{x}) := \tau \sum_{j=1}^{J} \mu_j f_j(R_j(\boldsymbol{x})), \tag{3}$$

where each patchwise fidelity term $f_j(R_j(\boldsymbol{x}))$ is defined as

$$f_j(R_j(\boldsymbol{x})) := \left\langle R_j(\boldsymbol{x}) + \frac{R_j(\boldsymbol{v})}{e^{R_j(\boldsymbol{x})}}, R_j(\mathbb{1}_N) \right\rangle_F + \rho \left\| \sqrt{\frac{e^{R_j(\boldsymbol{x})}}{R_j(\boldsymbol{v})}} - \gamma R_j(\mathbb{1}_N) \right\|_F^2 ,$$

$\mu_j > 0$ is the weight parameter, $\tau > 0$ is the data fidelity parameter, $\mathbb{1}_N$ denotes the vector of size $N \times 1$ with all ones, and parameters $\rho > 0$ and $\gamma \geq 1$ depend on the noise level. The exponential operation, division operation, and square root operation are performed componentwise. It follows from [25] that f is strictly convex if $\rho\gamma^4 \leq \frac{4096}{27}$.

The patchwise data fidelity term f in the above can be further viewed as a weighted pixelwise data fidelity term. Define $R_j^\top : \mathbb{R}^{m_j \times n_j} \to \mathbb{R}^N$ as

$$R_j^\top(\boldsymbol{Y}) := \sum_{l=1}^{n_j} \boldsymbol{R}_{jl}^\top \boldsymbol{y}_i,$$

where $\boldsymbol{y}_i \in \mathbb{R}^{m_j}$ is the ith vector of $\boldsymbol{Y}$. Since R_j and $R_j^\top$ are linear operators such that $\langle R_j(\boldsymbol{x}), \boldsymbol{Y} \rangle_F = \langle \boldsymbol{x}, R_j^\top(\boldsymbol{Y}) \rangle$ for all $\boldsymbol{x} \in \mathbb{R}^N$ and $\boldsymbol{Y} \in \mathbb{R}^{m_j \times n_j}$, we can rewrite the above as

$$f_j(R_j(\boldsymbol{x})) = \left\langle \boldsymbol{x} + \frac{\boldsymbol{v}}{e^{\boldsymbol{x}}}, (R_j^\top \circ R_j)\mathbb{1}_N \right\rangle + \rho \left\langle \sqrt{\frac{e^{\boldsymbol{x}}}{\boldsymbol{v}}} - \gamma \mathbb{1}_N, (R_j^\top \circ R_j) \left(\sqrt{\frac{e^{\boldsymbol{x}}}{\boldsymbol{v}}} - \gamma \mathbb{1}_N \right) \right\rangle .$$

By defining $\boldsymbol{W} := \sum_{j=1}^{J} \mu_j R_j^\top \circ R_j$, its matrix representation

$$\boldsymbol{W} = \sum_{j=1}^{J} \mu_j \sum_{l=1}^{n_j} \boldsymbol{R}_{jl}^\top \boldsymbol{R}_{jl} \tag{4}$$

is a diagonal matrix whose main diagonal entries indicate the weighted counts for each pixel. Since we assume that each pixel belongs to at least one nonlocal similar patch group, $\boldsymbol{W} \in \mathbb{S}_+^N$.

Hence, f in (3) can be written as

$$f(\boldsymbol{x}) = \tau \left(\langle \boldsymbol{x} + \frac{\boldsymbol{v}}{e^{\boldsymbol{x}}}, \mathbb{1}_N \rangle_{\boldsymbol{W}} + \rho \left\| \sqrt{\frac{e^{\boldsymbol{x}}}{\boldsymbol{v}}} - \gamma \mathbb{1}_N \right\|_{\boldsymbol{W}}^2 \right). \tag{5}$$

Model (2) with the data fidelity term f as (5) regularizes low-rank priors on image patch matrices, but it is not a feasible model from both theoretical and practical perspectives. First, model (2) as a composition optimization is not easy to solve. Second, the rank function is discontinuous and nonconvex, and minimizing a problem involving the rank function is NP-hard [29]; therefore, it is challenging to solve model (2). To tackle the above challenges, we relax model (2) in the following ways. We first apply the variable splitting method to model (2) to address the composition optimization problem and then adopt a nonconvex surrogate of the rank function to replace the rank function.

First, we consider the data fidelity term f as a weighted sum defined in (3) and apply the variable splitting method to relax model (2). By introducing auxiliary (splitting) variables $\boldsymbol{Y}_j \in \mathbb{R}^{m_j \times n_j}$ such that $\boldsymbol{Y}_j = R_j(\boldsymbol{x})$ and then relaxing these equalities of the splitting variables, we obtain the following model:

$$\min_{\boldsymbol{x}, \boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_J} \sum_{j=1}^{J} \left\{ \mu_j \left(\tau f_j(R_j(\boldsymbol{x})) + \frac{1}{2} \|\boldsymbol{Y}_j - R_j(\boldsymbol{x})\|_F^2 \right) + \lambda_j \operatorname{rank}(\boldsymbol{Y}_j) \right\},$$

where $\mu_j > 0$ is the weight parameter, $\tau > 0$ is the data fidelity parameter, and $\lambda_j > 0$ is the regularization parameter, $j = 1, 2, \ldots, J$.

Second, we utilize a nonconvex relaxation of the rank function to characterize the low-rank prior of patch matrices. By introducing a function $g : [0, \infty) \to \mathbb{R}$ such that g is monotonically increasing, a generalized relaxation of the rank function is defined as

$$\|\boldsymbol{Y}\|_{*,g} := \sum_{i=1}^{m} g(\sigma_i(\boldsymbol{Y})), \tag{6}$$

where $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$, $m \leq n$, and $\sigma_i(\boldsymbol{Y})$ is the ith largest singular value of $\boldsymbol{Y}$. Here, we give two special cases of the function g. If $g(t) = 1$ for $t \neq 0$ and $g(t) = 0$ otherwise, then $\|\boldsymbol{Y}\|_{*,g}$ reduces exactly to the rank function. If $g(t) = t$ as a linear function, then $\|\boldsymbol{Y}\|_{*,g} = \|\boldsymbol{Y}\|_*$ is exactly the nuclear norm, which is the tightest convex surrogate of the rank function. However, the rank minimization is NP-hard, while the nuclear norm minimization may overshrink the singular values with large values [14].

To better approximate the rank function, we would like its nonconvex relaxation $\|\cdot\|_{*,g}$ with the function g to be monotonically increasing, concave, and smooth. For example, a decent choice of $g : [0, \infty) \to \mathbb{R}$ is the logarithmic function defined as

$$g(t) := \log(t + \varepsilon), \tag{7}$$

where $\varepsilon > 0$.

Putting all of the above discussions together, we come up with the following *nonlocal low-rank model:*

$$\min_{\boldsymbol{x},\boldsymbol{Y}_1,\dots,\boldsymbol{Y}_J} \; f(\boldsymbol{x}) + \sum_{j=1}^{J} \left(\frac{\mu_j}{2} \|\boldsymbol{Y}_j - R_j(\boldsymbol{x})\|_F^2 + \lambda_j \sum_{i=1}^{m_j} g(\sigma_i(\boldsymbol{Y}_j)) \right), \tag{8}$$

where $\boldsymbol{x} \in \mathbb{R}^N$, $\boldsymbol{Y}_j \in \mathbb{R}^{m_j \times n_j}$, $f : \mathbb{R}^N \to (-\infty, +\infty]$ is defined as in (5), $g : [0, \infty) \to \mathbb{R}$ is defined as in (7), $R_j : \mathbb{R}^N \to \mathbb{R}^{m_j \times n_j}$ is the (normalized) extraction of the jth nonlocal similar patch matrix, $m_j \le n_j$, $\mu_j > 0$, $\lambda_j > 0$, $j = 1, 2, \dots, J$.

Clearly, the objective function of model (8) is nonconvex and nonsmooth. Existing algorithms are not directly applicable to this problem. Our goal is to design theoretically convergence-guaranteed and practically efficient algorithms to solve this nonconvex nonsmooth optimization problem. In the next section, we will propose an algorithm for the nonlocal low-rank model (8), and we will analyze its convergence in section 4.

3. Proximal alternating reweighted minimization algorithm.

We present a proximal alternating reweighted minimization (PARM) algorithm for solving the nonconvex nonsmooth optimization problem.

The nonlocal low-rank model regularizes the low-rank prior of patch matrices and can also be applicable to many image restoration problems, such as image denoising and compressive sensing, if the patch matrix extraction R_j and the data fidelity term f are appropriately selected. In the following, we consider the nonlocal low-rank model in a general setting. The objective function of model (8), denoted as Φ, can be written as

$$\Phi(\boldsymbol{x}, \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_J) := f(\boldsymbol{x}) + \sum_{j=1}^{J} \Phi_j(\boldsymbol{x}, \boldsymbol{Y}_j), \tag{9}$$

where

$$\Phi_j(\boldsymbol{x}, \boldsymbol{Y}) := \frac{\mu_j}{2} \|\boldsymbol{Y} - R_j(\boldsymbol{x})\|_F^2 + \lambda_j \sum_{i=1}^{m_j} g(\sigma_i(\boldsymbol{Y})). \tag{10}$$

We further assume the following:

(A1) $f : \mathbb{R}^N \to (-\infty, +\infty]$ is inf-bounded, proper, and lower semicontinuous, i.e., $\inf f > -\infty$,

$$\operatorname{dom} f := \{\boldsymbol{x} \in \mathbb{R}^N : f(\boldsymbol{x}) < +\infty\} \ne \emptyset, \quad \text{and} \quad f(\boldsymbol{a}) \le \liminf_{\boldsymbol{x} \to \boldsymbol{a}} f(\boldsymbol{x}) \quad \forall \boldsymbol{a} \in \mathbb{R}^N;$$

(A2) $g : [0, \infty) \to \mathbb{R}$ is monotonically increasing and concave (and nonconvex), and g is continuously differentiable with an L_g-Lipschitz continuous gradient, i.e.,

$$|g'(t_1) - g'(t_2)| \le L_g |t_1 - t_2| \quad \forall t_1 \ge 0, t_2 \ge 0;$$

(A3) $\Phi(\boldsymbol{x}, \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_J)$ is coercive, i.e.,

$$\lim_{\|(\boldsymbol{x}, \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_J)\| \to \infty} \Phi(\boldsymbol{x}, \boldsymbol{Y}_1, \dots, \boldsymbol{Y}_J) = +\infty.$$

The nonlocal low-rank model (8) with the data fidelity term f defined as in (5) and the block measure for extraction operator R_j defined as in (1) can reduce multiplicative noise; model (8) with $f(\boldsymbol{x}) = \frac{\tau}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2_{\boldsymbol{W}}$, where $\boldsymbol{y}$ is the noisy image, and the block measure as Euclidean distance can reduce Gaussian noise; and model (8) with $f(\boldsymbol{x}) = \frac{\tau}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|^2_{\boldsymbol{W}}$, where $\boldsymbol{y}$ is the observed measurements and $\boldsymbol{A}$ is the measurement matrix with full row rank, and block measure as Euclidean distance can recover sparse signals in compressive sensing. It is easy to verify that each function f given above satisfies assumption (A1) and is coercive. Also, for different low-rank regularizations, g can be chosen as (7) or $g(t) = (t + \varepsilon)^p$, where $0 < p < 1$ and $\varepsilon > 0$. Either choice of g satisfies assumption (A2) and the coercivity. Note that assumptions (A1) and (A2), together with the coercivity of f and g, imply that the objective function Φ is inf-bounded and coercive. Hence, assumptions (A1)–(A3) hold for models with any combination of f and g mentioned above, including our proposed model for multiplicative noise removal.

In this general setting, no convexity or smoothness is assumed for f, and the objective function Φ of the nonlocal low-rank model (8) is nonconvex and nonsmooth. For solving this nonconvex and nonsmooth optimization problem, the alternating minimization (AM) algorithm was adopted for compressive sensing [9], and the augmented Lagrange multiplier (ALM) algorithm was adopted for speckle noise removal [37]. However, there is no guarantee that these methods will converge, because the sequence generated by the AM algorithm may cycle indefinitely without converging if the minimum in each alternating step is not uniquely obtained [4], and the sequence generated by the ALM algorithm may diverge even with bounded penalty parameters [38].

The standard approach for solving model (8) is via the proximal regularization of the Gauss–Seidel scheme (see, e.g., [4]). That is, starting with some initial point $(\boldsymbol{x}^0, \boldsymbol{Y}_1^0, \ldots, \boldsymbol{Y}_J^0)$, we generate the sequence $\{(\boldsymbol{x}^k, \boldsymbol{Y}_1^k, \ldots, \boldsymbol{Y}_J^k)\}_{k\in\mathbb{N}}$ via the scheme

$$\boldsymbol{Y}_j^{k+1} \in \operatorname*{argmin}_{\boldsymbol{Y}_j} \left\{\Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j) + \frac{\alpha_{jk}}{2}\|\boldsymbol{Y}_j - \boldsymbol{Y}_j^k\|_F^2\right\}, \quad j = 1, 2, \ldots, J;$$
$$\boldsymbol{x}^{k+1} \in \operatorname*{argmin}_{\boldsymbol{x}} \left\{\Phi(\boldsymbol{x}, \boldsymbol{Y}_1^{k+1}, \ldots, \boldsymbol{Y}_J^{k+1}) + \frac{\beta_k}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|^2_{\boldsymbol{W}}\right\},$$

where Φ, Φ_j, and $\boldsymbol{W}$ are given as in (9), (10), and (4), respectively, and $\alpha_{jk} > 0$ and $\beta_k > 0$ are parameters. The proximal terms in the above scheme use norms $\|\cdot\|_F$ and $\|\cdot\|_{\boldsymbol{W}}$, which are different from those in the scheme in [4]. To address the computational difficulty in finding $\boldsymbol{Y}_j^{k+1}$ caused by the nonconvexity of function g in Φ_j, we propose linearizing $g(\sigma_i(\boldsymbol{Y}_j))$ around $\sigma_i(\boldsymbol{Y}_j^k)$, the ith singular value of $\boldsymbol{Y}_j^k$, via $g(\sigma_i(\boldsymbol{Y}_j^k)) + (w_j^k)_i(\sigma_i(\boldsymbol{Y}_j) - \sigma_i(\boldsymbol{Y}_j^k))$, where $(w_j^k)_i = g'(\sigma_i(\boldsymbol{Y}_j^k))$ is iteratively reweighted. As a consequence, we have

$$(11) \qquad \widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j) := \frac{\mu_j}{2}\|\boldsymbol{Y}_j - R_j(\boldsymbol{x}^k)\|_F^2 + \lambda_j \sum_{i=1}^{m_j} \Big(g(\sigma_i(\boldsymbol{Y}_j^k)) + (w_j^k)_i(\sigma_i(\boldsymbol{Y}_j) - \sigma_i(\boldsymbol{Y}_j^k))\Big)$$

as a reweighted approximation of $\Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$ at $\boldsymbol{Y}_j^k$. Now, we propose an algorithm called the proximal alternating reweighted minimization (PARM) algorithm customized for model (8) as shown below:

$$\boldsymbol{Y}_j^{k+1} \in \underset{\boldsymbol{Y}_j}{\operatorname{argmin}} \left\{ \widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j) + \frac{\alpha_{jk}}{2}\|\boldsymbol{Y}_j - \boldsymbol{Y}_j^k\|_F^2 \right\}, \quad j = 1, 2, \dots, J; \tag{12}$$

$$\boldsymbol{x}^{k+1} \in \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{ \Phi(\boldsymbol{x}, \boldsymbol{Y}_1^{k+1}, \dots, \boldsymbol{Y}_J^{k+1}) + \frac{\beta_k}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|_{\boldsymbol{W}}^2 \right\}. \tag{13}$$

In (12), we utilize $\widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$, a reweighted approximation of $\Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$, to approximate the nonconvex surrogate of the rank function, which yields a closed form for (12). In (13), the proximal term is in terms of the $\boldsymbol{W}$-weighted norm, which is to be consistent with the patchwise data fidelity term f, for example, as defined in (5). In fact, we will continue to use the $\boldsymbol{W}$-weighted norm to measure the variable $\boldsymbol{x}$ throughout the entire paper. Moreover, as an algorithm for nonlocal low-rank models applied to image restoration, the PARM algorithm can be intuitively interpreted as follows. Equation (12) can be viewed as a low-rank patch matrix estimation, which returns the nonlocal patch matrices $\boldsymbol{Y}_j$ with a low-rank property, while (13) can be viewed as the image restoration step, which aggregates all of the estimated nonlocal patch matrices from (12) to form the desired image $\boldsymbol{x}$.

Before further deriving our PARM algorithm, we review some preliminaries on subdifferentials and proximity operators for nonconvex and nonsmooth functions.

