AIMS Mathematics, 9(8): 21094–21124.
DOI: 10.3934/math.20241025
Received: 29 March 2024
Revised: 29 May 2024
Accepted: 07 June 2024
Published: 01 July 2024

http://www.aimspress.com/journal/Math

Research article

Nonconvex fractional order total variation based image denoising model under mixed stripe and Gaussian noise

Myeongmin Kang[1] **and Miyoun Jung**[2,*]

[1] Department of Mathematics, Chungnam National University, Daejeon 34134, Korea
[2] Department of Mathematics, Hankuk University of Foreign Studies, Yongin 17035, Korea

* **Correspondence:** Email: mjung@hufs.ac.kr; Tel: +820313304261.

Abstract: In this paper, we propose a minimization-based image denoising model for the removal of mixed stripe and Gaussian noise. The objective function includes the prior information from both the stripe noise and image. Specifically, we adopted a unidirectional regularization term and a nonconvex group sparsity term for the stripe noise component, while we utilized a nonconvex fractional order total variation (FTV) regularization for the image component. The priors for stripes enable adequate extraction of periodic or non-periodic stripes from an image in the presence of high levels of Gaussian noise. Moreover, the nonconvex FTV facilitates image restoration with less staircase artifacts and well-preserved edges and textures. To solve the nonconvex problem, we employed an iteratively reweighted ℓ_1 algorithm, and then the alternating direction method of multipliers was adopted for solving subproblems. This led to an efficient iterative algorithm, and its global convergence was proven. Numerical results show that the proposed model provides better denoising performance than existing models with respect to visual features and image quality evaluations.

Keywords: image denoising; stripe noise; Gaussian noise; fractional order total variation; nonconvex regularization
Mathematics Subject Classification: 68U10, 65K10, 94A08, 49J40

1. Introduction

Remote sensing images have been extensively used in various applications such as urban planning, military operations, and environment monitoring [1,2]. However, during the image acquisition process, remote sensing images are inevitably polluted by stripe noise, mainly due to the difference in the responses of the detectors and the calibration error. On the other hand, Gaussian noise is typically caused by the temperature of the sensor and the level of illumination in the environment that corrupts every pixel. The stripe noise is usually mixed with random Gaussian noise. This mixture of noise not

only degrades the image quality but also hampers the subsequent processing such as classification [3], object segmentation [4], target detection [5], and image unmixing [6]. Therefore, removing this mixed noise is an essential preprocessing step for remote sensing images. In this work, we focus on restoring a remote sensing image in the presence of mixed stripes and Gaussian noise.

The removal of stripe noise in an image can be categorized as filtering-based methods, statistics-based methods, and optimization-based methods. Filtering-based methods eliminate stripe noise by truncating the stripe component in the transformed domain such as the Fourier transform [7], wavelet domain [8], and the combined domain [9]. These algorithms are simple but they assume that the stripe noise is periodic and identifiable in the power spectrum. Statistics-based methods such as moment matching [10] and histogram matching [11] assume that the statistical features of each sensor are the same. These methods have a low computational cost, but their performance is greatly affected by a predetermined reference moment or histogram.

Optimization-based models consider an ill-posed inverse problem for the stripe noise removal and utilize prior knowledge of the ideal image or the stripe noise as regularization [12–34]. For example, Bouali and Ladjal [13] proposed a unidirectional total variation (UTV) model for MODIS image destriping, by exploiting the direction feature of stripes. Several studies have developed the UTV model using various regularizations [14–17]. Despite the satisfactory performance of these UTV-based models, they directly restore the image without considering the characteristics of stripes, which generates a loss of image details along with the stripes. To overcome this shortcoming, various works make use of the structural property of stripes. In particular, in [18, 24], the ℓ_0-norm was used to constrain the global sparse distribution of stripes, but this assumption does not apply when the stripes are very dense. Chang et al. [19] proposed an image decomposition model by employing a low-rankness prior for stripes and TV regularization [35] for the image. They adopted the nuclear norm to express the global redundancy of stripes. Meanwhile, Yang et al. [27] exploited the Schatten 1/2-norm to characterize the low-rankness of stripes and unidirectional high-order TV for the image. In [20, 22, 25, 34], the $\ell_{2,1}$-norm of stripes was suggested to promote the group sparsity of stripes. These destriping methods perform well when the stripes satisfy the low-rankness or group sparsity assumptions. However, these assumptions may be violated when the stripes are complex. To deal with complex stripes such as irregular or partial stripes, Wang et al. [26] suggested a reweighted $\ell_{2,1}$-norm regularization for stripes. However, this model only consider the features of stripes without prior image information, hence it cannot capture a clean image in the presence of Gaussian noise. In this work, we introduce a destriping model that enables restoring a clean image by simultaneously suppressing stripes and Gaussian noise, by utilizing the group sparsity characteristic and directional feature of stripes.

TV regularization has been widely used in various image restoration problems because of its edge-preserving advantage. However, TV tends to generate staircase artifacts in reconstructed images as it pursues piecewise-constant solutions. To mitigate the staircase effect, higher-order TV has been suggested; for example, second-order TV [36, 37], total generalized variation [38], and hybrid TV [39, 40]. Different from this type of high-order TV, fractional-order TV (FTV) uses derivatives with order greater than or equal to one, bringing a compact discrete form and thus yielding computational advantage. FTV regularization takes neighboring pixel values into account, so it preserves local geometric characteristics and thereby textures. Therefore, it has been adopted for various image processing problems, such as image denoising [41–48], texture enhancement [49], and super-resolution [50]. FTV has been empirically proven to suppress staircasing artifacts and improve

the effectiveness of texture preservation. On the other hand, nonconvex regularization has attracted attention because nonconvex regularizers have advantages over convex regularizers in maintaining edges and details [51–53]. In various works [54–57], nonconvex higher-order TV has been developed, which contributes to edges conservation and the reduction of the staircase effect. Also, many efficient algorithms have been developed for solving nonconvex minimization problems. In particular, iteratively convex majorization-minimization methods for solving nonsmooth nonconvex minimization problems with convergence analysis have been proposed in [58], which are generalizations of the iteratively reweighted ℓ_1 algorithm (IRL1) proposed for compressive sensing [59]. In the present work, we apply a nonconvex FTV regularziation to the image component to benefit from both FTV and nonconvex regularization. Besides, we employ IRL1 to solve the proposed model, along with a convergence analysis.

In this article, we introduce a novel image denoising model in the presence of a mixture of stripe and Gaussian noises. In this work, we consider a relatively high level of Gaussian noise, unlike previous works that consider a low level of Gaussian noise. To effectively remove the mixed noise and recover a clean image, the proposed model exploits prior knowledge of both stripe noise and image components. In particular, the group sparsity feature and directional property of stripes are utilized to extract stripe noise. Besides, a nonconvex FTV is used for the image component to recover its smooth regions with less staircase artifacts while preserving edges. To solve the proposed nonconvex model, we employ the IRL1 algorithm and alternating direction method of multipliers [60–62] and provide a convergence analysis.

The remainder of this article is organized as follows. In Section 2, we recall several optimization-based destriping models and review the FTV. Section 3 introduces the proposed model for the removal of stripes and Gaussian noise. An optimization algorithm for solving the proposed model is also provided, and its convergence is proven. Section 4 presents the experimental results of the proposed model, comparing it with several existing models. Finally, in Section 5, we summarize our work and provide some remarks.

2. Background

2.1. Optimization-based destriping models

This section reviews several optimization-based models for stripe noise removal. In remote sensing images, stripe noise typically includes additive and multiplicative noise components [12]. The multiplicative noise can be described as additive noise by a logarithmic operation [63], thus stripe noise can be regarded as additive noise. Therefore, the degradation model for the removal of stripe noise is usually given by

$$\mathbf{f} = \mathbf{u} + \mathbf{s} + \mathbf{n},$$

where $\mathbf{f}$, $\mathbf{u}$, $\mathbf{s}$, and $\mathbf{n} : \Omega \to \mathbb{R}$, where $\Omega = \{(x_1, x_2) : x_1 = 1, 2, \cdots, M, x_2 = 1, 2, \cdots, N\}$ (M and N denote the number of columns and rows of the 2D gray-scale image, respectively), represent the observed image, the desired clear image, stipe noise, and additive Gaussian white noise, respectively. Stripes are generally assumed to be vertical (x_2-direction). If the stripes are horizontal, one can rotate them to make the stripes vertical.

First, Chang et al. [19] introduced an image decomposition model that simultaneously models the characteristics of stripe and image components. Specifically, they utilized a low-rank constraint for stripes and TV regularization for the image, which led to the following model:

$$\min_{\mathbf{u},\mathbf{s}} \ \frac{1}{2}\|\mathbf{f}-\mathbf{u}-\mathbf{s}\|_2^2 + \lambda_1\|\nabla\mathbf{u}\|_1 + \lambda_2\|\mathbf{s}\|_*,$$

where λ_i $(i = 1, 2)$ are regularization parameters. Here, $\|\nabla\mathbf{u}\|_1$ is the anisotropic TV such as $\|\nabla\mathbf{u}\|_1 = \|\nabla_{x_1}\mathbf{u}\|_1 + \|\nabla_{x_2}\mathbf{u}\|_1$, where $\nabla\mathbf{u} = (\nabla_{x_1}\mathbf{u}, \nabla_{x_2}\mathbf{u})^T$ with ∇_{x_1} and ∇_{x_2} denoting the horizontal and vertical derivative operators, respectively. $\|A\|_*$ represents the nuclear norm of the matrix A, defined as the sum of its singular values, i.e., $\|A\|_* = \sum_i \sigma_i(A)$. This model performs well for handling vertical stripes, but is not explicitly applicable to oblique stripes.

Wang et al. [25] proposed a novel unified destriping model to effectively exploit the low-rankness and group sparsity of the oblique stripe noise:

$$\min_{\mathbf{u},\mathbf{s}} \ \frac{1}{2}\|\mathbf{f}-\mathbf{u}-\mathbf{s}\|_2^2 + \lambda_1\|\nabla_{x_1}\mathbf{u}\|_1 + \lambda_2\|\tau\circ\mathbf{s}\|_* + \lambda_3\|\tau\circ\mathbf{s}\|_{2,1},$$

where $\lambda_i > 0$ $(i = 1, 2, 3)$ are parameters, and τ is the shear operator to transform the oblique stripe noise vertically, which does not include rotation and filling up. $\|\mathbf{s}\|_{2,1} = \sum_i \|[\mathbf{s}]_i\|_2$ represents the group sparsity, where $[\mathbf{s}]_i$ is the i-th column of $\mathbf{s}$. This model attains excellent performance on thin and regular stripes, but it cannot effectively remove agglomerated, banded, and irregular stripes while generating the staircase effect in restored images.

To overcome these drawbacks, the authors in [27] utilized a unidirectional higher-order TV regularization for the image, and the Schatten $1/2$-norm to characterize the low-rankness of stripes:

$$\min_{\mathbf{u},\mathbf{s}} \ \frac{1}{2}\|\mathbf{f}-\mathbf{u}-\mathbf{s}\|_2^2 + \lambda_1\|\nabla_{x_1}\mathbf{u}\|_1 + \lambda_2\|\nabla^2_{x_1x_1}\mathbf{u}\|_1 + \lambda_3\|\mathbf{s}\|_{s_{1/2}}^{1/2},$$

where $\lambda_i > 0$ $(i = 1, 2, 3)$ are parameters, $\nabla^2_{x_1x_1}$ is the second-order gradient operator across the horizontal direction, and $\|\cdot\|_{s_{1/2}}^{1/2}$ is the Schatten $1/2$-norm defined as $\|\mathbf{s}\|_{s_{1/2}}^{1/2} = \sum_i \sigma_i^{1/2}$ with all singular values σ_i of $\mathbf{s}$. Despite the phenomenal destriping performance of the aforementioned models, the low-rank prior for stripe noise may be violated in remote sensing images, such as the stripes with small fragment cases.

On the other hand, there are various works [20, 21, 23, 24] that utilize the directional property of stripes, i.e., unidirectional gradient sparsity regularization of $\mathbf{s}$. Among them, the authors in [20] proposed an image decomposition model that can handle both stripes and Gaussian noise:

$$\min_{\mathbf{u},\mathbf{s}} \ \frac{1}{2}\|\mathbf{f}-\mathbf{u}-\mathbf{s}\|_2^2 + \lambda_1\|\nabla_{x_1}\mathbf{u}\|_1 + \lambda_2\|\nabla_{x_2}\mathbf{u}\|_1 + \lambda_3\|\nabla_{x_2}\mathbf{s}\|_1 + \lambda_4\|\mathbf{s}\|_{2,1},$$

where $\lambda_i > 0$ $(i = 1, 2, 3, 4)$ are parameters. The term $\|\nabla_{x_2}\mathbf{s}\|_1$ enforces that the stripe component has good smoothness in vertical direction.

2.2. Fractional-order total variation

This subsection recalls fractional-order derivatives and FTV. Fractional-order derivatives are seen as a generalization of the integer-order derivatives. Three well-known definitions of fractional-order derivatives are the Riemann-Liouville, Grünwald-Letnikov, and Caputo definitions [64–66]. In

particular, the Grünwald–Letnikov (GL) fractional-order derivative is based on the finite difference method and is easy to implement. For one-dimensional signals $f(x)$, $x \in [a, x]$, the GL fractional α-order derivative is defined as

$$D^{\alpha} f(x) = \frac{d^{\alpha} f(x)}{dx^{\alpha}} = \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{j=0}^{[\frac{x-a}{h}]} (-1)^j \binom{\alpha}{j} f(x - jh),$$

where $\alpha > 0$, $[b]$ is the integer such that $b - 1 < [b] \leq b$, and $\binom{\alpha}{j} = \frac{\alpha(\alpha-1)\cdots(\alpha-j+1)}{j!}$ is the combination parameter.

For a function $\mathbf{u} : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$ is an open and bounded set with compact support, let $\nabla^{\alpha}_{x_i} \mathbf{u} = \frac{\partial^{\alpha} \mathbf{u}}{\partial x_i^{\alpha}}$ $(i = 1, 2)$ be the fractional α-order derivative $D^{\alpha}\mathbf{u}$ of $\mathbf{u}$ along the x_i-direction. Then, the anisotropic fractional α-order TV is defined as

$$\int_{\Omega} | \nabla^{\alpha} \mathbf{u} | \, d\mathbf{x} = \int_{\Omega} | \nabla^{\alpha}_{x_1} \mathbf{u} | + | \nabla^{\alpha}_{x_2} \mathbf{u} | \, d\mathbf{x}, \tag{2.1}$$

while its isotropic version is given by $\int_{\Omega} \sqrt{(\nabla^{\alpha}_{x_1} \mathbf{u})^2 + (\nabla^{\alpha}_{x_2} \mathbf{u})^2} \, d\mathbf{x}$.

If $\Omega = \{(x_1, x_2) : x_1 = 1, 2, \cdots, M, x_2 = 1, 2, \cdots, N\}$ is a discretized domain, then an image $\mathbf{u}$ defined on Ω can be represented as a matrix in $\mathbb{R}^{N \times M}$, and $\mathbf{u}_{i,j}$ denotes the (i, j)-th element of $\mathbf{u}$ $(i = 1, ..., M, \, j = 1, ..., N)$. Then, its discrete fractional-order derivatives $\nabla^{\alpha}_{x_1} \mathbf{u}$ and $\nabla^{\alpha}_{x_2} \mathbf{u}$ are given by

$$(\nabla^{\alpha}_{x_1} \mathbf{u})_{i,j} = \sum_{k=0}^{K-1} (-1)^k C^{\alpha}_k \mathbf{u}_{i-k,j}, \quad (\nabla^{\alpha}_{x_2} \mathbf{u})_{i,j} = \sum_{k=0}^{K-1} (-1)^k C^{\alpha}_k \mathbf{u}_{i,j-k},$$

where K is the number of neighboring pixels used in computation of the fractional-order derivatives at each pixel, and C^{α}_k denotes the generalized binomial coefficients as $C^{\alpha}_k = \frac{\Gamma(\alpha+1)}{\Gamma(k+1)\Gamma(\alpha-k+1)}$, with $\Gamma(\cdot)$ denoting the Gamma function. For fixed α, the coefficients C^{α}_k rapidly tend to zero as k increases. Then, the discretized version of FTV in (2.1) is given by $\|\nabla^{\alpha}\mathbf{u}\|_1 = \sum_{i,j} | (\nabla^{\alpha}_{x_1} \mathbf{u})_{i,j} | + | (\nabla^{\alpha}_{x_2} \mathbf{u})_{i,j} |$. According to [43], the high-pass capability becomes stronger with larger α, so more texture regions are preserved when α increases. The experimental results in the literature [41–48] show that FTV performs well in terms of removing the staircase effect while preserving textures.