3.1. Preliminaries on subdifferentials and proximity operators.

For nonconvex and nonsmooth functions, we use the following definitions for subdifferentials and proximity operators.

Definition 3.1 (subdifferentials). *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper and lower semicontinuous function, and let $\boldsymbol{H} \in \mathbb{S}_+^d$.*

(1) *For a given $\boldsymbol{x} \in \operatorname{dom} f$, the Fréchet subdifferential of f with respect to $\boldsymbol{H}$ at $\boldsymbol{x}$, written $\hat{\partial}^{\boldsymbol{H}} f(\boldsymbol{x})$, is the set of all vectors $\boldsymbol{u} \in \mathbb{R}^d$ which satisfy*

$$\liminf_{\boldsymbol{y}\neq\boldsymbol{x}\ \boldsymbol{y}\to\boldsymbol{x}} \frac{f(\boldsymbol{y}) - f(\boldsymbol{x}) - \langle \boldsymbol{u}, \boldsymbol{y} - \boldsymbol{x}\rangle_{\boldsymbol{H}}}{\|\boldsymbol{y} - \boldsymbol{x}\|_{\boldsymbol{H}}} \geq 0.$$

When $\boldsymbol{x} \notin \operatorname{dom} f$, we set $\hat{\partial}^{\boldsymbol{H}} f(\boldsymbol{x}) = \emptyset$.

(2) *The subdifferential of f with respect to $\boldsymbol{H}$ at $\boldsymbol{x} \in \mathbb{R}^d$, written $\partial^{\boldsymbol{H}} f(\boldsymbol{x})$, is defined through the following closure process:*

$$\partial^{\boldsymbol{H}} f(\boldsymbol{x}) := \{\boldsymbol{u} \in \mathbb{R}^d : \exists \boldsymbol{x}_k \to \boldsymbol{x}, f(\boldsymbol{x}_k) \to f(\boldsymbol{x}) \text{ and } \boldsymbol{u}_k \in \hat{\partial}^{\boldsymbol{H}} f(\boldsymbol{x}_k) \to \boldsymbol{u} \text{ as } k \to \infty\}.$$

Definition 3.2 (proximity operators). *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper and lower semicontinuous function such that $\inf_{\mathbb{R}^d} f > -\infty$, and let $\boldsymbol{H} \in \mathbb{S}_+^d$. The proximity operator of f with respect to $\boldsymbol{H}$ at $\boldsymbol{x} \in \mathbb{R}^d$ is defined as*

$$\operatorname{prox}_f^{\boldsymbol{H}}(\boldsymbol{x}) := \underset{\boldsymbol{u}\in\mathbb{R}^d}{\operatorname{argmin}} \left\{ f(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}\|_{\boldsymbol{H}}^2 \right\}.$$

Note that $\operatorname{prox}_f^{\boldsymbol{H}}(\boldsymbol{x})$ is a set-valued map. If f is convex, then $\operatorname{prox}_f^{\boldsymbol{H}}(\boldsymbol{x})$ is reduced to a single-valued map.

The definitions above for subdifferentials and proximity operators are defined on vectors with respect to the weighted ℓ_2 norm. When $\boldsymbol{H}$ in the above definitions is the identity matrix, the weighted ℓ_2 norm reduces to the standard ℓ_2 norm, and we simply write $\hat{\partial}^{\boldsymbol{H}} f$, $\partial^{\boldsymbol{H}} f$, and $\operatorname{prox}_f^{\boldsymbol{H}}$ as $\hat{\partial} f$, ∂f, and prox_f, respectively. The definitions for subdifferentials and proximity operators can also be extended to matrices with respect to the Frobenius norm. For the function $f : \mathbb{R}^{m\times n} \to (-\infty, +\infty]$ at a matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ with respect to the Frobenius norm, its Fréchet subdifferential is denoted as $\hat{\partial}^F f(\boldsymbol{X})$ or $\hat{\partial} f(\boldsymbol{X})$, its subdifferential is denoted as $\partial^F f(\boldsymbol{X})$ or $\partial f(\boldsymbol{X})$, and its proximity operator is denoted as $\operatorname{prox}_f^F(\boldsymbol{X})$ or $\operatorname{prox}_f(\boldsymbol{X})$.

Now, we are ready to discuss in detail the proposed PARM algorithm in (12) and (13).

3.2. Patch matrix estimation via a reweighted scheme.

To estimate low-rank patch matrices as shown in (12), we utilize $\widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$ to approximate $\Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$, a generalized rank characterization of the patch matrix $\boldsymbol{Y}_j$, via a reweighted scheme.

The reweighted approximation $\widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$ defined as (11) is derived using the fact that g is concave on $[0, \infty)$ and continuously differentiable. By the definition of the supergradient, we have

$$g(\sigma_i(\boldsymbol{Y}_j)) \leq g(\sigma_i(\boldsymbol{Y}_j^k)) + (w_j^k)_i(\sigma_i(\boldsymbol{Y}_j) - \sigma_i(\boldsymbol{Y}_j^k)), \tag{14}$$

where $\boldsymbol{w}_j^k := [(w_j^k)_1, (w_j^k)_2, \dots, (w_j^k)_{m_j}]^\top$, and $(w_j^k)_i = g'(\sigma_i(\boldsymbol{Y}_j^k))$ is iteratively reweighted, $i = 1, 2, \dots, m_j$. By replacing the term $g(\sigma_i(\boldsymbol{Y}_j))$ in $\Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$ with the right-hand side of the inequality (14), the reweighted approximation $\widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$ is immediately obtained as (11). Hence, the update of the low-rank patch matrix $\boldsymbol{Y}_j^{k+1}$ in (12) at the $(k+1)$th step can be rewritten as

$$\boldsymbol{Y}_j^{k+1} \in \operatorname*{argmin}_{\boldsymbol{Y}_j} \left\{ \lambda_j \sum_{i=1}^{m_j} (w_j^k)_i \sigma_i(\boldsymbol{Y}_j) + \frac{\mu_j + \alpha_{jk}}{2} \left\| \boldsymbol{Y}_j - \frac{\mu_j R_j(\boldsymbol{x}^k) + \alpha_{jk} \boldsymbol{Y}_j^k}{\mu_j + \alpha_{jk}} \right\|_F^2 \right\}. \tag{15}$$

Observe that (15) involves minimizing a weighted nuclear norm. The weighted nuclear norm of $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$, $m \leq n$, with the weight vector $\boldsymbol{w} := [w_1, w_2, \dots, w_m]^\top$ and $w_i \geq 0$, $i = 1, 2, \dots, m$, is defined as

$$\|\boldsymbol{Y}\|_{*,\boldsymbol{w}} := \sum_{i=1}^{m} w_i \sigma_i(\boldsymbol{Y}),$$

where $\sigma_1(\boldsymbol{Y}) \geq \sigma_2(\boldsymbol{Y}) \geq \cdots \geq \sigma_m(\boldsymbol{Y}) \geq 0$. It was proved in [7] that $\|\cdot\|_{*,\boldsymbol{w}}$ is convex if and only if $w_1 \geq w_2 \geq \cdots \geq w_m \geq 0$. In other words, for $\|\cdot\|_{*,\boldsymbol{w}}$ a convex function, the weights must increase with singular values. However, in order for large singular values to receive a lesser penalty to help reduce the bias, and for smaller singular values to receive a heavier penalty to help promote sparsity, the opposite order of the weight is desirable, i.e., $0 \leq w_1 \leq w_2 \leq \cdots \leq w_m$. Under this order of the weights, the weighted nuclear norm is a nonconvex function, and in general its proximity operator may be a set-valued map. Fortunately, the proximity operator is a single-valued map, as shown in the following lemma.

Lemma 3.3 (see [7, Theorem 2.3]). *For any $\lambda > 0$, $\boldsymbol{Y} \in \mathbb{R}^{m\times n}$, $m \leq n$, and $\boldsymbol{w} = [w_1, w_2, \dots, w_m]^\top$ with $0 \leq w_1 \leq w_2 \leq \cdots \leq w_m$,*

$$\operatorname{prox}_{\lambda\|\cdot\|_{*,\boldsymbol{w}}}(\boldsymbol{Y}) = \boldsymbol{U} S_{\lambda,\boldsymbol{w}}(\boldsymbol{\Sigma}) \boldsymbol{V}^\top,$$

where $\boldsymbol{Y} = \boldsymbol{U\Sigma V}^\top$ *is the singular value decomposition (SVD) of* $\boldsymbol{Y}$, *and* $S_{\lambda,\boldsymbol{w}}(\boldsymbol{\Sigma}) = \mathrm{diag}\{(\Sigma_{ii} - \lambda w_i)_+\}$ *is the weighted singular value thresholding (WSVT) operator.*

The assumption that g is monotonically increasing and concave implies that g' is nonnegative and monotonically decreasing. Then the weight vector $\boldsymbol{w}_j^k$ satisfies the ascending constraint, i.e., $0 \le (w_j^k)_1 \le (w_j^k)_2 \le \cdots \le (w_j^k)_{m_j}$. Hence, by Lemma 3.3, the low-rank patch matrix $\boldsymbol{Y}_j^{k+1}$ can be uniquely achieved as

$$\begin{aligned}\boldsymbol{Y}_j^{k+1} &= \mathrm{prox}_{\frac{\lambda_j}{\mu_j+\alpha_{jk}}\|\cdot\|_{*,\boldsymbol{w}_j^k}}\left(\frac{\mu_j R_j(\boldsymbol{x}^k) + \alpha_{jk}\boldsymbol{Y}_j^k}{\mu_j+\alpha_{jk}}\right)\\ &= \frac{1}{\mu_j+\alpha_{jk}}\,\mathrm{prox}_{\lambda_j\|\cdot\|_{*,\boldsymbol{w}_j^k}}\left(\mu_j R_j(\boldsymbol{x}^k) + \alpha_{jk}\boldsymbol{Y}_j^k\right)\\ &= \frac{1}{\mu_j+\alpha_{jk}}\boldsymbol{U}_j^{k+1} S_{\lambda_j,\boldsymbol{w}_j^k}(\widetilde{\boldsymbol{\Sigma}}_j^k)(\boldsymbol{V}_j^{k+1})^\top,\end{aligned}$$

where $\boldsymbol{U}_j^{k+1}\widetilde{\boldsymbol{\Sigma}}_j^k(\boldsymbol{V}_j^{k+1})^\top$ is the SVD of $\mu_j R_j(\boldsymbol{x}^k) + \alpha_{jk}\boldsymbol{Y}_j^k$.

Remark 3.4. The ascending constraint on the weight vector $\boldsymbol{w}_j^k$ may not be automatically satisfied if g is not differentiable and $(w_j^k)_i$ is chosen as a supergradient of g at $\sigma_i(\boldsymbol{Y}_j^k)$, i.e., $-(w_j^k)_i \in \partial(-g)(\sigma_i(\boldsymbol{Y}_j^k))$, as defined in [24]. For example, suppose that g is not differentiable at $\sigma_{i_0}(\boldsymbol{Y}_j^k)$, and then $\partial(-g)(\sigma_{i_0}(\boldsymbol{Y}_j^k))$ contains more than one element. If $\sigma_{i_0+1}(\boldsymbol{Y}_j^k) = \sigma_{i_0}(\boldsymbol{Y}_j^k)$, then the weights $-(w_j^k)_{i_0+1}$ and $-(w_j^k)_{i_0}$ that are selected from the same set $\partial(-g)(\sigma_{i_0}(\boldsymbol{Y}_j^k)) = \partial(-g)(\sigma_{i_0+1}(\boldsymbol{Y}_j^k))$ may have $(w_j^k)_{i_0+1} < (w_j^k)_{i_0}$ rather than an ascending order. Thus, we have to carefully select the $(w_j^k)_i$ in the case where g is not differentiable. For example, let $(w_j^k)_i = -\min\partial(-g)(\sigma_i(\boldsymbol{Y}_j^k))$.

3.3. Image restoration via W-weighted proximal regularization.

After obtaining the estimates of the low-rank patch matrices $\boldsymbol{Y}_j^{k+1}$ from the generalized rank minimization in the previous step, we may have a situation where the same pixel may have several estimated values. That is because one pixel may belong to more than one nonlocal similar patch matrix when we group nonlocal similar patches by block matching. Thus, at this image restoration step in (13) of the PARM algorithm, we aggregate all of the estimated patches to restore the entire image by minimizing the proximal regularization of $\Phi(\boldsymbol{x}, \boldsymbol{Y}_1^{k+1}, \ldots, \boldsymbol{Y}_J^{k+1})$ with respect to $\boldsymbol{x}$.

Note that the term $\sum_{j=1}^J \frac{\mu_j}{2}\|\boldsymbol{Y}_j^{k+1} - R_j(\boldsymbol{x})\|_F^2$ in $\Phi(\boldsymbol{x}, \boldsymbol{Y}_1^{k+1}, \ldots, \boldsymbol{Y}_J^{k+1})$ can be written as

$$\begin{aligned}\sum_{j=1}^J \frac{\mu_j}{2}\|\boldsymbol{Y}_j^{k+1} - R_j(\boldsymbol{x})\|_F^2 = &\sum_{j=1}^J \frac{\mu_j}{2}\|\boldsymbol{Y}_j^{k+1} - R_j(\boldsymbol{x}^k)\|_F^2 + \sum_{j=1}^J \frac{\mu_j}{2}\|R_j(\boldsymbol{x}) - R_j(\boldsymbol{x}^k)\|_F^2\\ &- \sum_{j=1}^J \mu_j\langle R_j(\boldsymbol{x}) - R_j(\boldsymbol{x}^k), \boldsymbol{Y}_j^{k+1} - R_j(\boldsymbol{x}^k)\rangle_F.\end{aligned}$$

Recall that $R_j^\top : \mathbb{R}^{m_j\times n_j} \to \mathbb{R}^N$ is defined as $R_j^\top(\boldsymbol{Y}) = \sum_{l=1}^{n_j} \boldsymbol{R}_{jl}^\top \boldsymbol{y}_i$, where $\boldsymbol{y}_i$ is the ith vector of $\boldsymbol{Y}$. Since $\langle R_j(\boldsymbol{x}), \boldsymbol{Y}\rangle_F = \langle \boldsymbol{x}, R_j^\top(\boldsymbol{Y})\rangle = \langle \boldsymbol{x}, \boldsymbol{W}^{-1}R_j^\top(\boldsymbol{Y})\rangle_{\boldsymbol{W}}$ and $\boldsymbol{W} = \sum_{j=1}^J \mu_j R_j^\top \circ$

$R_j \in \mathbb{S}_+^N$, the right-hand side of the above equality can be written as

$$\begin{aligned}&\sum_{j=1}^{J} \frac{\mu_j}{2} \|\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x})\|_F^2 \\ &\quad = \sum_{j=1}^{J} \frac{\mu_j}{2} \|\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x}^k)\|_F^2 + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|_{\boldsymbol{W}}^2 - \left\langle \boldsymbol{x} - \boldsymbol{x}^k, \sum_{j=1}^{J} \mu_j \boldsymbol{W}^{-1} R_j^\top(\mathbf{Y}_j^{k+1}) - \boldsymbol{x}^k \right\rangle_{\boldsymbol{W}}.\end{aligned}$$

The update of the estimated image $\boldsymbol{x}^{k+1}$ in (13) at the $(k+1)$th step can be rewritten as

(16)
$$\begin{aligned}\boldsymbol{x}^{k+1} &\in \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{ f(\boldsymbol{x}) - \langle \boldsymbol{x} - \boldsymbol{x}^k, \sum_{j=1}^{J} \mu_j \boldsymbol{W}^{-1} R_j^\top(\mathbf{Y}_j^{k+1}) - \boldsymbol{x}^k \rangle_{\boldsymbol{W}} + \frac{\beta_k + 1}{2} \|\boldsymbol{x} - \boldsymbol{x}^k\|_{\boldsymbol{W}}^2 \right\} \\ &= \operatorname{prox}_{\frac{1}{\beta_k+1} f}^{\boldsymbol{W}} \left(\boldsymbol{x}^k + \frac{1}{\beta_k + 1} \left(\sum_{j=1}^{J} \mu_j \boldsymbol{W}^{-1} R_j^\top(\mathbf{Y}_j^{k+1}) - \boldsymbol{x}^k \right) \right).\end{aligned}$$

The overall procedure of the PARM algorithm in (12) and (13) is summarized in Algorithm 1.