3. Proposed model and algorithm

In this section, we introduce a denoising model to remove both stripe noise and Gaussian noise. We also present an optimization algorithm for solving the proposed model.

3.1. Proposed denoising model

We assume that the observed noisy image $\mathbf{f} : \Omega \to \mathbb{R}$ is degraded by both stripe noise and Gaussian noise as follows:

$$\mathbf{f}(x_1, x_2) = \mathbf{u}(x_1, x_2) + \mathbf{s}(x_1, x_2) + \mathbf{n}(x_1, x_2), \tag{3.1}$$

where $\Omega = \{(x_1, x_2) : x_1 = 1, 2, \cdots, M,\ x_2 = 1, 2, \cdots, N\}$. Here, $\mathbf{u}$ is the clean image, $\mathbf{s}$ represents the periodic or non-periodic stripes that are vertical (x_2-direction), and $\mathbf{n}$ represents the Gaussian noise following the normal distribution, $\mathcal{N}(0, \sigma^2)$, with a standard deviation σ.

In this work, unlike in previous works, the Gaussian noise level is considered to be relatively high, hence we intend to restore the image $\mathbf{u}$ by eliminating both stripe noise and Gaussian noise simultaneously. To effectively retrieve $\mathbf{u}$ from the data $\mathbf{f}$ in (3.1), we propose the following image decomposition model:

$$\min_{\mathbf{u},\mathbf{s}} \ \frac{1}{2}\|\mathbf{f} - \mathbf{u} - \mathbf{s}\|_2^2 + \lambda_1\langle\phi(|\ \nabla_{x_1}^{\alpha}\mathbf{u}\ |), \mathbf{1}\rangle + \lambda_2\langle\phi(|\ \nabla_{x_2}^{\alpha}\mathbf{u}\ |), \mathbf{1}\rangle + \lambda_3\|\nabla_{x_2}\mathbf{s}\|_1 + \lambda_4\sum_{x_1}\psi(\|\mathbf{s}(x_1, \cdot)\|_2), \quad (3.2)$$

where $\langle\cdot,\cdot\rangle$ denotes the inner product, $\lambda_i > 0$ ($i = 1, 2, 3, 4$) are regularization parameters, $1 < \alpha < 2$, and $\|\mathbf{s}(x_1, \cdot)\|_2 = \sqrt{\sum_{x_2}\mathbf{s}(x_1, x_2)^2}$. Moreover, ϕ and ψ are given by the following nonconvex functions

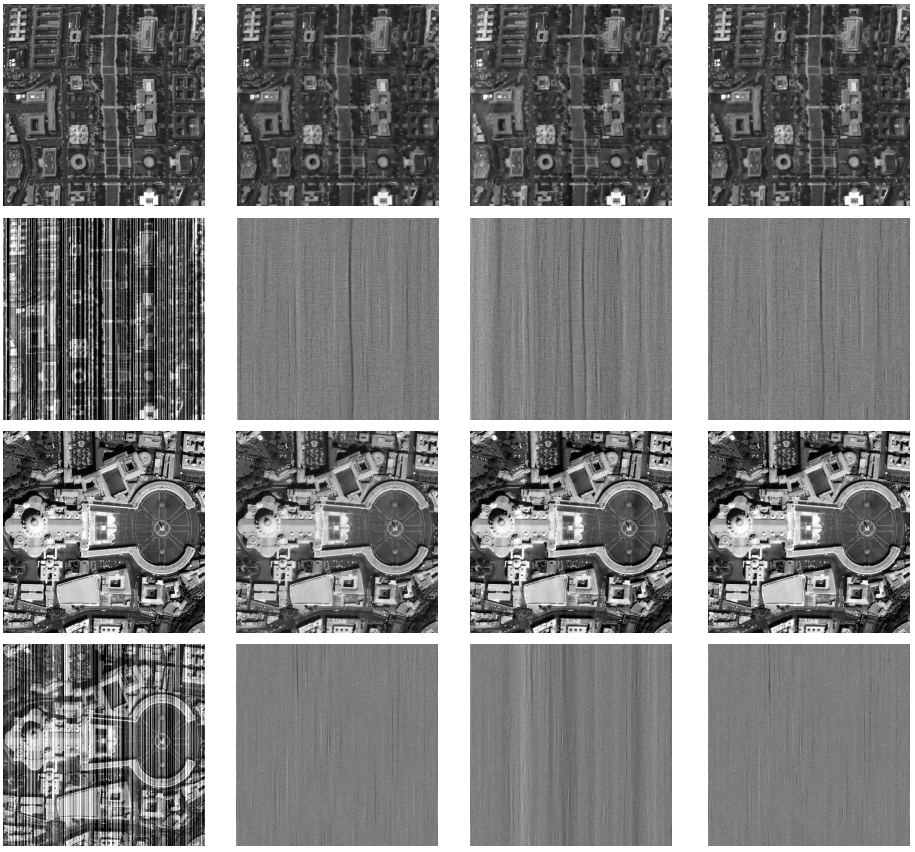
$$\phi(v) = \frac{1}{\rho}\log(1 + \rho v), \quad \psi(w) = \log(\beta + w),$$

where $\rho > 0$ controls the nonconvexity of FTV, and $\beta > 0$ is a small parameter.

The first term in (3.2) is a data-fidelity term that estimates the discrepancy between $\mathbf{f}$ and $\mathbf{u} + \mathbf{s}$. The second and third terms control the smoothness of $\mathbf{u}$, which is a nonconvex version of the anisotropic FTV. This FTV alleviates the staircase effect that is commonly seen in restored images from TV-based models. The nonconvex function ϕ does not penalize the formulation with strong gradients of $\mathbf{u}$, thus protecting large details and textures in the image. On the other hand, at near-zero points ($v \to 0+$), it is preferable for $\phi(v)$ to have the same behavior as the linear function, v, so that $\mathbf{u}$ can be better smoothed in homogeneous regions of the image. That is, the nonconvex function ϕ further enforces the preservation of edges or discontinuties, so the use of the nonconvex FTV (NFTV) leads to higher PSNR and SSIM values than using FTV, as shown in Figure 1. In fact, there are more choices for the nonconvex function ϕ such as v^q ($0 < q < 1$) and $\frac{v}{1+\rho v}$. However, it is hard to solve the minimization problem involving the nonconvex q-norm regularizer because finding a limiting-supergradient of $\|\cdot\|_q$ at zero is difficult. Besides, the fractional nonconvex function is more proper for the reconstruction of piecewise-constant images since $\lim_{v\to\infty}\phi(v) = c$ (c is constant), while the logarithmic function is suitable for the reconstruction of images that are no longer piecewise-constant since $\lim_{v\to\infty}\phi(v) = \infty$, as explained in [67]. There are many real synthetic aperture radar (SAR) images that are not piecewise-constant; thus, we utilize the logarithmic function as our nonconvex function ϕ. The regularization parameters λ_1 and λ_2 may be different. For example, when the Gaussian noise level is low, λ_1 may be chosen to be larger than λ_2. However, as the Gaussian noise level increases, smoothing of $\mathbf{u}$ along the x_2-direction is also necessary, so λ_2 must be close to λ_1. In practice, we consider two Gaussian noise levels, $\sigma = 10$ or 20, which are relatively high, so we set the values of λ_1 and λ_2 be the same.