Algorithm 1. Proximal alternating reweighted minimization algorithm for model (8).

```
1:  Set parameters $\mu_j$, $\lambda_j$, $\alpha_{jk}$, and $\beta_k$
2:  Set extraction $R_j$ by block matching
3:  Compute matrix $\boldsymbol{W}$
4:  Initialize $\boldsymbol{x}^0$, $\mathbf{Y}_j^0$, and $\boldsymbol{w}_j^0$
5:  Set $k = 0$
6:  repeat
7:      for $j$ from 1 to $J$ do
8:          $[\boldsymbol{U}_j^{k+1}, \widetilde{\boldsymbol{\Sigma}}_j^k, \boldsymbol{V}_j^{k+1}] = \mathrm{SVD}\left(\mu_j R_j(\boldsymbol{x}^k) + \alpha_{jk}\mathbf{Y}_j^k\right)$        ▷ SVD
9:          $\boldsymbol{\Sigma}_j^{k+1} = \frac{1}{\mu_j+\alpha_{jk}} S_{\lambda_j, \boldsymbol{w}_j^k}(\widetilde{\boldsymbol{\Sigma}}_j^k)$        ▷ WSVT
10:         $\mathbf{Y}_j^{k+1} = \boldsymbol{U}_j^{k+1}\boldsymbol{\Sigma}_j^{k+1}(\boldsymbol{V}_j^{k+1})^\top$        ▷ Update $\mathbf{Y}_j^{k+1}$
11:         $(w_j^{k+1})_i = g'((\Sigma_j^{k+1})_{ii})$        ▷ Update $\boldsymbol{w}_j^{k+1}$
12:     end for
13:     $\boldsymbol{x}^{k+1} \in \operatorname{prox}_{\frac{1}{\beta_k+1} f}^{\boldsymbol{W}} \left(\boldsymbol{x}^k + \frac{1}{\beta_k+1}\left(\sum_{j=1}^{J} \mu_j \boldsymbol{W}^{-1} R_j^\top(\mathbf{Y}_j^{k+1}) - \boldsymbol{x}^k\right)\right)$        ▷ Update $\boldsymbol{x}^{k+1}$
14:     $k \leftarrow k + 1$
15: until stopping criterion is satisfied
```

3.4. The PARM algorithm for multiplicative noise removal. To remove multiplicative noise, we apply the PARM algorithm in Algorithm 1 to solve the nonlocal low-rank model (8), with f defined as in (5) and g defined as in (7). Accordingly, using the definitions of f and g, Algorithm 1 can be specifically implemented as follows. In line 11 of Algorithm 1,

$(w_j^{k+1})_i = \frac{1}{(\Sigma_j^{k+1})_{ii}+\varepsilon}$; in line 13, the proximity operator of $\frac{1}{\beta_k+1}f$ with respect to $\boldsymbol{W}$ can be computed using Newton's method. Given that $\rho\gamma^4 \leq \frac{4096}{27}$, the function f is strictly convex. Hence, $\mathrm{prox}^{\boldsymbol{W}}_{\frac{1}{\beta_k+1}f}$ is single-valued and defined as

$$\boldsymbol{x} = \mathrm{prox}^{\boldsymbol{W}}_{\frac{1}{\beta_k+1}f}(\widetilde{\boldsymbol{x}}) = \operatorname*{argmin}_{\boldsymbol{x}} \left\{ f(\boldsymbol{x}) + \frac{\beta_k+1}{2}\|\boldsymbol{x}-\widetilde{\boldsymbol{x}}\|^2_{\boldsymbol{W}} \right\}.$$

Since f is differentiable with respect to the $\boldsymbol{W}$-weighted ℓ_2 norm with its gradient $\nabla^{\boldsymbol{W}} f(\boldsymbol{x}) = \tau\left(\mathbb{1} - \frac{\boldsymbol{v}}{e^{\boldsymbol{x}}} + \rho\left(\frac{e^{\boldsymbol{x}}}{\boldsymbol{v}} - \gamma\sqrt{\frac{e^{\boldsymbol{x}}}{\boldsymbol{v}}}\right)\right)$, $\boldsymbol{x}$ is the unique solution of the equation

$$\nabla^{\boldsymbol{W}} f(\boldsymbol{x}) + (\beta_k+1)(\boldsymbol{x}-\widetilde{\boldsymbol{x}}) = 0,$$

and this equation can be efficiently solved by Newton's method.

4. Convergence analysis.

The aim of this section is to analyze the convergence of the PARM algorithm for model (8). The proof is motivated by the inexact descent convergence results for KL functions in [2, 4].

In what follows, we use the notation

$$\boldsymbol{Z} := (\boldsymbol{x}, \mathbf{Y}_1, \ldots, \mathbf{Y}_J) \quad \text{and} \quad \|\boldsymbol{Z}\| := \sqrt{\|\boldsymbol{x}\|^2_{\boldsymbol{W}} + \sum_{j=1}^{J} \|\mathbf{Y}_j\|_F^2},$$

and we denote by $\Phi(\boldsymbol{Z})$ the objective function of model (8).

Following are three essential conditions to guarantee convergence of the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ generated by the PARM algorithm:

(H1) Sufficient descent condition: There exists a positive constant c_1 such that for all $k \in \mathbb{N}$,

$$c_1\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\|^2 \leq \Phi(\boldsymbol{Z}^k) - \Phi(\boldsymbol{Z}^{k+1}).$$

(H2) Relative error condition: There exists a positive constant c_2 such that for all $k \in \mathbb{N}$,

$$\|\boldsymbol{A}^{k+1}\| \leq c_2\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\| \quad \text{and} \quad \boldsymbol{A}^{k+1} \in \partial\Phi(\boldsymbol{Z}^{k+1}).$$

(H3) Continuity condition: There exists a subsequence $\{\boldsymbol{Z}^{k_t}\}_{t\in\mathbb{N}}$ and $\boldsymbol{Z}^*$ such that

$$\lim_{t\to\infty} \boldsymbol{Z}^{k_t} = \boldsymbol{Z}^* \quad \text{and} \quad \lim_{t\to\infty} \Phi(\boldsymbol{Z}^{k_t}) = \Phi(\boldsymbol{Z}^*).$$

Next, let us first review the definition of the KL property of a function and recall a convergence theorem on a function having the KL property.

Definition 4.1 (Kurdyka–Łojasiewicz). *Let $f : \mathbb{R}^d \to (-\infty, +\infty]$ be proper and lower semicontinuous.*

(a) *The function f is called to have the KL property at $\tilde{\boldsymbol{x}} \in \operatorname{dom}\partial f$ if there exist $\eta \in (0, +\infty]$, a neighborhood U of $\tilde{\boldsymbol{x}}$, and a continuous function $\varphi : [0, \eta) \to [0, \infty)$ such that*

(i) *$\varphi(0) = 0$;*

(ii) *φ is C^1 on $(0, \eta)$ and continuous at 0;*

(iii) *for all $s \in (0, \eta)$, $\varphi'(s) > 0$;*

(iv) *for all $\boldsymbol{x} \in U \cap \{\boldsymbol{x} \in \mathbb{R}^d : f(\tilde{\boldsymbol{x}}) < f(\boldsymbol{x}) < f(\tilde{\boldsymbol{x}}) + \eta\}$, the following KL inequality holds:*

$$\varphi'(f(\boldsymbol{x}) - f(\tilde{\boldsymbol{x}}))\, dist(\mathbf{0}, \partial f(\boldsymbol{x})) \geq 1.$$

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

Theorem 4.2 (see [2, Theorem 2.9]). *Let $\Phi : \mathbb{R}^d \to (-\infty, +\infty]$ be a proper lower semicontinuous function. Consider a sequence $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$ that satisfies conditions* (H1)–(H3)*. If Φ has the KL property at the limit point $\boldsymbol{x}^*$ specified in* (H3)*, then the sequence $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$ converges to $\boldsymbol{x}^*$, and $\boldsymbol{x}^*$ is a critical point of Φ. Moreover, the sequence $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$ has a finite length, i.e.,*

$$\sum_{k=0}^{\infty} \|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\| < \infty.$$

The KL theory is a powerful tool for nonconvex nonsmooth optimization problems, and KL functions are ubiquitous. For example, real polynomial functions, logarithmic functions, exponential functions, $\|\cdot\|_0$, and $\|\cdot\|_p$ (where $p > 0$ is rational) are KL functions [1, 2]. Also, the function $\|\cdot\|_{*,g}$ defined as in (6), with g a logarithmic or exponential function, is a KL function [34]. For more examples of KL functions, see [1, 2, 4, 34] and references therein.

In the following, we prove that the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ generated by the PARM algorithm satisfies conditions (H1)–(H3), and then conclude that $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ converges to a critical point of Φ using the fact that Φ is a KL function. To this end, we need the following assumption on the parameters α_{jk} and β_k in the PARM algorithm:

(A4) For the sequences $\{\alpha_{jk}\}_{k\in\mathbb{N}}$, $j = 1, 2, \ldots, J$, and the sequence $\{\beta_k\}_{k\in\mathbb{N}}$, there exist positive constants α_-, α_+, β_-, β_+ such that

$$\inf\{\alpha_{jk} : k \in \mathbb{N}, j = 1, 2, \ldots, J\} \geq \alpha_- \quad \text{and} \quad \inf\{\beta_k : k \in \mathbb{N}\} \geq \beta_-,$$
$$\sup\{\alpha_{jk} : k \in \mathbb{N}, j = 1, 2, \ldots, J\} \leq \alpha_+ \quad \text{and} \quad \sup\{\beta_k : k \in \mathbb{N}\} \leq \beta_+.$$

4.1. Sufficient descent condition.

We show that the objective function Φ of model (8) evaluated at $\boldsymbol{Z}^k$, denoted $\Phi(\boldsymbol{Z}^k)$, decreases sufficiently as k increases.

Proposition 4.3 (sufficient descent condition). *Suppose that the objective function Φ of model* (8) *satisfies assumptions* (A1)–(A3)*. Let $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the PARM algorithm provided that assumption* (A4) *holds. Then $\{\Phi(\boldsymbol{Z}^k)\}_{k\in\mathbb{N}}$ is strictly decreasing and, in particular, there exists a positive constant c_1 such that for all $k \in \mathbb{N}$,*

$$c_1 \|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\|^2 \leq \Phi(\boldsymbol{Z}^k) - \Phi(\boldsymbol{Z}^{k+1}). \tag{17}$$

Proof. Let $\Phi_j(\boldsymbol{x}, \boldsymbol{Y})$ be defined as in (10), and let $\widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$ be defined as in (11). Then, according to the concavity of g illustrated in inequality (14), $\Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^{k+1})$ and its reweighted approximation $\widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j)$ have the following relationship:

$$\Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^{k+1}) \leq \widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^{k+1}) \quad \text{and} \quad \Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^k) = \widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^k).$$

Thus, the objective function Φ in (9) evaluated at $\boldsymbol{x}^k$ and $\boldsymbol{Y}_j^{k+1}$ can be rewritten as

$$\Phi(\boldsymbol{x}^k, \boldsymbol{Y}_1^{k+1}, \ldots, \boldsymbol{Y}_J^{k+1}) = f(\boldsymbol{x}^k) + \sum_{j=1}^{J} \Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^{k+1}) \leq f(\boldsymbol{x}^k) + \sum_{j=1}^{J} \widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^{k+1}).$$

By the update of $\boldsymbol{Y}_j^{k+1}$ in (12), we have

$$\widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^{k+1}) + \frac{\alpha_{jk}}{2}\|\boldsymbol{Y}_j^{k+1} - \boldsymbol{Y}_j^k\|_F^2 \leq \widetilde{\Phi}_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^k) = \Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^k).$$

Combining the two inequalities above, we have the following inequality on $\Phi(\boldsymbol{Z}^k)$ and $\Phi(\boldsymbol{x}^k, \boldsymbol{Y}_1^{k+1}, \ldots, \boldsymbol{Y}_J^{k+1})$:

$$\begin{aligned}\Phi(\boldsymbol{x}^k, \boldsymbol{Y}_1^{k+1}, \ldots, \boldsymbol{Y}_J^{k+1}) &\leq f(\boldsymbol{x}^k) + \sum_{j=1}^{J} \Phi_j(\boldsymbol{x}^k, \boldsymbol{Y}_j^k) - \sum_{j=1}^{J} \frac{\alpha_{jk}}{2}\|\boldsymbol{Y}_j^{k+1} - \boldsymbol{Y}_j^k\|_F^2 \\ &= \Phi(\boldsymbol{Z}^k) - \sum_{j=1}^{J} \frac{\alpha_{jk}}{2}\|\boldsymbol{Y}_j^{k+1} - \boldsymbol{Y}_j^k\|_F^2.\end{aligned}$$

By the update of $\boldsymbol{x}^{k+1}$ in (13), we have

$$\Phi(\boldsymbol{Z}^{k+1}) \leq \Phi(\boldsymbol{x}^k, \boldsymbol{Y}_1^{k+1}, \ldots, \boldsymbol{Y}_J^{k+1}) - \frac{\beta_k}{2}\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_{\boldsymbol{W}}^2.$$

Combining the two inequalities above, we have that $\Phi(\boldsymbol{Z}^{k+1})$ and $\Phi(\boldsymbol{Z}^k)$ satisfy the following inequality:

$$\frac{\beta_k}{2}\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_{\boldsymbol{W}}^2 + \sum_{j=1}^{J} \frac{\alpha_{jk}}{2}\|\boldsymbol{Y}_j^{k+1} - \boldsymbol{Y}_j^k\|_F^2 \leq \Phi(\boldsymbol{Z}^k) - \Phi(\boldsymbol{Z}^{k+1}).$$

Equation (17) holds with $c_1 = \frac{1}{2}\min\{\beta_-, \alpha_-\} > 0$, and $\{\Phi(\boldsymbol{Z}^k)\}_{k\in\mathbb{N}}$ is strictly decreasing. Here, β_- and α_- are the two positive parameters given in assumption (A4). ■

The sufficient descent condition proved in Proposition 4.3 immediately yields the following corollary.

Corollary 4.4. *Suppose that the objective function Φ of model* (8) *satisfies assumptions* (A1)–(A3). *Let $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the PARM algorithm provided that assumption* (A4) *holds. Then*

$$\lim_{k\to\infty} \|\boldsymbol{Z}^k - \boldsymbol{Z}^{k+1}\| = 0.$$

Proof. Summing inequality (17) from $k = 0$ to $k = K - 1$, we have

$$c_1 \sum_{k=0}^{K-1} \|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\|^2 \leq \Phi(\boldsymbol{Z}^0) - \Phi(\boldsymbol{Z}^K) \leq \Phi(\boldsymbol{Z}^0) - \Phi_{\text{inf}},$$

where $\Phi_{\text{inf}} = \inf_{\boldsymbol{Z}} \Phi(\boldsymbol{Z}) > -\infty$.

Taking $K \to \infty$, we have

$$\sum_{k=0}^{\infty} \|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\|^2 < \infty,$$

which implies $\lim_{k\to\infty} \|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\| = 0$. ■

4.2. Relative error condition. Before proving that a subgradient of Φ at $\boldsymbol{Z}^{k+1}$ is bounded above by the iterates gap, we first characterize the subdifferential of Φ.

Recall that the variable $\boldsymbol{x}$ is measured in terms of the $\boldsymbol{W}$-weight ℓ_2 norm and that the variables $\boldsymbol{Y}_j$ are measured in terms of the Frobenius norm. Then using the notation introduced in subsection 3.1 we define the subdifferential of Φ as

$$\partial\Phi(\boldsymbol{Z}) := \left\{(\boldsymbol{A}_{\boldsymbol{x}}, \boldsymbol{A}_{\boldsymbol{Y}_1}, \dots, \boldsymbol{A}_{\boldsymbol{Y}_J}) : \boldsymbol{A}_{\boldsymbol{x}} \in \partial_{\boldsymbol{x}}^{\boldsymbol{W}}\Phi(\boldsymbol{Z}), \boldsymbol{A}_{\boldsymbol{Y}_j} \in \partial_{\boldsymbol{Y}_j}\Phi(\boldsymbol{Z}), j = 1, 2, \dots, J\right\},$$

where $\partial_{\boldsymbol{x}}^{\boldsymbol{W}}\Phi(\boldsymbol{Z})$ is the partial subdifferential of Φ with respect to the variable $\boldsymbol{x}$ and with respect to the $\boldsymbol{W}$-weight ℓ_2 norm, and $\partial_{\boldsymbol{Y}_j}\Phi(\boldsymbol{Z})$ is the partial subdifferential of Φ with respect to the variable $\boldsymbol{Y}_j$ and with respect to the Frobenius norm.

By the definition of Φ in model (8) and the fact that

$$\sum_{j=1}^{J} \frac{\mu_j}{2}\|\boldsymbol{Y}_j - R_j(\boldsymbol{x})\|_F^2 = \frac{1}{2}\left\langle \boldsymbol{x}, \boldsymbol{x} - 2\sum_{j=1}^{J}\mu_j \boldsymbol{W}^{-1}R_j^\top(\boldsymbol{Y}_j)\right\rangle_{\boldsymbol{W}} + \sum_{j=1}^{J}\frac{\mu_j}{2}\|\boldsymbol{Y}_j\|_F^2,$$

we have

$$\partial_{\boldsymbol{x}}^{\boldsymbol{W}}\Phi(\boldsymbol{Z}) = \partial^{\boldsymbol{W}} f(\boldsymbol{x}) + \boldsymbol{x} - \sum_{j=1}^{J}\mu_j \boldsymbol{W}^{-1}R_j^\top(\boldsymbol{Y}_j)$$

and

$$\partial_{\boldsymbol{Y}_j}\Phi(\boldsymbol{Z}) = \mu_j(\boldsymbol{Y}_j - R_j(\boldsymbol{x})) + \lambda_j \partial\left(\sum_{i=1}^{m_j} g\circ\sigma_i\right)(\boldsymbol{Y}_j).$$

To compute the subdifferential of the singular value function $\sum_{i=1}^{m_j} g\circ\sigma_i$ and further characterize $\partial_{\boldsymbol{Y}_j}\Phi(\boldsymbol{Z})$, we recall some definitions and a lemma on singular value functions introduced in [21, 22].

Definition 4.5. *A function $f : \mathbb{R}^n \to \mathbb{R}$ is absolutely symmetric if*

$$f(x_1, x_2, \dots, x_n) = f(|x_{\pi(1)}|, |x_{\pi(2)}|, \dots, |x_{\pi(n)}|)$$

for any permutation π.

Definition 4.6. *A function $F : \mathbb{R}^{m\times n} \to \mathbb{R}$, $m \le n$, is a singular value function if $F(\boldsymbol{X}) = (f\circ\sigma)(\boldsymbol{X})$, where $f : \mathbb{R}^m \to \mathbb{R}$ is an absolutely symmetric function, $\sigma(\boldsymbol{X}) = [\sigma_1(\boldsymbol{X}), \sigma_2(\boldsymbol{X}), \dots, \sigma_m(\boldsymbol{X})]^\top$, and $\sigma_i(\boldsymbol{X})$ is the ith largest singular value of $\boldsymbol{X}$.*

The function $\sum_{i=1}^{m} g\circ\sigma_i$ can be viewed as a singular value function of the form

$$\left(\sum_{i=1}^{m} g\circ\sigma_i\right)(\boldsymbol{Y}) = (\widetilde{g}\circ\sigma)(\boldsymbol{Y}),$$

where $\widetilde{g} : \mathbb{R}^m \to \mathbb{R}$ is defined as $\widetilde{g}(\boldsymbol{t}) = \sum_{i=1}^{m} g(|t_i|)$ and is absolutely symmetric.

Lemma 4.7. *The subdifferential of a singular value function $f \circ \sigma$ at $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ is given by the formula*

$$\partial(f\circ\sigma)(\boldsymbol{X}) = \left\{\boldsymbol{U}\operatorname{diag}(\boldsymbol{d})\boldsymbol{V}^\top : \boldsymbol{d}\in\partial f(\sigma(\boldsymbol{X})), (\boldsymbol{U},\boldsymbol{V})\in\mathcal{M}(\boldsymbol{X})\right\},$$

where $\mathcal{M}(\boldsymbol{X}) := \left\{(\boldsymbol{U},\boldsymbol{V})\in\mathbb{R}^{m\times l}\times\mathbb{R}^{n\times l} : \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}, \boldsymbol{X} = \boldsymbol{U}\operatorname{diag}(\sigma(\boldsymbol{X}))\boldsymbol{V}^\top\right\}$.

By Lemma 4.7, the subdifferential of $\sum_{i=1}^m g\circ\sigma_i$ at $\boldsymbol{Y}\in\mathbb{R}^{m\times n}$ can be computed as follows:

$$\partial\left(\sum_{i=1}^m g\circ\sigma_i\right)(\boldsymbol{Y}) = \{\boldsymbol{U}\operatorname{diag}(\boldsymbol{d})\boldsymbol{V}^\top : d_i = c_i g'[\sigma_i(\boldsymbol{Y})], c_i\in\partial|\cdot|(\sigma_i(\boldsymbol{Y})), i = 1,2,\ldots,m,$$
$$(\boldsymbol{U},\boldsymbol{V})\in\mathcal{M}(\boldsymbol{Y})\},$$

where

$$\partial|\cdot|(\sigma_i(\boldsymbol{Y})) = \begin{cases}\{1\} & \text{if } \sigma_i(\boldsymbol{Y})>0,\\ [-1,1] & \text{if } \sigma_i(\boldsymbol{Y}) = 0.\end{cases}$$

Now we are ready to derive a subgradient of Φ at $\boldsymbol{Z}^{k+1}$ using the next lemma and to prove that it is bounded above.

Lemma 4.8. *Suppose that the objective function Φ of model* (8) *satisfies assumptions* (A1)–(A3)*. Let $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the PARM algorithm provided that assumption* (A4) *holds. Let $\boldsymbol{U}_j^{k+1}\boldsymbol{\Sigma}_j^{k+1}(\boldsymbol{V}_j^{k+1})^\top$ be the SVD of $\mathbf{Y}_j^{k+1}$. Then, for each k and each j, there exists $\boldsymbol{c}_j^{k+1}\in\mathbb{R}^{m_j}$ such that*

$$(c_j^{k+1})_i \in \partial|\cdot|(\sigma_i(\mathbf{Y}_j^{k+1})), \quad i = 1,2,\ldots,m_j, \tag{18}$$

and

$$\lambda_j\boldsymbol{U}_j^{k+1}\operatorname{diag}(\boldsymbol{d}_j^{k+1})(\boldsymbol{V}_j^{k+1})^\top = -\alpha_{jk}(\mathbf{Y}_j^{k+1}-\mathbf{Y}_j^k) - \mu_j(\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x}^k)), \tag{19}$$

where $\operatorname{diag}(\boldsymbol{d}_j^{k+1}) = \operatorname{diag}(\boldsymbol{c}_j^{k+1})\operatorname{diag}(\boldsymbol{w}_j^k)$.

Proof. According to the update of $\mathbf{Y}_j^{k+1}$ in (15), we have

$$0\in\mu_j(\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x}^k)) + \lambda_j\partial\|\cdot\|_{*,\boldsymbol{w}_j^k}(\mathbf{Y}_j^{k+1}) + \alpha_{jk}(\mathbf{Y}_j^{k+1} - \mathbf{Y}_j^k).$$

Since the weighted nuclear norm $\|\cdot\|_{*,\boldsymbol{w}}$ is a singular value function, by Lemma 4.7 the subdifferential of $\|\cdot\|_{*,\boldsymbol{w}}$ can be computed as follows:

$$\partial\|\cdot\|_{*,\boldsymbol{w}}(\boldsymbol{Y}) = \left\{\boldsymbol{U}\operatorname{diag}(\boldsymbol{d})\boldsymbol{V}^\top : d_i = c_i w_i, c_i\in\partial|\cdot|(\sigma_i(\boldsymbol{Y})), i = 1,\ldots,m, (\boldsymbol{U},\boldsymbol{V})\in\mathcal{M}(\boldsymbol{Y})\right\}.$$

Note that $(\boldsymbol{U}_j^{k+1},\boldsymbol{V}_j^{k+1})\in\mathcal{M}(\mathbf{Y}_j^{k+1})$. Thus, there exists $\boldsymbol{c}_j^{k+1}\in\mathbb{R}^{m_j}$ such that (18) holds and

$$-\alpha_{jk}(\mathbf{Y}_j^{k+1}-\mathbf{Y}_j^k) - \mu_j(\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x}^k)) = \lambda_j\boldsymbol{U}_j^{k+1}\operatorname{diag}(\boldsymbol{d}_j^{k+1})(\boldsymbol{V}_j^{k+1})^\top \in \lambda_j\partial\|\cdot\|_{*,\boldsymbol{w}_j^k}(\mathbf{Y}_j^{k+1}),$$

where $\operatorname{diag}(\boldsymbol{d}_j^{k+1}) = \operatorname{diag}(\boldsymbol{c}_j^{k+1})\operatorname{diag}(\boldsymbol{w}_j^k)$. ■

Proposition 4.9 (relative error condition). *Suppose that the objective function* Φ *of model* (8) *satisfies assumptions* (A1)–(A3). *Let* $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ *be the sequence generated by the PARM algorithm provided that assumption* (A4) *holds. Let* $\boldsymbol{U}_j^{k+1}\boldsymbol{\Sigma}_j^{k+1}(\boldsymbol{V}_j^{k+1})^\top$ *be the SVD of* $\mathbf{Y}_j^{k+1}$, *and let* $\boldsymbol{c}_j^{k+1}$ *and* $\boldsymbol{d}_j^{k+1}$ *be in* $\mathbb{R}^{m_j}$ *satisfying* (18) *and* (19).

Define $\boldsymbol{A}^{k+1} := (\boldsymbol{A}_{\boldsymbol{x}}^{k+1}, \boldsymbol{A}_{\mathbf{Y}_1}^{k+1}, \dots, \boldsymbol{A}_{\mathbf{Y}_J}^{k+1})$, *where*

$$\boldsymbol{A}_{\boldsymbol{x}}^{k+1} := \beta_k(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k) \tag{20}$$

and

$$\boldsymbol{A}_{\mathbf{Y}_j}^{k+1} := \mu_j(R_j(\boldsymbol{x}^k) - R_j(\boldsymbol{x}^{k+1})) + \lambda_j \boldsymbol{U}_j^{k+1}\operatorname{diag}(\widetilde{\boldsymbol{d}}_j^{k+1} - \boldsymbol{d}_j^{k+1})(\boldsymbol{V}_j^{k+1})^\top - \alpha_{jk}(\mathbf{Y}_j^{k+1} - \mathbf{Y}_j^k), \tag{21}$$

where $\operatorname{diag}(\widetilde{\boldsymbol{d}}_j^{k+1}) = \operatorname{diag}(\boldsymbol{c}_j^{k+1})\operatorname{diag}(\boldsymbol{w}_j^{k+1})$.

Then the following assertions hold for all $k \in \mathbb{N}$,

(a) $\boldsymbol{A}^{k+1} \in \partial\Phi(\boldsymbol{Z}^{k+1})$,

(b) $\|\boldsymbol{A}^{k+1}\| \le c_2\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\|$ *for some* $c_2 > 0$.

Proof. (a) According to the update of $\boldsymbol{x}^{k+1}$ in (16), we have

$$0 \in \partial^{\boldsymbol{W}} f(\boldsymbol{x}^{k+1}) + \boldsymbol{x}^k - \sum_{j=1}^{J}\mu_j \boldsymbol{W}^{-1}R_j^\top(\mathbf{Y}_j^{k+1}) + (\beta_k + 1)(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k).$$

Then the definition of $A_{\boldsymbol{x}}^{k+1}$ in (20) implies

$$\boldsymbol{A}_{\boldsymbol{x}}^{k+1} \in \partial^{\boldsymbol{W}} f(\boldsymbol{x}^{k+1}) + \boldsymbol{x}^{k+1} - \sum_{j=1}^{J}\mu_j \boldsymbol{W}^{-1}R_j^\top(\mathbf{Y}_j^{k+1}) = \partial_{\boldsymbol{x}}^{\boldsymbol{W}}\Phi(\boldsymbol{Z}^{k+1}).$$

Also, for each j, the definition of $\boldsymbol{A}_{\mathbf{Y}_j}^{k+1}$ in (21) and Lemma 4.8 imply

$$\begin{aligned}
\boldsymbol{A}_{\mathbf{Y}_j}^{k+1} &= \mu_j(\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x}^{k+1})) + \lambda_j \boldsymbol{U}_j^{k+1}\operatorname{diag}(\widetilde{\boldsymbol{d}}_j^{k+1})(\boldsymbol{V}_j^{k+1})^\top \\
&\in \mu_j(\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x}^{k+1})) + \lambda_j\partial\|\cdot\|_{*,\boldsymbol{w}_j^{k+1}}(\mathbf{Y}_j^{k+1}) \\
&= \mu_j(\mathbf{Y}_j^{k+1} - R_j(\boldsymbol{x}^{k+1})) + \lambda_j\partial\left(\sum_{i=1}^{m_j} g\circ\sigma_i\right)(\mathbf{Y}_j^{k+1}) \\
&= \partial_{\mathbf{Y}_j}\Phi(\boldsymbol{Z}^{k+1}).
\end{aligned}$$

(b) It follows from the Cauchy–Schwarz inequality that

$$\|\boldsymbol{A}^{k+1}\| \le \|\boldsymbol{A}_{\boldsymbol{x}}^{k+1}\|_{\boldsymbol{W}} + \sum_{j=1}^{J}\|\boldsymbol{A}_{\mathbf{Y}_j}^{k+1}\|_F,$$

where $\|\boldsymbol{A}_{\boldsymbol{x}}^{k+1}\|_{\boldsymbol{W}} = \beta_k\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|_{\boldsymbol{W}}$ and

$$\begin{aligned}
\|\boldsymbol{A}_{\mathbf{Y}_j}^{k+1}\|_F \le &\mu_j\|R_j(\boldsymbol{x}^k) - R_j(\boldsymbol{x}^{k+1})\|_F + \lambda_j\|\boldsymbol{U}_j^{k+1}\operatorname{diag}(\widetilde{\boldsymbol{d}}_j^{k+1} - \boldsymbol{d}_j^{k+1})(\boldsymbol{V}_j^{k+1})^\top\|_F \\
&+ \alpha_{jk}\|\mathbf{Y}_j^{k+1} - \mathbf{Y}_j^k\|_F.
\end{aligned}$$

The right-hand side of the above inequality can be computed term by term as follows. The square of the first term is bounded above by the square of the weighted iterates of the variable $\boldsymbol{x}$,

$$\mu_j^2\|R_j(\boldsymbol{x}^k)-R_j(\boldsymbol{x}^{k+1})\|_F^2 \le \mu_j \sum_{j=1}^{J} \mu_j \|R_j(\boldsymbol{x}^k-\boldsymbol{x}^{k+1})\|_F^2 = \mu_j\|\boldsymbol{x}^k-\boldsymbol{x}^{k+1}\|_{\boldsymbol{W}}^2.$$

This implies that $\mu_j\|R_j(\boldsymbol{x}^k)-R_j(\boldsymbol{x}^{k+1})\|_F \le \sqrt{\mu_j}\|\boldsymbol{x}^k-\boldsymbol{x}^{k+1}\|_{\boldsymbol{W}}$.

Also, the second term is bounded above by the iterates of the variable $\mathbf{Y}_j$. Since $\|\widetilde{\boldsymbol{d}}_j^{k+1}-\boldsymbol{d}_j^{k+1}\|_2 \le \|\widetilde{\boldsymbol{d}}_j^{k+1}-\boldsymbol{d}_j^{k+1}\|_1$, we have

$$\begin{aligned}\lambda_j\|\boldsymbol{U}_j^{k+1}\operatorname{diag}(\widetilde{\boldsymbol{d}}_j^{k+1}-\boldsymbol{d}_j^{k+1})(\boldsymbol{V}_j^{k+1})^\top\|_F &= \lambda_j\|\widetilde{\boldsymbol{d}}_j^{k+1}-\boldsymbol{d}_j^{k+1}\|_2\\ &\le \lambda_j\sum_{i=1}^{m_j}\left|(c_j^{k+1})_i\right|\left|(w_j^{k+1})_i-(w_j^k)_i\right|\\ &\le \lambda_j\sum_{i=1}^{m_j}\left|g'(\sigma_i(\mathbf{Y}_j^{k+1}))-g'(\sigma_i(\mathbf{Y}_j^k))\right|.\end{aligned}$$

Using the condition that g' is L_g-Lipschitz continuous, we further obtain

$$\begin{aligned}\lambda_j\|\boldsymbol{U}_j^{k+1}\operatorname{diag}(\widetilde{\boldsymbol{d}}_j^{k+1}-\boldsymbol{d}_j^{k+1})(\boldsymbol{V}_j^{k+1})^\top\|_F &\le \lambda_j\sum_{i=1}^{m_j}L_g|\sigma_i(\mathbf{Y}_j^{k+1})-\sigma_i(\mathbf{Y}_j^k)|\\ &\le \lambda_j m_j L_g\|\mathbf{Y}_j^{k+1}-\mathbf{Y}_j^k\|_F,\end{aligned}$$

where the last line follows from Theorem 3.3.16 in [15] and $\|\mathbf{Y}_j^{k+1}-\mathbf{Y}_j^k\|_2 \le \|\mathbf{Y}_j^{k+1}-\mathbf{Y}_j^k\|_F$.