The last two terms in (3.2) exploit the directional characteristics and group sparsity of stripe noise. In particular, the fourth term is a unidirectional TV regularization of the stripe component that imposes the sparsity of its vertical derivatives. Furthermore, the stripe component is composed of stripe lines and stripe-free lines, and each line can be viewed as a group. Thus, $\sum_{x_1}\|\mathbf{s}(x_1, \cdot)\|_2$ enforces the group sparsity (GS) of stripes. However, $\ell_{2,1}$-norm is not able to effectively promote the group sparsity of stripes in many cases [26, 59]. As a result, we characterize the intrinsic structure of stripes using a nonconvex function ψ for $\|\mathbf{s}(x_1, \cdot)\|_2$. Likewise, w^p ($0 < p < 1$) or $\frac{w}{\beta+w}$ could be other options for the

nonconvex function ψ. Figure 1 shows a comparison of the proposed model (3.2) with a model (3.2) that includes the GS term, $\sum_{x_1} \|\mathbf{s}(x_1, \cdot)\|_2$, instead of the nonconvex GS (NGS) term. The model (3.2) with the GS term fails to properly extract the stripe noise, resulting in some traces of stripes in the restored images. This can also be seen more clearly in the difference images in the second and forth rows of Figure 1. Meanwhile, the proposed model (3.2), which includes the NGS term, more suitably extracts the stripe noise from the images. This also leads to the restored images with better conserved details and edges.



Original $\mathbf{u}_*$ & $\mathbf{f}$ (a) FTV+NGS (b) NFTV+GS (c) NFTV+NGS

Figure 1. Restored images of the proposed model with (a) FTV and NGS terms, (b) NFTV and GS terms, (c) NFTV and NGS terms. Data $\mathbf{f}$ with non-periodic stripes with $(r, m) = (70, 100)$ and Gaussian noise with $\sigma = 10$. First and third rows: original $\mathbf{u}_*$ and restored $\mathbf{u}$, second and fourth rows: data $\mathbf{f}$ and $\mathbf{u} - \mathbf{u}_*$. PSNR/SSIM: (top) (a) 27.67/0.8767, (b) 24.92/0.8530, (c) **27.98/0.8813**, (bottom) (a) 27.41/0.8779, (b) 25.84/0.8666, (c) **27.83/0.8849**.

3.2. Optimization algorithms for solving model (3.2)

In this section, we present an optimization algorithm for solving the proposed model in (3.2). Given the matrix $\mathbf{f} \in \mathbb{R}^{N\times M}$, model (3.2) can be rewritten as

$$\min_{\mathbf{u},\mathbf{s}} \ \frac{1}{2}\|\mathbf{f} - \mathbf{u} - \mathbf{s}\|_2^2 + \lambda_1\langle\phi(|\ \nabla_{x_1}^{\alpha}\mathbf{u}\ |), \mathbf{1}\rangle + \lambda_2\langle\phi(|\ \nabla_{x_2}^{\alpha}\mathbf{u}\ |), \mathbf{1}\rangle + \lambda_3\|\nabla_{x_2}\mathbf{s}\|_1 + \lambda_4 \sum_i \log(\beta + \|[\mathbf{s}]_i\|_2), \quad (3.3)$$

where $[\mathbf{s}]_i$ is the i-th column of $\mathbf{s}$, with $i = 1, 2, ...M$.

To solve the nonconvex problem (3.3), we first employ the IRL1 proposed in [58] for solving a nonconvex minimization problem. Let us consider the following nonconvex minimization problem: $\min_z \ E_1(z)+E_2(G(z))$, where E_1 is a proper, lower semicontinuous and convex function, E_2 is a concave and coordinatewise nondecreasing function, and G is a coordinatewise convex function. Applying IRL1 to this problem leads to the following iterative algorithm:

$$\begin{cases} w^{\ell+1} & \in & \bar{\partial}E_2(G(z^\ell)), \\ z^{\ell+1} & := & \arg\min\limits_z \ E_1(z) + \langle w^{\ell+1}, G(z)\rangle, \end{cases} \tag{3.4}$$

where $\bar{\partial}E_2 := -\partial(-E_2)$ denotes the *superdifferential* of the function E_2. For the global convergence of IRL1, it is required that $E_1(z) + \langle w^{\ell+1}, G(z)\rangle$ is strongly convex with a constant independent of ℓ. Thus, the authors in [58] suggest a modified version of IRL1 by adding a proximal term $\frac{\delta}{2}\|z - z^\ell\|_2^2$ to the convex surrogate problem in (3.4) with arbitrarily small $\delta > 0$.

To apply IRL1 to model (3.3), we can set up as follows:

$$\begin{aligned} E_1(\mathbf{u}, \mathbf{s}) &= \frac{1}{2}\|\mathbf{f} - \mathbf{u} - \mathbf{s}\|_2^2 + \lambda_3\|\nabla_{x_2}\mathbf{s}\|_1, \\ E_2(\mathbf{v}_1, \mathbf{v}_2, \mathbf{t}) &= \lambda_1\langle\phi(\mathbf{v}_1), \mathbf{1}\rangle + \lambda_2\langle\phi(\mathbf{v}_2), \mathbf{1}\rangle + \lambda_4\sum_i \log(\beta + \mathbf{t}_i), \\ G(\mathbf{u}, \mathbf{s}) &= (|\ \nabla_{x_1}^\alpha\mathbf{u}\ |, |\ \nabla_{x_2}^\alpha\mathbf{u}\ |, \|[\mathbf{s}]_1\|_2, \|[\mathbf{s}]_2\|_2, ..., \|[\mathbf{s}]_M\|_2), \end{aligned} \tag{3.5}$$

where $\mathbf{t} = (\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_M)$.

Then, we adopt the modified IRL1 in [58], which is obtained by adding two proximal terms, $\frac{\delta}{2}\|\mathbf{u} - \mathbf{u}^\ell\|_2^2$ and $\frac{\delta}{2}\|\mathbf{s} - \mathbf{s}^\ell\|_2^2$:

$$\begin{cases} \mathbf{w}_1^{\ell+1} = \dfrac{1}{1 + \rho\ |\ \nabla_{x_1}^\alpha\mathbf{u}^\ell\ |}, \quad \mathbf{w}_2^{\ell+1} = \dfrac{1}{1 + \rho\ |\ \nabla_{x_2}^\alpha\mathbf{u}^\ell\ |}, \quad (\mathbf{w}_3^{\ell+1})_i = \dfrac{1}{\beta + \|[\mathbf{s}^\ell]_i\|_2}, \quad i = 1, \cdots, M \\ (\mathbf{u}^{\ell+1}, \mathbf{s}^{\ell+1}) := \arg\min\limits_{\mathbf{u},\mathbf{s}} \ \dfrac{1}{2}\|\mathbf{f} - \mathbf{u} - \mathbf{s}\|_2^2 + \lambda_3\|\nabla_{x_2}\mathbf{s}\|_1 + \lambda_1\langle\mathbf{w}_1^{\ell+1}, |\ \nabla_{x_1}^\alpha\mathbf{u}\ |\rangle + \lambda_2\langle\mathbf{w}_2^{\ell+1}, |\ \nabla_{x_2}^\alpha\mathbf{u}\ |\rangle \\ \qquad\qquad +\lambda_4\sum\limits_i(\mathbf{w}_3^{\ell+1})_i\|[\mathbf{s}]_i\|_2 + \dfrac{\delta}{2}\|\mathbf{u} - \mathbf{u}^\ell\|_2^2 + \dfrac{\delta}{2}\|\mathbf{s} - \mathbf{s}^\ell\|_2^2, \end{cases} \tag{3.6}$$

where $(\lambda_1\mathbf{w}_1, \lambda_2\mathbf{w}_2, \lambda_4\mathbf{w}_3)^T \in \bar{\partial}E_2(G(\mathbf{u}, \mathbf{s}))$ with $\bar{\partial}E_2 = \nabla E_2$, and $\delta > 0$ is a small parameter.