Therefore, combining all of the inequalities above, we obtain

$$\begin{aligned}\|\boldsymbol{A}^{k+1}\| &\le (\beta_k+M_\mu)\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^k\|_{\boldsymbol{W}}+\sum_{j=1}^{J}(\lambda_j m_j L_g+\alpha_{jk})\|\mathbf{Y}_j^{k+1}-\mathbf{Y}_j^k\|_F\\ &\le c_2\|\boldsymbol{Z}_j^{k+1}-\boldsymbol{Z}_j^k\|,\end{aligned}$$

where $M_\mu=\sum_{j=1}^{J}\sqrt{\mu_j}$ and $c_2=\max\{\beta_+ + M_\mu, \lambda_1 m_1 L_g+\alpha_+,\ldots,\lambda_J m_J L_g+\alpha_+\}$. ■

The relative error condition proved in Proposition 4.9 immediately yields the following corollary.

Corollary 4.10. *Suppose that the objective function Φ of model* (8) *satisfies assumptions* (A1)–(A3). *Let $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the PARM algorithm provided that assumption* (A4) *holds. Define $\boldsymbol{A}^{k+1}=(\boldsymbol{A}_{\boldsymbol{x}}^{k+1},\boldsymbol{A}_{\mathbf{Y}_1}^{k+1},\ldots,\boldsymbol{A}_{\mathbf{Y}_J}^{k+1})$, where $\boldsymbol{A}_{\boldsymbol{x}}^{k+1}$ is defined as* (20) *and $\boldsymbol{A}_{\mathbf{Y}_j}^{k+1}$ is defined as in* (21), *with $j=1,2,\ldots,J$. Then*

$$\lim_{k\to\infty}\|\boldsymbol{A}^{k+1}\|=0.$$

Proof. The result follows immediately from Proposition 4.9 and Corollary 4.4. ■

4.3. Continuity condition.

We first show the existence of a limit point of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ using the boundedness of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ and then prove a continuity condition for any convergent subsequence of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$, which implies condition (H3).

Proposition 4.11. *Suppose that the objective function Φ of model (8) satisfies Assumptions (A1)–(A3). Let $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the PARM algorithm provided that assumption (A4) holds. Let $\mathcal{S}$ denote the set of all limit points of the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$. Then the following assertions hold:*

(a) $\mathcal{S} \neq \emptyset$;

(b) *if $\{\boldsymbol{Z}^{k_t}\}_{t\in\mathbb{N}}$ is a subsequence of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ such that $\lim_{t\to\infty} \boldsymbol{Z}^{k_t} = \boldsymbol{Z}^* \in \mathcal{S}$, then*

$$\lim_{t\to\infty} \Phi(\boldsymbol{Z}^{k_t}) = \Phi(\boldsymbol{Z}^*).$$

Proof. (a) We show that $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ is bounded by contradiction.

Assume to the contrary that there exists a subsequence $\{\boldsymbol{Z}^{k_l}\}_{l\in\mathbb{N}}$ such that $\|\boldsymbol{Z}^{k_l}\| \to \infty$ as $l \to \infty$. According to assumption (A3), Φ is coercive, and then $\Phi(\boldsymbol{Z}^{k_l}) \to \infty$ as $l \to \infty$. However, since $\{\Phi(\boldsymbol{Z}^k)\}_{k\in\mathbb{N}}$ is strictly decreasing and lower bounded by $\Phi_{\inf} > -\infty$, $\{\Phi(\boldsymbol{Z}^k)\}_{k\in\mathbb{N}}$ converges, and $\{\Phi(\boldsymbol{Z}^{k_l})\}_{l\in\mathbb{N}}$ also converges, which yields a contradiction. Thus, $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ is bounded, and there exists a convergent subsequence of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$.

(b) Let $\{\boldsymbol{Z}^{k_t}\}_{t\in\mathbb{N}}$ be a subsequence such that $\boldsymbol{Z}^{k_t} \to \boldsymbol{Z}^*$ as $t \to \infty$.

Since f is lower semicontinuous, we have

$$\liminf_{t\to\infty}\ f(\boldsymbol{x}^{k_t+1}) \geq f(\boldsymbol{x}^*).$$

From the update of $\boldsymbol{x}^{k_t+1}$ referring to (16), we obtain the following inequality:

$$\begin{aligned} f(\boldsymbol{x}^{k_t+1}) &+ \left\langle \boldsymbol{x}^{k_t+1} - \boldsymbol{x}^{k_t}, \boldsymbol{x}^{k_t} - \sum_{j=1}^{J} \mu_j \boldsymbol{W}^{-1} R_j^\top (\mathbf{Y}_j^{k_t+1}) \right\rangle_{\boldsymbol{W}} + \frac{\beta_{k_t}+1}{2}\|\boldsymbol{x}^{k_t+1} - \boldsymbol{x}^{k_t}\|_{\boldsymbol{W}}^2 \\ &\leq f(\boldsymbol{x}^*) + \left\langle \boldsymbol{x}^* - \boldsymbol{x}^{k_t}, \boldsymbol{x}^{k_t} - \sum_{j=1}^{J} \mu_j \boldsymbol{W}^{-1} R_j^\top (\mathbf{Y}_j^{k_t+1}) \right\rangle_{\boldsymbol{W}} + \frac{\beta_{k_t}+1}{2}\|\boldsymbol{x}^* - \boldsymbol{x}^{k_t}\|_{\boldsymbol{W}}^2. \end{aligned}$$

Letting $t \to \infty$ on both sides of the above inequality, we get

$$\begin{aligned} &\limsup_{t\to\infty}\ f(\boldsymbol{x}^{k_t+1}) \\ &\quad\leq f(\boldsymbol{x}^*) + \limsup_{t\to\infty} \left\{ \langle \boldsymbol{x}^* - \boldsymbol{x}^{k_t}, \boldsymbol{x}^{k_t} - \sum_{j=1}^{J} \mu_j \boldsymbol{W}^{-1} R_j^\top (\mathbf{Y}_j^{k_t+1}) \rangle_{\boldsymbol{W}} + \frac{\beta_{k_t}+1}{2}\|\boldsymbol{x}^* - \boldsymbol{x}^{k_t}\|_{\boldsymbol{W}}^2 \right\} \\ &\quad= f(\boldsymbol{x}^*), \end{aligned}$$

where we use the boundedness of the sequences $\{\boldsymbol{x}^{k_t+1}\}_{t\in\mathbb{N}}$, $\{\mathbf{Y}_j^{k_t+1}\}_{t\in\mathbb{N}}$, and $\{\beta_{k_t}\}_{t\in\mathbb{N}}$ and the result that $\lim_{t\to\infty} \|\boldsymbol{x}^{k_t+1} - \boldsymbol{x}^{k_t}\|_{\boldsymbol{W}} \to 0$ following from Corollary 4.4. Hence, $\lim_{t\to\infty} f(\boldsymbol{x}^{k_t+1}) = f(\boldsymbol{x}^*)$.

Due to the continuity of $\frac{\mu_j}{2}\|\boldsymbol{Y}_j - R_j(\boldsymbol{x})\|_F^2$ with respect to $\boldsymbol{Y}_j$ and $\boldsymbol{x}$ and the continuity of $g(t)$ with respect to t, we have

$$\lim_{t\to\infty} \Phi(\boldsymbol{Z}^{k_t+1}) = f(\boldsymbol{x}^*) + \sum_{j=1}^{J}\left(\frac{\mu_j}{2}\|\boldsymbol{Y}_j^* - R_j(\boldsymbol{x}^*)\|_F^2 + \lambda_j \sum_{i=1}^{m_j} g(\sigma_i(\boldsymbol{Y}_j^*))\right) = \Phi(\boldsymbol{Z}^*). \qquad ■$$

4.4. Convergence results. In this subsection, we show the convergence of the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ generated by the PARM algorithm.

Equipped with conditions (H1)–(H3), we can show that any limit point of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ is a critical point of Φ in the following theorem.

Theorem 4.12. *Suppose that the objective function Φ of model* (8) *satisfies assumptions* (A1)–(A3). *Let $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the PARM algorithm provided that assumption* (A4) *holds. Let $\mathcal{S}$ denote the set of all limit points of the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$, and let* $\operatorname{crit}(\Phi)$ *denote the set of all critical points of the function Φ. Then $\emptyset \neq \mathcal{S} \subseteq \operatorname{crit}(\Phi)$; that is, any limit point of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ is a critical point of Φ.*

Proof. Let $\boldsymbol{Z}^*$ be in $\mathcal{S} \neq \emptyset$, and let $\{\boldsymbol{Z}^{k_t}\}_{t\in\mathbb{N}}$ be a subsequence of $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ such that $\lim_{t\to\infty} \boldsymbol{Z}^{k_t} = \boldsymbol{Z}^*$. Then by Proposition 4.11, $\lim_{t\to\infty}\Phi(\boldsymbol{Z}^{k_t}) = \Phi(\boldsymbol{Z}^*)$. Also, it follows from Proposition 4.9 and Corollary 4.10 that $\boldsymbol{A}^{k_t} \in \partial\Phi(\boldsymbol{Z}^{k_t})$ and $\boldsymbol{A}^{k_t} \to \boldsymbol{0}$ as $t \to \infty$. Thus, by the definition of subdifferential in Definition 3.1, we have $\boldsymbol{0} \in \partial\Phi(\boldsymbol{Z}^*)$. ■

In addition to holding for conditions (H1)–(H3), if Φ is a KL function, then a stronger convergence result can be achieved for the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$. That is, we can prove that the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ itself converges to a critical point of Φ using the KL theory. For example, Φ in model (8) with f defined as in (5) and g as in (7) for multiplicative noise removal is a KL function, and then the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ converges as shown in the following theorem.

Theorem 4.13. *Suppose that the objective function Φ of model* (8) *satisfies assumptions* (A1)–(A3). *Let $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ be the sequence generated by the PARM algorithm provided that assumption* (A4) *holds. If Φ is a KL function, then the following assertions hold:*

(a) *The sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ has finite length, that is,*

$$\sum_{k=0}^{\infty}\|\boldsymbol{Z}^{k+1} - \boldsymbol{Z}^k\| < \infty;$$

(b) *the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ converges to a critical point of Φ.*

Proof. It follows from Propositions 4.3, 4.9, and 4.11 that the sequence $\{\boldsymbol{Z}^k\}_{k\in\mathbb{N}}$ satisfies conditions (H1)–(H3), respectively. Then the assertions (a) and (b) follow from Theorem 4.2. ■

5. Numerical results. In this section, we first describe a practical version of Algorithm 1 and then test the proposed algorithms to solve the proposed nonlocal low-rank model for multiplicative noise removal. We compare our proposed method with six existing methods: the DZ method [10], the HNW method [18], the I-DIV method [33], the TwL-mV method [20], the learned dictionary (Dict) method [17], and the SAR-BM3D method [28]. These methods are compared in terms of the visual quality and of the PSNR and SSIM values of the denoised images. Numerical results show superior performance of the proposed method over existing ones.

The experiments were implemented in MATLAB 2016b running a 64-bit Ubuntu 18.04 system and executed on an eight-core Intel Xeon E5-2640v3 128GB CPU at 2.6 GHz, with four NVIDIA Tesla P100 16GB GPUs. The proposed algorithms were accelerated using graphics processing units (GPUs), as the estimation of each patch matrix can be computed in parallel.

5.1. Practical version of PARM algorithm.

The PARM algorithm presented in Algorithm 1 converges theoretically, as shown in section 4, if the patch extraction operator R_j is assumed to be fixed. The extraction operator R_j plays an important role in improving the denoising performance because a better initialization of R_j can yield a better denoised image. In the case in which the optimal R_j is not available, it is empirically challenging to find an appropriate choice of R_j when only a noisy image given.

Algorithm 2. Practical version of the PARM algorithm.

```
Set parameters $\mu_j$, $\lambda_j$, and $\beta_k$
Initialize $\boldsymbol{x}^0$ and $\boldsymbol{w}_j^0$
Set $k = 0$
repeat
    Set extraction $\hat{R}_j^k$ by block matching
    Compute matrix $\hat{\boldsymbol{W}}^k$
    for $j$ from 1 to $J$ do
        $[\boldsymbol{U}_j^{k+1}, \widetilde{\boldsymbol{\Sigma}}_j^k, \boldsymbol{V}_j^{k+1}] = SVD\left(\hat{R}_j^k(\boldsymbol{x}^k)\right)$    ▷ SVD
        $\boldsymbol{\Sigma}_j^{k+1} = S_{\lambda_j/\mu_j, \boldsymbol{w}_j^k}(\widetilde{\boldsymbol{\Sigma}}_j^k)$    ▷ WSVT
        $\boldsymbol{Y}_j^{k+1} = \boldsymbol{U}_j^{k+1}\boldsymbol{\Sigma}_j^{k+1}(\boldsymbol{V}_j^{k+1})^\top$    ▷ Update $\boldsymbol{Y}_j^{k+1}$
        $(w_j^{k+1})_i = g'((\Sigma_j^{k+1})_{ii})$    ▷ Update $\boldsymbol{w}_j^{k+1}$
    end for
    $\boldsymbol{x}^{k+1} \in \mathrm{prox}^{\hat{\boldsymbol{W}}^k}_{\frac{1}{\beta_k+1}f}\left(\boldsymbol{x}^k + \frac{1}{\beta_k+1}\left(\sum_{j=1}^J \mu_j \hat{\boldsymbol{W}^k}^{-1} \hat{R_j^k}^\top (\boldsymbol{Y}_j^{k+1}) - \boldsymbol{x}^k\right)\right)$    ▷ Update $\boldsymbol{x}^{k+1}$
    $k \leftarrow k + 1$
until stopping criterion is satisfied
```

Here, we provide a practical version of the PARM algorithm with a dynamically updated patch extraction operator, denoted as $\hat{R}_j^k$. The operator $\hat{R}_j^k$ is recomputed at the kth step by block matching based on the estimated image $e^{\boldsymbol{x}^k}$, and the weighted counts matrix, now denoted as $\hat{\boldsymbol{W}}^k$, is recomputed based on the updated $\hat{R}_j^k$. As a result of this dynamically updating scheme, the extraction operator $\hat{R}_j^k$ becomes more efficient at collecting patches with similar textures, and this further helps improve the estimated image $\boldsymbol{x}^k$.

In this practical version of the PARM algorithm, we set $\boldsymbol{Y}_j^{k+1}$ as updated without using the previous update $\boldsymbol{Y}_j^k$ and its parameter α_{jk}. This is because $\boldsymbol{Y}_j^{k+1}$ is associated with $\hat{R}_j^k(\boldsymbol{x}^k)$, while $\boldsymbol{Y}_j^k$ is associated with $\hat{R}_j^{k-1}(\boldsymbol{x}^{k-1})$ using a different extraction operator. The patch matrices $\boldsymbol{Y}_j^{k+1}$ and $\boldsymbol{Y}_j^k$ may not refer to the same patch group.

The overall procedure of a practical version of the PARM algorithm for multiplicative noise removal is summarized as Algorithm 2.

5.2. Parameter settings.

First, we utilize block matching and normalization with mean zero to extract patch matrices using the following parameter settings for block matching. In Algorithm 1, the fixed extraction R_j is initialized via block matching based on the estimated image from the SAR-BM3D method; in Algorithm 2, the dynamically updated extraction $\hat{R}_j^k$ is computed at each step via block matching based on the update of $e^{\boldsymbol{x}^k}$. Additionally, both algorithms share the same parameter settings for block matching, including the search window, the patch size, and the number of patches in each patch group, as presented in Table 2.

Table 2
Settings for block matching.

L	Search window	Patch size	Patch number
1	50	10×10	150
3	50	9×9	120
5	50	8×8	100

Second, we manually set the model parameters and algorithm parameters for Algorithm 1 and Algorithm 2, respectively. The model parameters τ, λ_j, μ_j, ρ, γ, and ε depend on the noise level. The algorithm parameters (α_{jk} and) β_k influence the computational speed. The settings of the above parameters are presented in Tables 3 and 4.

Table 3
Parameter settings for Algorithm 1.

L	Standard images		Remote images		Common parameters					
	τ	λ_j	τ	λ_j	μ_j	ρ	γ	ε	α_{jk}	β_k
1	$(\beta_k+1)/50$	1.8	$(\beta_k+1)/100$	1	1	0.01	4	10^{-10}	0.001	0.001
3	$(\beta_k+1)/150$	1	$(\beta_k+1)/150$	0.45	1	1.5	1.9	10^{-10}	0.001	0.001
5	$(\beta_k+1)/250$	0.6	$(\beta_k+1)/200$	0.15	1	2	1.3	10^{-10}	0.001	0.001

Table 4
Parameter settings for Algorithm 2.

L	Standard images		Remote images		Common parameters				
	τ	λ_j	τ	λ_j	μ_j	ρ	γ	ε	β_k
1	$(\beta_k+1)/50$	1.8	$(\beta_k+1)/50$	1.7	1	0.01	4	10^{-10}	0.001
3	$(\beta_k+1)/150$	1	$(\beta_k+1)/150$	0.8	1	1.5	1.9	10^{-10}	0.001
5	$(\beta_k+1)/250$	0.6	$(\beta_k+1)/250$	0.5	1	2	1.3	10^{-10}	0.001

Third, the initialization settings and the stopping criteria are set differently for Algorithm 1 and Algorithm 2. For the case where an estimated clean image is available, Algorithm 1 is initialized using the estimated image from the SAR-BM3D method and is terminated if the relative error reaches a tolerance threshold ϵ_{tol}, e.g., 10^{-3}, or 50% of the initial relative error as follows:

$$\frac{\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^k\|_{\boldsymbol{W}}}{\|\boldsymbol{x}^k\|_{\boldsymbol{W}}} < \max\left\{\epsilon_{\mathrm{tol}}, \frac{\|\boldsymbol{x}^1-\boldsymbol{x}^0\|_{\boldsymbol{W}}}{\|\boldsymbol{x}^0\|_{\boldsymbol{W}}} \times 50\%\right\}.$$

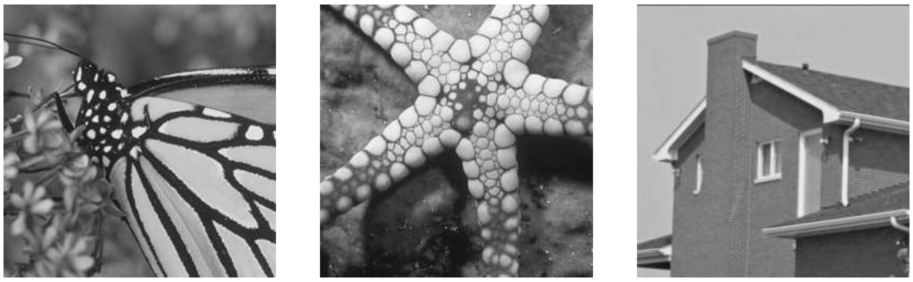
For the case where an estimated clean image is not available, Algorithm 2 is initialized using the given noisy image and terminated if the relative error reaches a tolerance threshold ϵ_{tol}, e.g., 10^{-3}, namely

$$\frac{\|\boldsymbol{x}^{k+1}-\boldsymbol{x}^{k}\|_{\boldsymbol{W}}}{\|\boldsymbol{x}^{k}\|_{\boldsymbol{W}}} < \epsilon_{\text{tol}}.$$

Finally, the restored image is estimated by $\hat{\boldsymbol{u}} = e^{\hat{\boldsymbol{x}}}$, where $\hat{\boldsymbol{x}}$ is the log-transformed image obtained from the proposed algorithms.

5.3. Numerical results tested on standard test images.

In this experiment, we use standard test images "Monarch," "Starfish," and "House," all of size 256 × 256, as shown in Figure 1. To generate the observed images, we degrade the original test images by multiplicative Gamma noise at $L = 1$, $L = 3$, and $L = 5$.



(a) Monarch (b) Starfish (c) House

Figure 1. *Standard test images.*

The evaluation of the image quality is measured in the intensity format between the original image $\boldsymbol{u} \in \mathbb{R}^N$ and the estimated image $\hat{\boldsymbol{u}} \in \mathbb{R}^N$ using the peak-signal-to-noise ratio (PSNR) defined as

$$\text{PSNR} = 10 \log_{10} \left(\frac{255^2 N}{\|\boldsymbol{u} - \hat{\boldsymbol{u}}\|_2^2} \right)$$

and the structural similarity index measure (SSIM) [40].

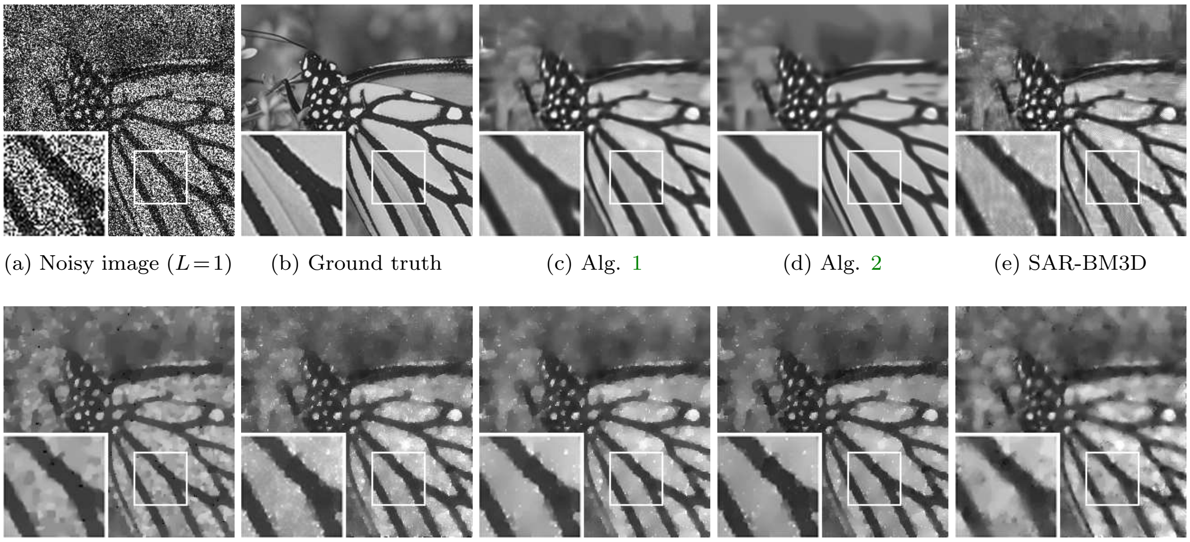
Table 5 reports the PSNR and SSIM values of the denoised images tested on three standard test images. The best results for each case are marked in bold, and the second-best results are underlined. Both Algorithm 1 and Algorithm 2 outperform all of the other methods in terms of PSNR and SSIM values. Compared with the benchmark SAR-BM3D method, Algorithm 1 achieves 0.27-0.59dB, 0.14-0.66dB, and 0.16-0.68dB improvements in PSNR when $L = 1$, $L = 3$, and $L = 5$, respectively. Algorithm 2 with updated patch extraction also surpasses the SAR-BM3D method and even surpasses Algorithm 1 in some of the cases, especially in terms of SSIM values.

Figures 2–4 present the denoised images tested on "Monarch" at noise level $L = 1$, "Starfish" at $L = 3$, and "House" at $L = 5$. In terms of the visual quality, Algorithms 1 and 2 perform better than other methods, because they reconstruct more details and more smooth textures but less noise and fewer artifacts. For example, compared to the DZ, HNW, I-DIV, TwL-mV, and learned dictionary methods, the proposed algorithms preserve more details of the rays of "Starfish" and generate more smooth textures on the wings of "Monarch"

Table 5

Numerical results tested on standard test images at different noise levels by different methods.

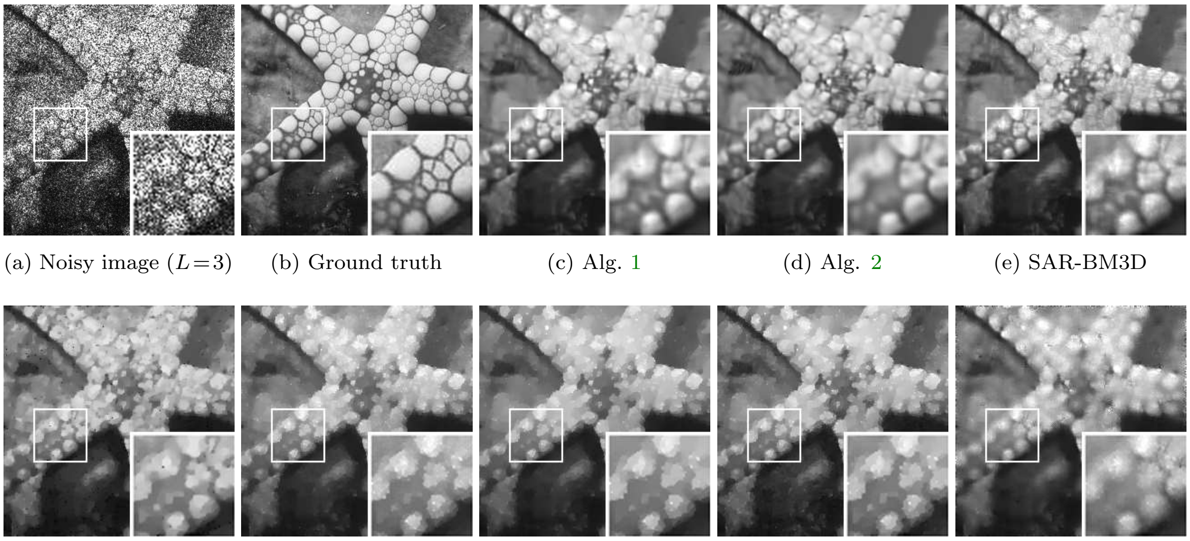
Image	L	Meas.	Alg. 1	Alg. 2	SAR-BM3D	DZ	HNW	I-DIV	TwL-4V	Dict.
Monarch	1	PSNR	**21.94**	21.64	21.36	19.38	19.73	19.91	19.26	19.50
		SSIM	0.6926	**0.7013**	0.6404	0.5758	0.5523	0.5883	0.5848	0.5726
	3	PSNR	**24.90**	24.58	24.48	22.66	22.55	22.69	22.43	23.02
		SSIM	0.8051	**0.8072**	0.7693	0.7156	0.7049	0.7244	0.7096	0.7449
	5	PSNR	**26.31**	26.24	25.78	24.04	23.88	23.98	23.74	24.38
		SSIM	0.8524	**0.8533**	0.8232	0.7648	0.7588	0.7723	0.7621	0.7740
Starfish	1	PSNR	**20.83**	20.69	20.56	19.35	20.36	20.01	19.89	20.18
		SSIM	0.5583	**0.5627**	0.5421	0.5051	0.5049	0.5028	0.4916	0.5192
	3	PSNR	**22.95**	22.86	22.81	21.88	22.01	21.67	21.85	21.77
		SSIM	**0.6748**	0.6700	0.6662	0.6095	0.6075	0.5922	0.5981	0.6051
	5	PSNR	**23.94**	23.92	23.78	22.87	22.80	22.56	22.67	22.57
		SSIM	**0.7176**	0.7162	0.7084	0.6623	0.6498	0.6415	0.6438	0.6499
House	1	PSNR	23.42	**24.10**	22.83	21.52	21.57	21.99	21.72	21.70
		SSIM	0.6726	**0.7222**	0.5916	0.6119	0.4925	0.5860	0.6017	0.5801
	3	PSNR	27.20	**27.27**	26.54	24.16	24.26	24.51	24.25	23.84
		SSIM	**0.7823**	0.7725	0.7139	0.6806	0.6365	0.6938	0.6597	0.6602
	5	PSNR	29.04	**29.13**	28.36	25.70	25.73	25.84	25.79	24.56
		SSIM	0.8115	**0.8155**	0.7641	0.7339	0.6995	0.7291	0.7197	0.6474



(a) Noisy image ($L = 1$) (b) Ground truth (c) Alg. 1 (d) Alg. 2 (e) SAR-BM3D

(f) DZ (g) HNW (h) I-DIV (i) TwL-4V (j) Dictionary

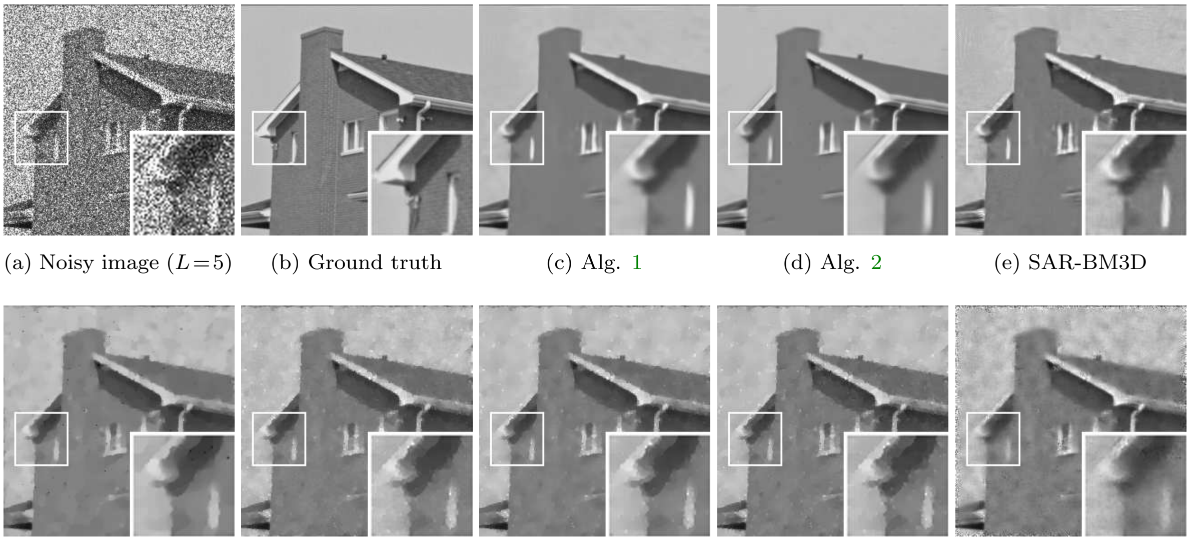
Figure 2. *Comparison of denoised images restored from “Monarch” at noise level* $L = 1$ *by different methods. The (PSNR, SSIM) values for each denoised image:* (c) *Alg* 1 *(*21.94*dB,* 0.6926*);* (d) *Alg* 2 *(*21.64*dB,* 0.7013*);* (e) *SAR-BM3D (*21.36*dB,* 0.6404*);* (f) *DZ (*19.38*dB,* 0.5758*);* (g) *HNW (*19.73*dB,* 0.5523*);* (h) *I-DIV (*19.91*dB,* 0.5883*);* (i) *TwL-4V (*19.26*dB,* 0.5848*);* (j) *dictionary (*19.50*dB,* 0.5726*).*



(a) Noisy image ($L=3$) (b) Ground truth (c) Alg. 1 (d) Alg. 2 (e) SAR-BM3D

(f) DZ (g) HNW (h) I-DIV (i) TwL-4V (j) Dictionary

Figure 3. *Comparison of denoised images restored from "Starfish" at noise level $L = 3$ by different methods. The (PSNR, SSIM) values for each denoised image:* (c) *Alg.* 1 (22.95*dB,* 0.6748); (d) *Alg.* 2 (22.86*dB,* 0.6700); (e) *SAR-BM3D* (22.81*dB,* 0.6662); (f) *DZ* (21.88*dB,* 0.6095); (g) *HNW* (22.01*dB,* 0.6075); (h) *I-DIV* (21.67*dB,* 0.5922); (i) *TwL-4V* (21.85*dB,* 0.5981); (j) *dictionary* (21.77*dB,* 0.6051).



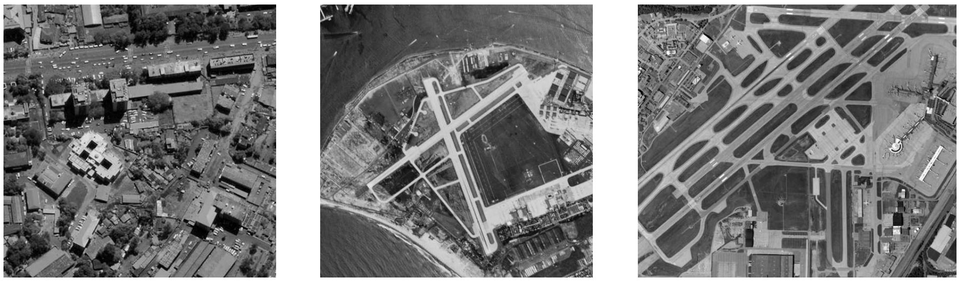
(a) Noisy image ($L=5$) (b) Ground truth (c) Alg. 1 (d) Alg. 2 (e) SAR-BM3D

(f) DZ (g) HNW (h) I-DIV (i) TwL-4V (j) Dictionary

Figure 4. *Comparison of denoised images restored from "House" at noise level $L = 5$ by different methods. The (PSNR, SSIM) values for each denoised image:* (c) *Alg.* 1 *(*29.04*dB,* 0.8115*);* (d) *Alg.* 2 *(*29.13*dB,* 0.8155*);* (e) *SAR-BM3D (*28.36*dB,* 0.7641*);* (f) *DZ (*25.70*dB,* 0.7339*);* (g) *HNW (*25.73*dB,* 0.6995*);* (h) *I-DIV (*25.84*dB,* 0.7291*);* (i) *TwL-4V (*25.79*dB,* 0.7197*);* (j) *dictionary (*24.56*dB,* 0.6474*).*

and the sky of "House." Compared to the benchmark SAR-BM3D method, the proposed algorithms generate fewer artifacts, resulting in better images in terms of PSNR and SSIM values.