Based on the convergence of IRL1 in [58], we prove the global convergence of IRL1 (3.6) as follows:

Theorem 1. *Let $\{(\boldsymbol{u}^\ell, \boldsymbol{s}^\ell)\}$ be the sequence generated by the IRL1 (3.6). Then, $\{(\boldsymbol{u}^\ell, \boldsymbol{s}^\ell)\}$ converges to $(\boldsymbol{u}^*, \boldsymbol{s}^*)$ as $\ell \to \infty$, where $(\boldsymbol{u}^*, \boldsymbol{s}^*)$ is a critical point of (3.3). Furthermore, the sequence $\{(\boldsymbol{u}^\ell, \boldsymbol{s}^\ell)\}$ has finite length: $\sum_{\ell=0}^\infty \|\boldsymbol{u}^\ell - \boldsymbol{u}^{\ell+1}\|_2 + \|\boldsymbol{s}^\ell - \boldsymbol{s}^{\ell+1}\|_2 < \infty$.*

Proof. We need to check the assumptions in Theorem 2 in [58]. First, the objective function in (3.3) (or $\mathcal{E}(\mathbf{u}, \mathbf{s}) = E_1(\mathbf{u}, \mathbf{s}) + E_2(G(\mathbf{u}, \mathbf{s}))$ from (3.5)) is clearly coercive and bounded below. In addition, due to the proximal terms, $\frac{\delta}{2}\|\mathbf{u} - \mathbf{u}^\ell\|_2^2$ and $\frac{\delta}{2}\|\mathbf{s} - \mathbf{s}^\ell\|_2^2$, the objective function of the convex subproblem in (3.6) is strongly convex with a constant independent of ℓ. Next, we show that the following three assumptions are satisfied:

(a) The objective function $\mathcal{E}(\mathbf{u}, \mathbf{s})$ has the Kurdyka–Łojasiewicz (KL) property at a cluster point.
(b) $E_2(\mathbf{v}_1, \mathbf{v}_2, \mathbf{t})$ has locally Lipschitz continuous gradients on a compact set containing all the points $G(\mathbf{u}^\ell, \mathbf{s}^\ell)$.
(c) The convex function $z \mapsto \langle \mathbf{w}^{\ell+1}, z\rangle$ for all ℓ, where $z = (\mathbf{v}_1, \mathbf{v}_2, \mathbf{t})^T$ and $\mathbf{w} = (\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3)^T$, has a globally Lipschitz continuous gradient with a common Lipschitz constant.

a) A function is called a KL function if the function is lower semicontinuous and KL inequality holds for every point in the domain. According to [68], polynomials, indicator function, $\|\cdot\|_1$, and $\|\cdot\|_2$ are KL functions. Moreover, the log and exponential functions are also KL functions. Indeed, they are included in the log–exp structure [69], and the functions that are definable in such an o-minimal structure have the KL property. Hence, the objective function in (3.3) is a KL function.

b) The gradient and Hessian of E_2 are given by

$$\nabla E_2(\mathbf{v}_1, \mathbf{v}_2, \mathbf{t}) = \left(\frac{\lambda_1}{1+\rho\mathbf{v}_1}, \frac{\lambda_2}{1+\rho\mathbf{v}_2}, \frac{\lambda_4}{\beta+\mathbf{t}_1}, ..., \frac{\lambda_4}{\beta+\mathbf{t}_M}\right)^T,$$

$$\nabla^2 E_2(\mathbf{v}_1, \mathbf{v}_2, \mathbf{t}) = \begin{bmatrix} -\frac{\lambda_1\rho}{(1+\rho\mathbf{v}_1)^2} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & -\frac{\lambda_2\rho}{(1+\rho\mathbf{v}_2)^2} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\frac{\lambda_4}{(\beta+\mathbf{t}_1)^2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & -\frac{\lambda_4}{(\beta+\mathbf{t}_M)^2} \end{bmatrix}.$$

Hence, $\|\nabla^2 E_2\|_\infty \le \max(\lambda_1\rho, \lambda_2\rho, \lambda_4/\beta^2)$ on the image of G. Thus, E_2 has a Lipschitz continuous gradient.

c) Trivially, $z \mapsto \langle \mathbf{w}^{\ell+1}, z\rangle$ has a globally Lipschitz continuous gradient with a common Lipschitz constant 0.

Therefore, all the assumptions of Theorem 2 in [58] are satisfied, so the theorem is proved. □

Now we solve the $(\mathbf{u}, \mathbf{s})$-subproblem in IRL1 (3.6). This subproblem is convex, but it involves non-differentiable terms. To resolve this problem, numerous efficient convex optimization algorithms have been suggested, such as [60–62, 70, 71]. In particular, we adopt the alternating direction method of multipliers (ADMM) in [60–62]. The ADMM is a widely-known algorithm for solving linearly constrained convex optimization problems, with its convergence proven in [61, 62].

First, we introduce auxiliary variables $\mathbf{p}_i$ $(i = 1, 2, 3, 4)$, based on the variable splitting technique. Hence, the $(\mathbf{u}, \mathbf{s})$-subproblem in (3.6) can be converted into the following constrained minimization problem:

$$\begin{aligned} \min_{\mathbf{u},\mathbf{s},\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4} \ & \frac{1}{2}\|\mathbf{f}-\mathbf{u}-\mathbf{s}\|_2^2 + \lambda_1\langle \mathbf{w}_1^{\ell+1}, |\ \mathbf{p}_1\ |\rangle + \lambda_2\langle \mathbf{w}_2^{\ell+1}, |\ \mathbf{p}_2\ |\rangle \\ & +\lambda_3\|\mathbf{p}_3\|_1 + \lambda_4\sum_i (\mathbf{w}_3^{\ell+1})_i \|[\mathbf{p}_4]_i\|_2 + \frac{\delta}{2}\|\mathbf{u}-\mathbf{u}^\ell\|_2^2 + \frac{\delta}{2}\|\mathbf{s}-\mathbf{s}^\ell\|_2^2, \\ \text{subject to:} \quad & \mathbf{p}_1 = \nabla_{x_1}^\alpha \mathbf{u}, \ \ \mathbf{p}_2 = \nabla_{x_2}^\alpha \mathbf{u}, \ \ \mathbf{p}_3 = \nabla_{x_2}\mathbf{s}, \ \ \mathbf{p}_4 = \mathbf{s}. \end{aligned} \tag{3.7}$$

The augmented Lagrangian function of problem (3.7) is given by

$$\begin{aligned}\mathcal{L}_\mu(\mathbf{u},\mathbf{s},\vec{\mathbf{p}},\vec{\mathbf{h}}) &= \frac{1}{2}\|\mathbf{f}-\mathbf{u}-\mathbf{s}\|_2^2+\lambda_1\langle \mathbf{w}_1^{\ell+1},|\ \mathbf{p}_1\ |\rangle+\lambda_2\langle \mathbf{w}_2^{\ell+1},|\ \mathbf{p}_2\ |\rangle\\ &+\lambda_3\|\mathbf{p}_3\|_1+\lambda_4\sum_i(\mathbf{w}_3^{\ell+1})_i\|[\mathbf{p}_4]_i\|_2+\frac{\delta}{2}\|\mathbf{u}-\mathbf{u}^\ell\|_2^2+\frac{\delta}{2}\|\mathbf{s}-\mathbf{s}^\ell\|_2^2\\ &-\langle \mathbf{h}_1,\mathbf{p}_1-\nabla_{x_1}^\alpha\mathbf{u}\rangle+\frac{\mu}{2}\|\mathbf{p}_1-\nabla_{x_1}^\alpha\mathbf{u}\|_2^2-\langle \mathbf{h}_2,\mathbf{p}_2-\nabla_{x_2}^\alpha\mathbf{u}\rangle+\frac{\mu}{2}\|\mathbf{p}_2-\nabla_{x_2}^\alpha\mathbf{u}\|_2^2\\ &-\langle \mathbf{h}_3,\mathbf{p}_3-\nabla_{x_2}\mathbf{s}\rangle+\frac{\mu}{2}\|\mathbf{p}_3-\nabla_{x_2}\mathbf{s}\|_2^2-\langle \mathbf{h}_4,\mathbf{p}_4-\mathbf{s}\rangle+\frac{\mu}{2}\|\mathbf{p}_4-\mathbf{s}\|_2^2,\end{aligned}$$

where $\vec{\mathbf{p}}=(\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4)^T$, $\vec{\mathbf{h}}=(\mathbf{h}_1,\mathbf{h}_2,\mathbf{h}_3,\mathbf{h}_4)^T$, where $\mathbf{h}_i\in\mathbb{R}^{N\times M\times 2}$ $(i=1,2,3)$ and $\mathbf{h}_4\in\mathbb{R}^{N\times M}$ are the Lagrangian multipliers, and $\mu>0$ is a penalty parameter.