5.4. Numerical results tested on remote sensing images. In this experiment, we use remote sensing images "Remote 1" and "Remote 2," both of size 512 × 512, and "Remote 3" of size 540 × 632 as shown in Figure 5. To generate the observed images, we degrade the original test images by multiplicative Gamma noise at $L = 1$, $L = 3$, and $L = 5$. The image quality is evaluated using PSNR and SSIM values.



(a) Remote 1 (b) Remote 2 (c) Remote 3

Figure 5. *Remote sensing images.*

Table 6

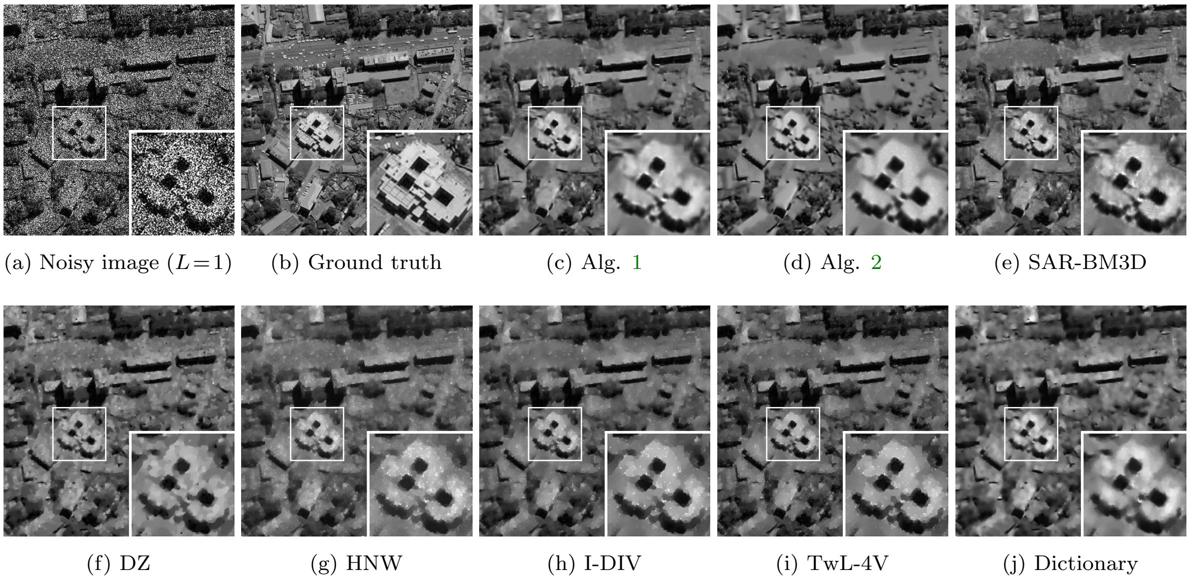
Numerical results tested on remote sensing images at different noise levels by different methods.

Image	L	Meas.	Alg. 1	Alg. 2	SAR-BM3D	DZ	HNW	I-DIV	TwL-4V	Dict.
Remote 1	1	PSNR	**21.23**	21.15	21.12	20.47	20.24	20.03	20.07	20.44
		SSIM	0.5459	**0.5474**	0.5393	0.4950	0.4551	0.4709	0.4934	0.4867
	3	PSNR	**23.45**	23.38	23.39	22.51	21.96	22.05	22.38	20.52
		SSIM	0.6730	**0.6736**	0.6716	0.6199	0.5686	0.5935	0.6268	0.4953
	5	PSNR	24.55	**24.62**	24.49	23.69	22.90	23.17	23.55	20.93
		SSIM	0.7283	**0.7336**	0.7261	0.6800	0.6274	0.6595	0.6824	0.5350
Remote 2	1	PSNR	**21.91**	21.90	21.68	20.37	20.89	20.58	20.51	20.40
		SSIM	0.5461	**0.5486**	0.5334	0.4827	0.4783	0.4791	0.4789	0.4665
	3	PSNR	**24.13**	24.05	24.03	22.76	22.71	22.49	22.66	22.34
		SSIM	0.6471	**0.6504**	0.6449	0.5758	0.5805	0.5744	0.5845	0.5592
	5	PSNR	25.32	**25.36**	25.21	23.98	23.79	23.59	23.81	23.67
		SSIM	0.6964	**0.6988**	0.6939	0.6302	0.6294	0.6265	0.6364	0.6179
Remote 3	1	PSNR	22.16	**22.22**	21.88	20.93	20.89	20.81	20.79	20.59
		SSIM	0.5895	**0.5970**	0.5565	0.5292	0.4916	0.5131	0.5182	0.4955
	3	PSNR	**24.66**	24.51	24.47	23.34	22.81	23.02	23.18	22.14
		SSIM	**0.7002**	0.6989	0.6811	0.6236	0.6077	0.6250	0.6218	0.5545
	5	PSNR	25.80	**25.82**	25.65	24.45	23.79	24.10	24.30	22.70
		SSIM	0.7427	**0.7464**	0.7316	0.6745	0.6562	0.6713	0.6737	0.5736

Table 6 reports the PSNR and SSIM values of the denoised images tested on three remote sensing images. Algorithms 1 and 2 achieve great performance in PSNR and SSIM values over other methods. For example, Algorithm 1 outperforms the benchmark SAR-BM3D method by 0.11-0.28dB, 0.06-0.19dB, and 0.06-0.15dB in PSNR when $L = 1$, $L = 3$, and $L = 5$, respectively, and it outperforms the other traditional methods by 0.76-1.57dB, 0.94-2.93dB, and 0.86-3.62dB in PSNR when $L = 1$, $L = 3$, and $L = 5$, respectively. Algorithm 2 is also comparable to Algorithm 1 and the SAR-BM3D method.

Figures 6–8 present the denoised images by different methods tested on "Remote 1" at noise level $L = 1$, "Remote 2" at $L = 3$, and "Remote 3" at $L = 5$. Algorithm 1, Algorithm 2, and the benchmark SAR-BM3D method achieve significantly better visual quality over other methods. For example, they reconstruct buildings, roads, and patterns with fine edges and textures.

5.5. Numerical results tested on real SAR images. In this experiment, we use real SAR images "SAR 1" of size 370 × 370 and "SAR 2" of size 350 × 350 as shown in Figures 9(a) and 10(a), respectively.


(a) Noisy image ($L\!=\!1$) (b) Ground truth (c) Alg. 1 (d) Alg. 2 (e) SAR-BM3D

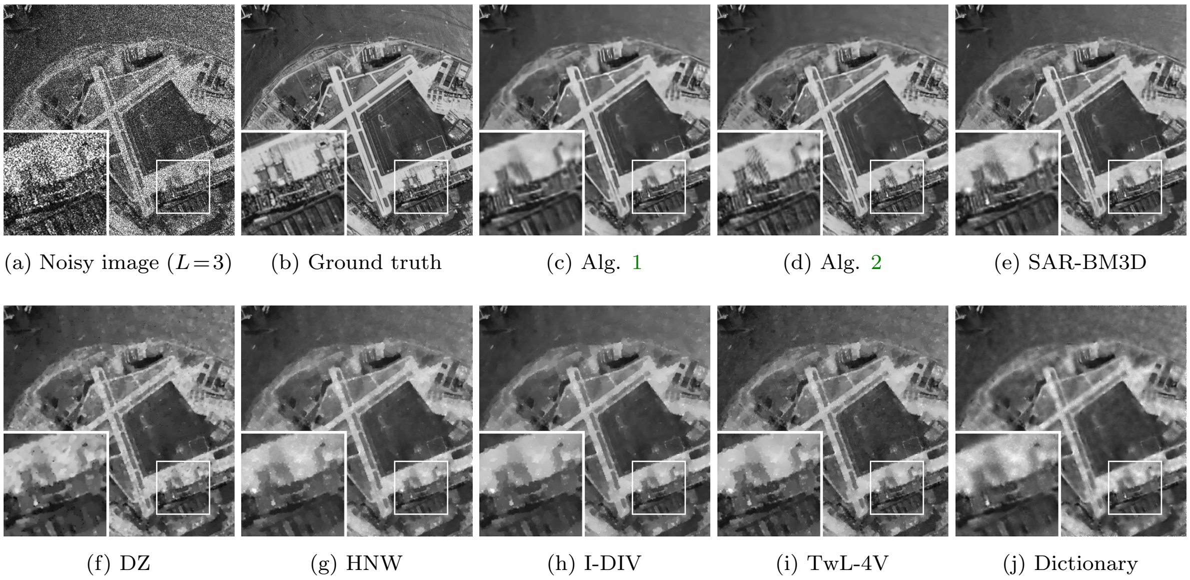
(f) DZ (g) HNW (h) I-DIV (i) TwL-4V (j) Dictionary

Figure 6. *Comparison of denoised images restored from "Remote 1" at noise level $L = 1$ by different methods. The (PSNR, SSIM) values for each denoised image:* (c) *Alg. 1 (21.23dB, 0.5459);* (d) *Alg. 2 (21.15dB, 0.5474);* (e) *SAR-BM3D (21.12dB, 0.5393);* (f) *DZ (20.47dB, 0.4950);* (g) *HNW (20.24dB, 0.4551);* (h) *I-DIV (20.03dB, 0.4709);* (i) *TwL-4V (20.07dB, 0.4934);* (j) *dictionary (20.44dB, 0.4867).*

Figures 9 and 10 demonstrate that Algorithms 1 and 2 achieve better denoising performance than other methods. For example, they reconstruct more local structures and smooth textures than the DZ, HNW, I-DIV, TwL-4V, and learned dictionary methods, and they remove more noise and generate fewer artifacts than the benchmark SAR-BM3D method.

In addition to the visual quality comparison on the denoised images, we can also receive guidance by computing the equivalent number of looks (ENL) and analyzing the ratio images for different methods.

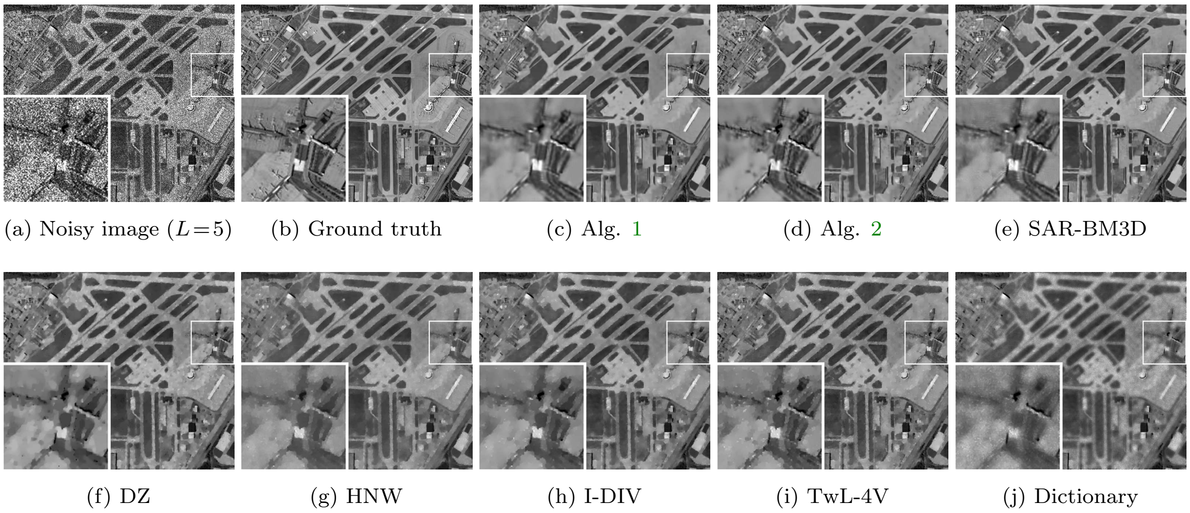
The ENL of an estimated image $\hat{\boldsymbol{u}} \in \mathbb{R}^N$ measures the multiplicative noise reduction in



(a) Noisy image ($L=3$) (b) Ground truth (c) Alg. 1 (d) Alg. 2 (e) SAR-BM3D

(f) DZ (g) HNW (h) I-DIV (i) TwL-4V (j) Dictionary

Figure 7. *Comparison of denoised images restored from "Remote 2" at noise level $L = 3$ by different methods. The (PSNR, SSIM) values for each denoised image:* (c) *Alg.* 1 *(*24.13*dB,* 0.6471*);* (d) *Alg.* 2 *(*24.05*dB,* 0.6504*);* (e) *SAR-BM3D (*24.03*dB,* 0.6449*);* (f) *DZ (*22.76*dB,* 0.5758*);* (g) *HNW (*22.71*dB,* 0.5805*);* (h) *I-DIV (*22.49*dB,* 0.5744*);* (i) *TwL-4V (*22.66*dB,* 0.5845*);* (j) *dictionary (*22.34*dB,* 0.5592*).*



(a) Noisy image ($L=5$) (b) Ground truth (c) Alg. 1 (d) Alg. 2 (e) SAR-BM3D

(f) DZ (g) HNW (h) I-DIV (i) TwL-4V (j) Dictionary

Figure 8. *Comparison of denoised images restored from "Remote 3" at noise level $L = 5$ by different methods. The (PSNR, SSIM) values for each denoised image:* (c) *Alg.* 1 *(*25.80*dB,* 0.7427*);* (d) *Alg.* 2 *(*25.82*dB,* 0.7464*);* (e) *SAR-BM3D (*25.65*dB,* 0.7316*);* (f) *DZ (*24.45*dB,* 0.6745*);* (g) *HNW (*23.79*dB,* 0.6562*);* (h) *I-DIV (*24.10*dB,* 0.6713*);* (i) *TwL-4V (*24.30*dB,* 0.6737*);* (j) *dictionary (*22.70*dB,* 0.5736*).*

homogeneous regions and is defined as

$$\mathrm{ENL} = \frac{\mu_{\hat{\boldsymbol{u}}}^2}{\sigma_{\hat{\boldsymbol{u}}}^2},$$

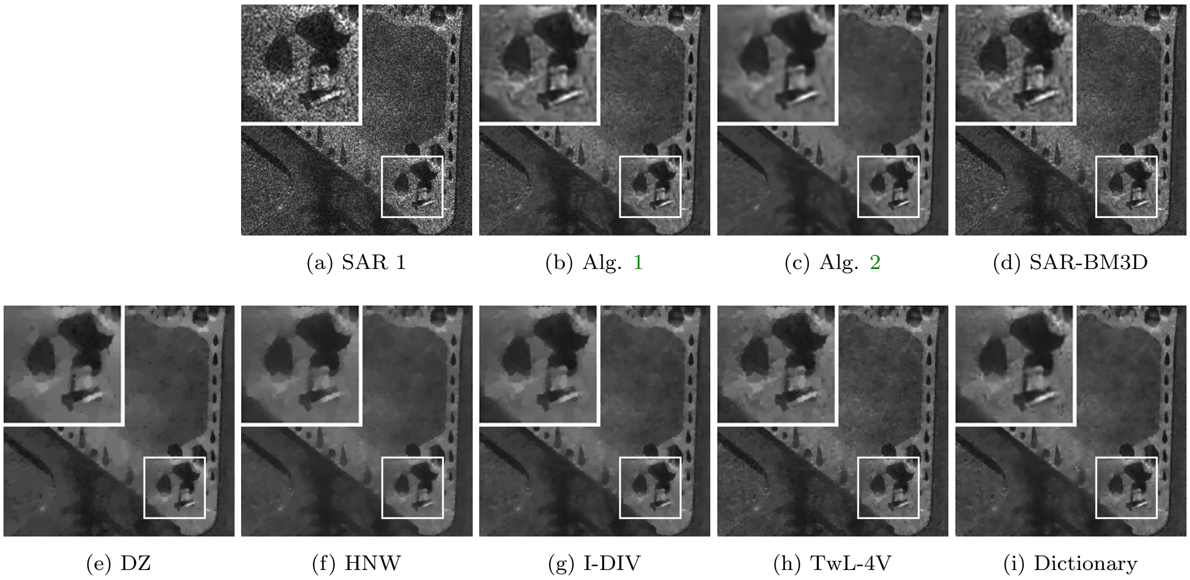

(a) SAR 1 (b) Alg. 1 (c) Alg. 2 (d) SAR-BM3D
(e) DZ (f) HNW (g) I-DIV (h) TwL-4V (i) Dictionary

Figure 9. *Comparison of denoised images restored from "SAR 1" by different methods.*

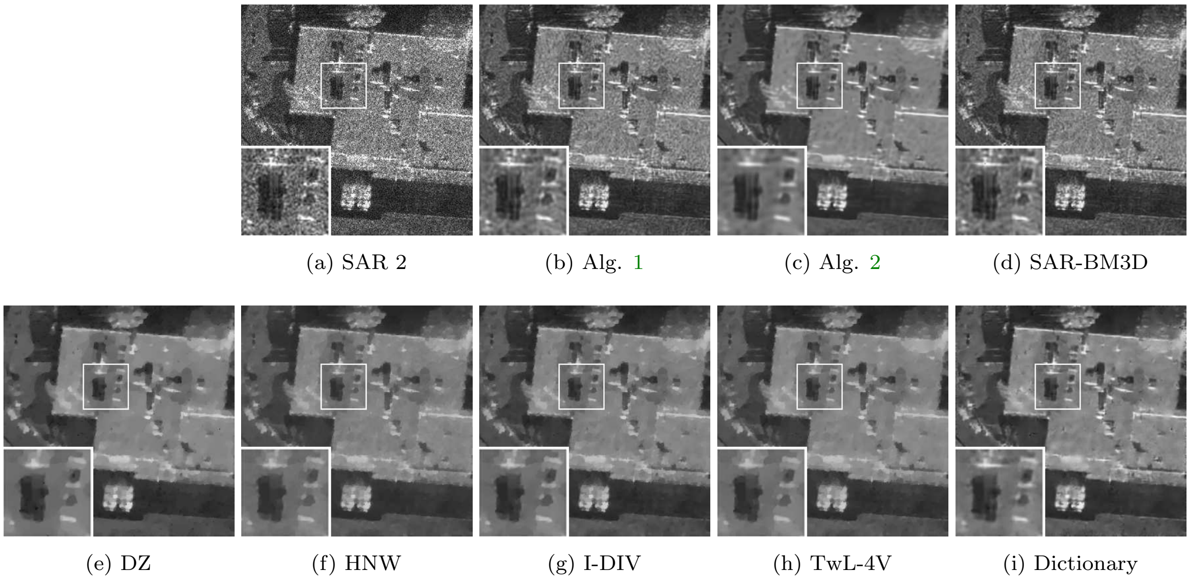

(a) SAR 2 (b) Alg. 1 (c) Alg. 2 (d) SAR-BM3D
(e) DZ (f) HNW (g) I-DIV (h) TwL-4V (i) Dictionary

Figure 10. *Comparison of denoised images restored from "SAR 2" by different methods.*

where $\mu_{\hat{\boldsymbol{u}}}$ is the average intensity of the selected area and $\sigma^2_{\hat{\boldsymbol{u}}}$ is its variance.