Then, the ADMM applied to (3.7) brings the following iterative algorithm:

$$\begin{cases}(\mathbf{u}^{k+1},\mathbf{s}^{k+1}):=\arg\min\limits_{\mathbf{u},\mathbf{s}}\ \mathcal{L}_\mu(\mathbf{u},\mathbf{s},\vec{\mathbf{p}}^k,\vec{\mathbf{h}}^k)\\ \vec{\mathbf{p}}^{k+1}:=\arg\min\limits_{\vec{\mathbf{p}}}\ \mathcal{L}_\mu(\mathbf{u}^{k+1},\mathbf{s}^{k+1},\vec{\mathbf{p}},\vec{\mathbf{h}}^k),\\ \mathbf{h}_1^{k+1}=\mathbf{h}_1^k-\gamma\mu(\mathbf{p}_1^{k+1}-\nabla_{x_1}^\alpha\mathbf{u}^{k+1}),\\ \mathbf{h}_2^{k+1}=\mathbf{h}_2^k-\gamma\mu(\mathbf{p}_2^{k+1}-\nabla_{x_2}^\alpha\mathbf{u}^{k+1}),\\ \mathbf{h}_3^{k+1}=\mathbf{h}_3^k-\gamma\mu(\mathbf{p}_3^{k+1}-\nabla_{x_2}\mathbf{s}^{k+1}),\\ \mathbf{h}_4^{k+1}=\mathbf{h}_4^k-\gamma\mu(\mathbf{p}_4^{k+1}-\mathbf{s}^{k+1}),\end{cases}\tag{3.8}$$

where $\gamma\in(0,\frac{\sqrt{5}+1}{2})$. We can attain the following convergence results according to the convergence in [61]:

Theorem 2. *If the sequence $\{(\boldsymbol{u}^k,\boldsymbol{s}^k,\vec{\boldsymbol{p}}^k,\vec{\boldsymbol{h}}^k)\}$ is generated by ADMM in (3.8) and $\gamma\in(0,\frac{\sqrt{5}+1}{2})$, then $\{(\boldsymbol{u}^k,\boldsymbol{s}^k,\vec{\boldsymbol{p}}^k)\}$ strongly converges to a limit point $(\boldsymbol{u}_*,\boldsymbol{s}_*,\vec{\boldsymbol{p}}_*)$, $\{\vec{\boldsymbol{h}}^{k+1}-\vec{\boldsymbol{h}}^k\}$ converges to $\boldsymbol{0}$, and $\{\vec{\boldsymbol{h}}^k\}$ is bounded. Moreover, if $\vec{\boldsymbol{h}}_*$ is a weak cluster point of $\vec{\boldsymbol{h}}^k$, then $(\boldsymbol{u}_*,\boldsymbol{s}_*,\vec{\boldsymbol{p}}_*,\vec{\boldsymbol{h}}_*)$ is a saddle point of the augmented Lagrangian $\mathcal{L}_\mu$.*

Now we solve the $(\mathbf{u},\mathbf{s})$-subproblem in ADMM (3.8). This subproblem can be reformulated as the following least squares problem:

$$\begin{aligned}(\mathbf{u}^{k+1},\mathbf{s}^{k+1}) &:= \arg\min_{\mathbf{u},\mathbf{s}}\ \frac{1}{2}\|\mathbf{f}-\mathbf{u}-\mathbf{s}\|_2^2+\frac{\delta}{2}\|\mathbf{u}-\mathbf{u}^\ell\|_2^2+\frac{\delta}{2}\|\mathbf{s}-\mathbf{s}^\ell\|_2^2\\ &+\frac{\mu}{2}\|\nabla_{x_1}^\alpha\mathbf{u}-\mathbf{p}_1^k+\mathbf{h}_1^k/\mu\|_2^2+\frac{\mu}{2}\|\nabla_{x_2}^\alpha\mathbf{u}-\mathbf{p}_2^k+\mathbf{h}_2^k/\mu\|_2^2\\ &+\frac{\mu}{2}\|\nabla_{x_2}\mathbf{s}-\mathbf{p}_3^k+\mathbf{h}_3^k/\mu\|_2^2+\frac{\mu}{2}\|\mathbf{s}-\mathbf{p}_4^k+\mathbf{h}_4^k/\mu\|_2^2.\end{aligned}$$

The first-order optimality condition leads to the following normal equation:

$$\begin{bmatrix}B_1 & I\\ I & B_2\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{s}\end{bmatrix}=\begin{bmatrix}RHS_u\\ RHS_s\end{bmatrix},\tag{3.9}$$

where B_1, B_2, RHS_u, and RHS_s are given by

$$B_1 = (1+\delta)I+\mu(\nabla_{x_1}^\alpha)^T\nabla_{x_1}^\alpha+\mu(\nabla_{x_2}^\alpha)^T\nabla_{x_2}^\alpha,$$

$$\begin{aligned} B_2 &= (1+\delta+\mu)I + \mu(\nabla_{x_2})^T\nabla_{x_2}, \\ RHS_u &= \mathbf{f} + \delta\mathbf{u}^\ell + \mu(\nabla^\alpha_{x_1})^T(\mathbf{p}_1^k - \mathbf{h}_1^k/\mu) + \mu(\nabla^\alpha_{x_2})^T(\mathbf{p}_2^k - \mathbf{h}_2^k/\mu), \\ RHS_s &= \mathbf{f} + \delta\mathbf{s}^\ell + \mu(\nabla_{x_2})^T(\mathbf{p}_3^k - \mathbf{h}_3^k/\mu) + \mu(\mathbf{p}_4^k - \mathbf{h}_4^k/\mu). \end{aligned}$$

Here, $(\nabla^\alpha)^T = (-1)^\alpha \mathrm{div}^\alpha$, where $\mathrm{div}^\alpha \mathbf{q} \in \mathbb{R}^{N\times M}$ for $\mathbf{q} = (\mathbf{q}_1, \mathbf{q}_2) \in \mathbb{R}^{N\times M\times 2}$ is the discrete fractional-order divergence defined as

$$(\mathrm{div}^\alpha \mathbf{q})_{i,j} = (\nabla^\alpha_{x_1}\mathbf{q}_1)_{i,j} + (\nabla^\alpha_{x_2}\mathbf{q}_2)_{i,j}.$$

The elements B_1, B_2, and I in Eq (3.9) can be diagonalized by the 2-dimensional fast Fourier transform (FFT2) under the periodic boundary condition. Thus, the block matrix in the left-hand side of Eq (3.9) can be diagonalized by using FFT2. Therefore, the solution $(\mathbf{u}^{\ell+1}, \mathbf{s}^{\ell+1})$ can be obtained exactly using the inversion formula of the block matrix.