For computing the ENL values, two homogeneous regions are, respectively, selected from "SAR 1" and "SAR 2" as indicated by the white boxes in Figures 11(a) and 12(a). Table 7 presents the ENL values for different methods. The SAR-BM3D method has the lowest ENL values compared to other methods, which indicates that either the multiplicative noise is not effectively reduced or there exist some artifacts in the estimated image. The other methods

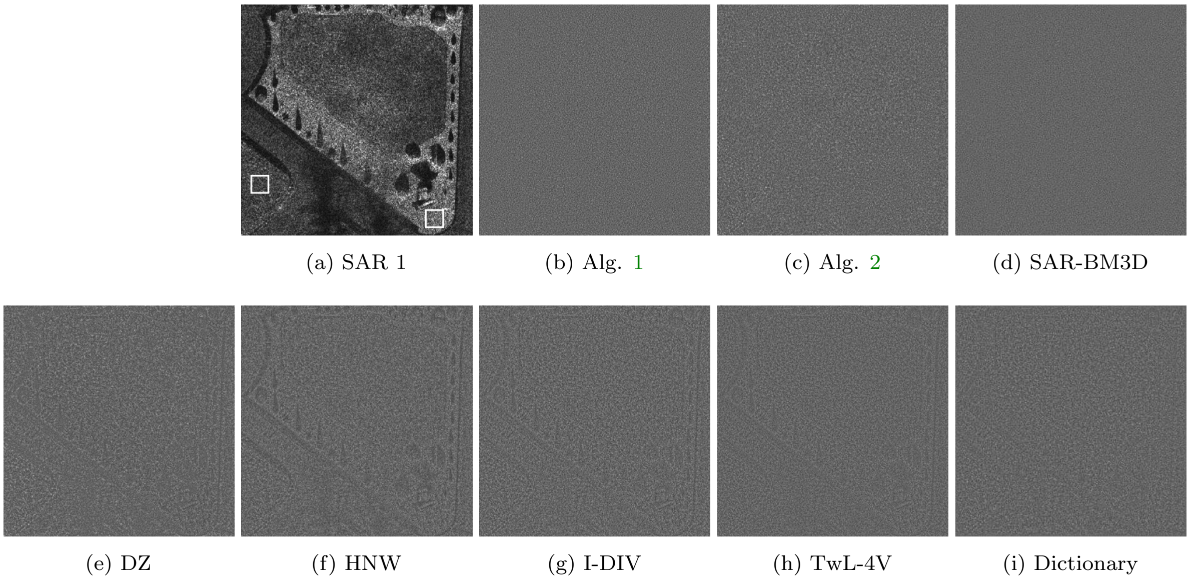

(a) SAR 1 (b) Alg. 1 (c) Alg. 2 (d) SAR-BM3D
(e) DZ (f) HNW (g) I-DIV (h) TwL-4V (i) Dictionary

Figure 11. *Comparison of the ratio images between "SAR 1" and the estimated images by different methods.*

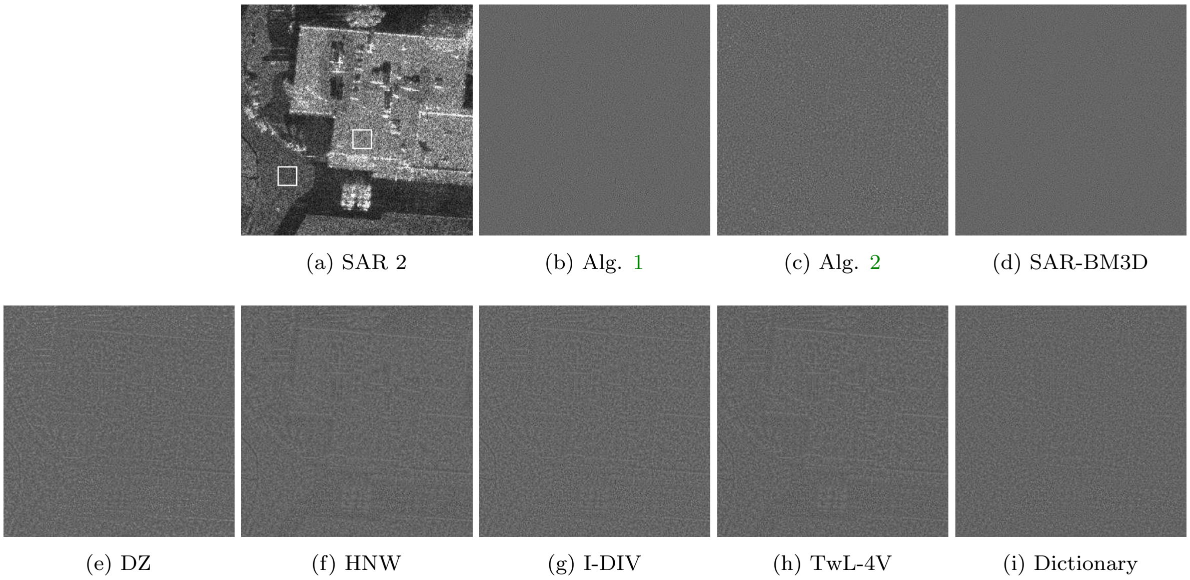

(a) SAR 2 (b) Alg. 1 (c) Alg. 2 (d) SAR-BM3D
(e) DZ (f) HNW (g) I-DIV (h) TwL-4V (i) Dictionary

Figure 12. *Comparison of the ratio images between "SAR 2" and the estimated images by different methods.*

have relatively large ENL values, which indicates that either the multiplicative noise is well removed or the estimated image is oversmooth.

The pointwise ratio between the real SAR image $\boldsymbol{u} \in \mathbb{R}^N$ and the estimated image $\hat{\boldsymbol{u}} \in \mathbb{R}^N$ simulates the multiplicative noise that has been removed by the given method and is defined as

$$\text{Ratio} = \frac{\boldsymbol{u}}{\hat{\boldsymbol{u}}}.$$

Table 7
ENL values of denoised images restored from real SAR images by different methods.

Image	Region	Noisy	Alg 1	Alg 2	SAR-BM3D	DZ	HNW	I-DIV	TwL-4V	Dict
SAR 1	Left	9.46	63.09	332.55	42.84	745.14	521.24	306.90	117.48	183.57
	Right	10.65	82.59	230.13	57.36	333.18	360.12	276.04	144.54	175.49
SAR 2	Left	22.64	97.02	393.72	91.92	1008.50	816.97	501.22	579.26	336.58
	Right	21.91	96.58	395.04	91.03	985.64	740.27	566.30	724.76	444.01

The ratio images for different methods are presented in Figures 11 and 12. The ratio images for Algorithm 1, Algorithm, 2 and the SAR-BM3D method present almost random speckle, which matches the expected statistics. On the contrary, the ratio images for the other methods still contain some geometric structures, such as edges and details correlated to the real SAR images, which indicates that those methods have removed some valuable information in addition to noise.

6. Conclusions.

We have proposed an effective method for multiplicative noise removal. The proposed method consists of a nonlocal low-rank model, which exploits the low-rank prior of nonlocal similar patch matrices, and the PARM iterative algorithm, which solves the nonconvex nonsmooth optimization problem resulting from the proposed model. We have established the global convergence of the sequence generated by the PARM algorithm to a critical point of the nonconvex nonsmooth objective function of the resulting optimization problem. Numerical results have demonstrated that the proposed method with a theoretical convergence guarantee outperforms several existing methods, including the state-of-the-art SAR-BM3D method.

REFERENCES

[1] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, *Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Lojasiewicz inequality*, Math. Oper. Res., 35 (2010), pp. 438–457.

[2] H. Attouch, J. Bolte, and F. B. Svaiter, *Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods*, Math. Program., 137 (2013), pp. 91–129.

[3] G. Aubert and J.-F. Aujol, *A variational approach to removing multiplicative noise*, SIAM J. Appl. Math., 68 (2008), pp. 925–946, https://doi.org/10.1137/060671814.

[4] J. Bolte, S. Sabach, and M. Teboulle, *Proximal alternating linearized minimization for nonconvex and nonsmooth problems*, Math. Program., 146 (2014), pp. 459–494.

[5] A. Buades, B. Coll, and J.-M. Morel, *A non-local algorithm for image denoising*, in Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, Vol. 2, 2005, pp. 60–65.

[6] R. Chan, H. Yang, and T. Zeng, *A two-stage image segmentation method for blurry images with Poisson or multiplicative gamma noise*, SIAM J. Imaging Sci., 7 (2014), pp. 98–127, https://doi.org/10.1137/130920241.

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

[8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse 3-D transform-*

domain collaborative filtering, IEEE Trans. Image Process., 16 (2007), pp. 2080–2095.

[9] W. Dong, G. Shi, X. Li, Y. Ma, and F. Huang, *Compressive sensing via nonlocal low-rank regularization*, IEEE Trans. Image Process., 23 (2014), pp. 3618–3632.

[10] Y. Dong and T. Zeng, *A convex variational model for restoring blurred images with multiplicative noise*, SIAM J. Imaging Sci., 6 (2013), pp. 1598–1625, https://doi.org/10.1137/120870621.

[11] W. Feng, H. Lei, and Y. Gao, *Speckle reduction via higher order total variation approach*, IEEE Trans. Image Process., 23 (2014), pp. 1831–1843.

[12] M. A. T. Figueiredo and J. M. Bioucas-Dias, *Restoration of Poissonian images using alternating direction optimization*, IEEE Trans. Image Process., 19 (2010), pp. 3133–3145.

[13] J. W. Goodman, *Some fundamental properties of speckle*, J. Opt. Soc. Amer., 66 (1976), pp. 1145–1150.

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

[15] R. A. Horn and C. R. Johnson, *Topics in Matrix Analysis*, Cambridge University Press, Cambridge, UK, 1994.

[16] 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 Trans. Image Process., 26 (2017), pp. 3171–3186.

[17] Y.-M. Huang, L. Moisan, M. K. Ng, and T. Zeng, *Multiplicative noise removal via a learned dictionary*, IEEE Trans. Image Process., 21 (2012), pp. 4534–4543.

[18] Y.-M. Huang, M. K. Ng, and Y.-W. Wen, *A new total variation method for multiplicative noise removal*, SIAM J. Imaging Sci., 2 (2009), pp. 20–40, https://doi.org/10.1137/080712593.

[19] Y.-M. Huang, H.-Y. Yan, Y.-W. Wen, and X. Yang, *Rank minimization with applications to image noise removal*, Inform. Sci., 429 (2018), pp. 147–163.

[20] M. Kang, S. Yun, and H. Woo, *Two-level convex relaxed variational model for multiplicative denoising*, SIAM J. Imaging Sci., 6 (2013), pp. 875–903, https://doi.org/10.1137/11086077X.

[21] A. S. Lewis and H. S. Sendov, *Nonsmooth analysis of singular values, part* I: *Theory*, Set-Valued Anal., 13 (2005), pp. 213–241.

[22] A. S. Lewis and H. S. Sendov, *Nonsmooth analysis of singular values, part* II: *Applications*, Set-Valued Anal., 13 (2005), pp. 243–264.

[23] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi, *Image recovery via nonlocal operators*, J. Sci. Comput., 42 (2010), pp. 185–197.

[24] C. Lu, J. Tang, S. Yan, and Z. Lin, *Generalized nonconvex nonsmooth low-rank minimization*, in Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 4130–4137.

[25] J. Lu, L. Shen, C. Xu, and Y. Xu, *Multiplicative noise removal in imaging: An exp-model and its fixed-point proximity algorithm*, Appl. Comput. Harmon. Anal., 41 (2016), pp. 518–539.

[26] X. Nie, X. Huang, and W. Feng, *A new nonlocal TV-based variational model for SAR image despeckling based on the G^0 distribution*, Digital Signal Process., 68 (2017), pp. 44–56.

[27] C. J. Oliver and S. Quegan, *Understanding Synthetic Aperture Radar Images*, SciTech Publishing, Inc., Raleigh, NC, 2004.

[28] S. Parrilli, M. Poderico, C. V. Angelino, and L. Verdoliva, *A nonlocal SAR image denoising algorithm based on LLMMSE wavelet shrinkage*, IEEE Trans. Geosci. Remote Sensing, 50 (2011), pp. 606–616.

[29] B. Recht, M. Fazel, and P. A. Parrilo, *Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization*, SIAM Rev., 52 (2010), pp. 471–501, https://doi.org/10.1137/070697835.

[30] J. M. Schmitt, S. Xiang, and K. M. Yung, *Speckle in optical coherence tomography*, J. Biomed. Opt., 4 (1999), pp. 95–105.

[31] S. Setzer, G. Steidl, and T. Teuber, *Deblurring Poissonian images by split Bregman techniques*, J. Vis. Comm. Image Represent., 21 (2010), pp. 193–199.

[32] J. Shi and S. Osher, *A nonlinear inverse scale space method for a convex multiplicative noise model*, SIAM J. Imaging Sci., 1 (2008), pp. 294–321, https://doi.org/10.1137/070689954.

[33] G. Steidl and T. Teuber, *Removing multiplicative noise by Douglas-Rachford splitting methods*, J.

Math. Imaging Vis., 36 (2010), pp. 168–184.

[34] T. Sun, H. Jiang, and L. Cheng, *Convergence of proximal iteratively reweighted nuclear norm algorithm for image processing*, IEEE Trans. Image Process., 26 (2017), pp. 5632–5644.

[35] R. F. Wagner, S. W. Smith, J. M. Sandrik, and H. Lopez, *Statistics of speckle in ultrasound B-scans*, IEEE Trans. Sonics Ultrasonics, 30 (1983), pp. 156–163.

[36] J. Wei, Y. Huang, L. Ke, and L. Wang, *Nonlocal low-rank-based compressed sensing for remote sensing image reconstruction*, IEEE Geosci. Remote Sensing Lett., 13 (2017), pp. 1557–1561.

[37] Y. Wu, *Speckle noise removal via nonlocal low-rank regularization*, J. Vis. Comm. Image Represent., 39 (2016), pp. 172–180.

[38] W. Yu, W. Yin, and J. Zeng, *Global convergence of ADMM in nonconvex nonsmooth optimization*, J. Sci. Comput., 78 (2018), pp. 1–35.

[39] S. Yun and H. Woo, *A new multiplicative denoising variational model based on mth root transformation*, IEEE Trans. Image Process., 21 (2012), pp. 2523–2533.

[40] W. Zhou, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image quality assessment: From error visibility to structural similarity*, IEEE Trans. Image Process., 13 (2004), pp. 600–612.