Next, we solve the $\vec{\mathbf{p}}$-subproblem in ADMM (3.8). The variables $\mathbf{p}_i$ are independent of each other, so we can solve the subproblem for each $\mathbf{p}_i$:

$$\begin{aligned} \mathbf{p}_1^{k+1} &:= \arg\min_{\mathbf{p}_1} \ \lambda_1\langle \mathbf{w}_1^{\ell+1}, |\ \mathbf{p}_1\ |\rangle + \frac{\mu}{2}\|\mathbf{p}_1 - \nabla^\alpha_{x_1}\mathbf{u}^{k+1} - \mathbf{h}_1^k/\mu\|_2^2, \\ \mathbf{p}_2^{k+1} &:= \arg\min_{\mathbf{p}_2} \ \lambda_2\langle \mathbf{w}_2^{\ell+1}, |\ \mathbf{p}_2\ |\rangle + \frac{\mu}{2}\|\mathbf{p}_2 - \nabla^\alpha_{x_2}\mathbf{u}^{k+1} - \mathbf{h}_2^k/\mu\|_2^2, \\ \mathbf{p}_3^{k+1} &:= \arg\min_{\mathbf{p}_3} \ \lambda_3\|p_3\|_1 + \frac{\mu}{2}\|\mathbf{p}_3 - \nabla_{x_2}\mathbf{s}^{k+1} - \mathbf{h}_3^k/\mu\|_2^2, \\ \mathbf{p}_4^{k+1} &:= \arg\min_{\mathbf{p}_4} \ \lambda_4\sum_i (\mathbf{w}_3^{\ell+1})_i\|[\mathbf{p}_4]_i\|_2 + \frac{\mu}{2}\|\mathbf{p}_4 - \mathbf{s}^{k+1} - \mathbf{h}_4^k/\mu\|_2^2. \end{aligned} \tag{3.10}$$

The $\mathbf{p}_1$-subproblem in (3.10) has the closed form solution as

$$\mathbf{p}_1^{k+1} = shrink\left(\nabla^\alpha_{x_1}\mathbf{u}^{k+1} + \mathbf{h}_1^k/\mu, \lambda_1\mathbf{w}_1^{\ell+1}/\mu\right), \tag{3.11}$$

where *shrink* is the soft-thresholding operator defined as

$$shrink(\mathbf{a}, \mathbf{b})_t = \frac{\mathbf{a}_t}{|\ \mathbf{a}_t\ |} \cdot \max(|\ \mathbf{a}_t\ | - \mathbf{b}_t, 0), \quad t \in \Omega.$$

Similarly, $\mathbf{p}_2^{k+1}$, $\mathbf{p}_3^{k+1}$, and $\mathbf{p}_4^{k+1}$ are explicitly achieved as

$$\begin{aligned} \mathbf{p}_2^{k+1} &= shrink\left(\nabla^\alpha_{x_2}\mathbf{u}^{k+1} + \mathbf{h}_2^k/\mu, \lambda_2\mathbf{w}_2^{\ell+1}/\mu\right), \\ \mathbf{p}_3^{k+1} &= shrink\left(\nabla_{x_2}\mathbf{s}^{k+1} + \mathbf{h}_3^k/\mu, \lambda_3/\mu\right), \\ [\mathbf{p}_4^{k+1}]_i &= \frac{[\tilde{\mathbf{p}}_4^k]_i}{\|[\tilde{\mathbf{p}}_4^k]_i\|_2} \cdot \max\left(\|[\tilde{\mathbf{p}}_4^k]_i\|_2 - \lambda_4(\mathbf{w}_3^{\ell+1})_i/\mu, 0\right), \end{aligned} \tag{3.12}$$

with $\tilde{\mathbf{p}}_4^k = \mathbf{s}^{k+1} + \mathbf{h}_4^k/\mu$ and $i = 1, ..., M$.

Consequently, the proposed algorithm is summarized in Algorithm 1.

Algorithm 1 IRL1 for solving model (3.2).

Input: choose the parameters λ_i $(i = 1, 2, 3, 4)$, $\alpha, \beta, \rho, \delta, \mu > 0$, $\gamma \in (0, \frac{\sqrt{5}+1}{2})$ and the maximum numbers of iterations N_{out}, N_{in}.
Initialization: $\mathbf{u}^0 = \mathbf{f}$, $\mathbf{s}^0 = \mathbf{0}$, $\mathbf{p}_i^0 = \mathbf{0}$, $\mathbf{h}_i^0 = \mathbf{0}$ $(i = 1, 2, 3, 4)$.
for $k = 0, 1, 2, \cdots N_{out}$ **do**
  Compute $\mathbf{w}_i^{\ell+1}$ $(i = 1, 2, 3)$ using (3.6),
  for $k = 0, 1, 2, \cdots, N_{in}$ **do**
    Compute $(\mathbf{u}^{k+1}, \mathbf{s}^{k+1})$ by solving Eq (3.9) using FFT2,
    Compute $\mathbf{p}_i^{k+1}$ $(i = 1, 2, 3, 4)$ using (3.11) and (3.12),
    Update $\mathbf{h}_i^{k+1}$ $(i = 1, 2, 3, 4)$ using (3.8),
  end for
end for
Output: restored image $\mathbf{u}$.

4. Numerical experiments

This section presents numerical results for the removal of mixed stripe and Gaussian noise. We compare the performance of the proposed model with existing image decomposition models such as LRSID [19], LRGS [25], TVGS [20], Schatten [27], and ELRTV [30]. In the LRGS, Schatten, and ELRTV models, they use a unidirectional TV for the image component and thus fail to adequately remove the Gaussian noise in the presence of a high level of Gaussian noise. Thus, after removing the stripe noise by adopting these models, we utilize the TV denoising model [35] as a postprocessing step to remove the Gaussian noise. We call these models LRGS+TV, Schatten+TV, and ELRTV+TV, respectively. All numerical results are available in the material at the following link: `https://han.gl/ouTofQ`.

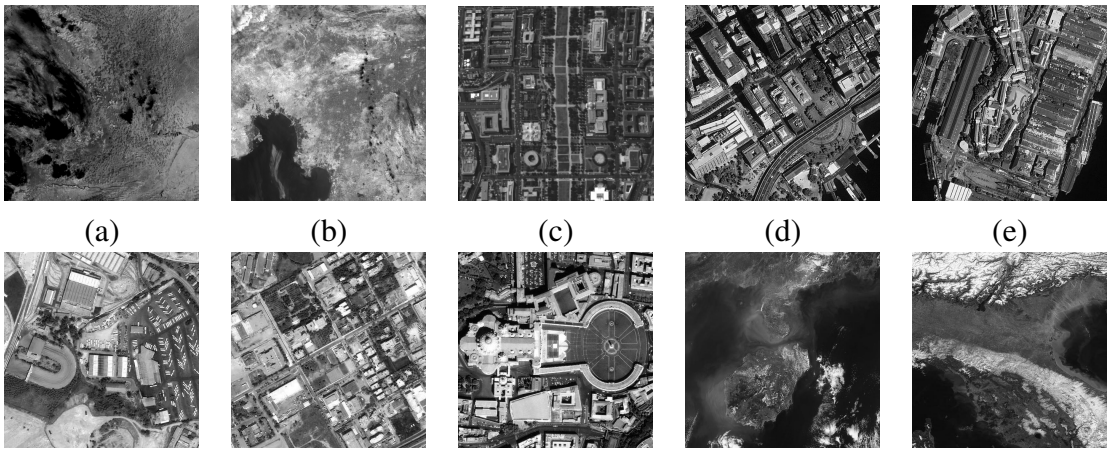
4.1. Experimental setting

The original remote sensing images are given in Figure 2, and the range of intensity values in the original images is assumed to be $[0, 1]$. We consider two types of stripe noise such as periodic or non-periodic stripe noise and assume that the stripes are vertical. Specifically, we randomly select columns of the image to add stripes. In the case of periodic stripes, initial stripes are randomly selected from the first 32 (period=32) columns, and these stripes are periodically added to the original images. On the other hand, non-periodic stripes are randomly selected from the entire columns. The amount of stripe noise is determined by the percentage of degraded region, r, and the intensity of the added stripes, m. In our experiments, we select $r \in \{30, 50, 70\}$ and $m \in \{50, 100\}$. Moreover, the Gaussian noise level, σ, is set to 10 or 20. The numerical experiments were implemented using MATLAB R2020b on a 64-bit Windows 10 operating system using an Intel Xeon Silver CPU at 2.40 GHz and 64 GB memory.

To estimate the quality of the restored images, we compute the peak-signal-to-noise-ratio (PSNR) value, which is defined as

$$\mathrm{PSNR}(\mathbf{u}, \mathbf{u}_*) = 10 \log_{10} \left(\frac{MN}{\|\mathbf{u} - \mathbf{u}_*\|_2^2} \right),$$

where $\mathbf{u}$ and $\mathbf{u}_*$ represent the recovered image and the original image, respectively, and MN is the size of the image. We also calculate the structure similarity (SSIM) index [72], which is a perception-based measure that carries visual information about the structure of the objects.



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

(f) (g) (h) (i) (j)

Figure 2. Original images. (a) MODIS-BAND20 (512×512), (b) BAND20 (512×512), (c) Original band30 (307×307), (d) ikonos rio (512×512), (e) ikonos rio1 (512×512), (f) ikonos helliniko (512×512), (g) ikonos helliniko1 (512×512), (h) vatican (512×512), (i) image02162021 (502×616), (j) image02172021 (512×512).

The stopping criterion of the proposed model is given by

$$\frac{\|\mathbf{u}^{\ell} - \mathbf{u}^{\ell-1}\|_2}{\|\mathbf{u}^{\ell}\|_2} < tol \quad \text{or} \quad \ell > MaxIter,$$

where tol is a given tolerance number, and $MaxIter$ is a given maximum iteration number. For our outer loop, we set $tol = 10^{-4}$ and $N_{out} = 400$, and for our inner loop, $N_{in} = 1$. We use the stopping conditions given in their own works for existing models.

The parameters are tuned to achieve the best visual quality of restored image. The parameter settings of the proposed model are as follows. First, we set the same values for the regularization parameters λ_1 and λ_2, which depend on the level of Gaussian noise. When $\sigma = 10$, λ_1 and λ_2 are chosen from $\{0.02, 0.03\}$, and when $\sigma = 20$, they are chosen from $\{0.04, 0.05, 0.06\}$. Meanwhile, the parameter λ_3 and λ_4 are the regularization parameters for the stripe noise component. λ_3 is fixed at 0.6, while λ_4 is chosen more carefully than λ_3. Specifically, for periodic stripes, $\lambda_4 \in \{0.05, 0.08, 0.1, 0.2, 0.4, 0.6, 0.8\}$, while for non-periodic strips, $\lambda_4 \in \{0.01, 0.03, 0.05, 0.08, 0.1, 0.2, 0.4\}$ when $r = 30$ or 50, and $\lambda_4 \in \{0.005, 0.01, 0.03, 0.05, 0.08\}$ when $r = 70$. The values of λ_1 and λ_4 are presented in each figure. The parameter β in the NGS term is set to 10^{-15}, and the parameter ρ in the NFTV terms is set to 1. The derivative order α in the FTV terms is set to 1.3 or 1.5, and K is set to 20. The parameter δ in IRL1 is fixed at 0.0001. The parameters μ and γ in the ADMM algorithm are set to 0.1 and 1.618, respectively.

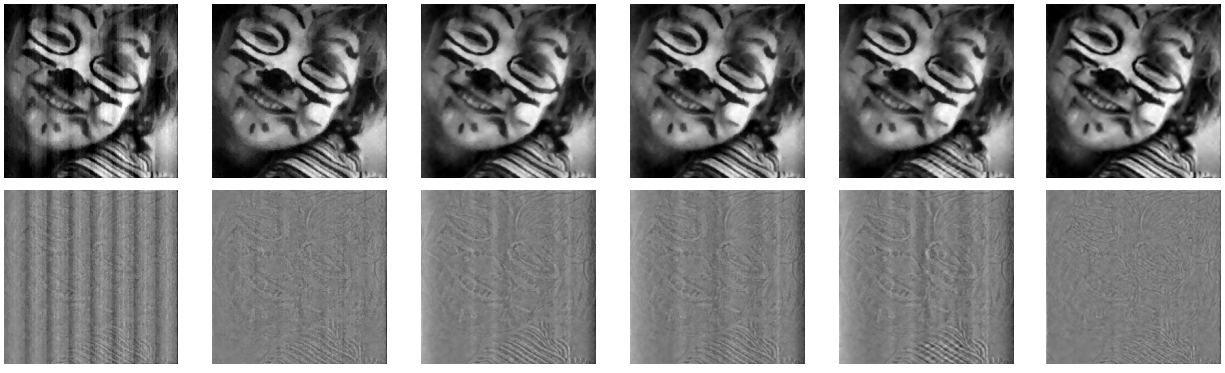
4.2. Denoising results with periodic stripes and Gaussian noise

In this section, we present the denoising results in the presence of periodic stripe noise and Gaussian noise.

First, Figure 3 presents the data images of a natural image in the presence of periodic stripe noise with $r = 50$ or 70 and $m = 100$ and Gaussian noise with $\sigma = 20$, while Figures 4 and 5 present the denoising results. The difference images between the restored and original images are also presented to effectively show the denoising results. First, it can be seen that LRSID, LRGS+TV, Schatten+TV, and ELRTV+TV models fail to correctly decompose the stripe and image components. This leads to the leftover of some stripes or loss of details in the restored images, which are clearly visible in the difference images. Meanwhile, our model and TVGS better extract the stripe noise than the aforementioned models that utilize the low-rankness of stripes. This is due to the use of a directional term and a group sparsity term for the stripes. Comparing our model with TVGS, their restored images look very similar, but we can see from the difference images that our model removes the stripe noise better than TVGS. This shows the effectiveness of our nonconvex group sparsity term for stripes extraction. Moreover, our FTV regularization helps mitigate the staircase artifacts found in the restored images of TVGS. Besides, our nonconvex FTV regularization allows the conservation of finer features and details, so the difference images of our model include much fewer image structures than other models. All these observations result in higher PSNR and SSIM values for our model than TVGS. As a result, these examples show better denoising performance of the proposed model than other models, by effectively removing both stripes and Gaussian noise in a natural image. Since we focus on denoising of SAR images in this work, we put more denoising results for natural images in the supplementary file at the following link: `https://url.kr/6wj8n7`.

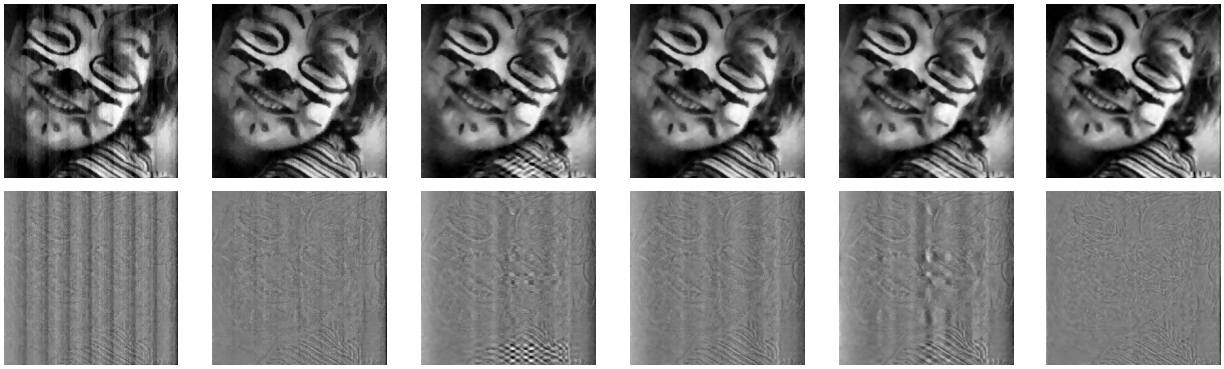
$\mathbf{u}_*$ $\mathbf{f}$ $(r = 50)$ $\mathbf{f}$ $(r = 70)$

Figure 3. Original image $\mathbf{u}_*$ and data images with periodic stripes with $r = 50$ or 70 and $m = 100$ and Gaussian noise with $\sigma = 20$.



(a) 23.19/0.7139 (b) 27.45/0.8463 (c) 24.77/0.8431 (d) 24.59/0.8381 (e) 23.56/0.8177 (f) **29.34/0.8887**

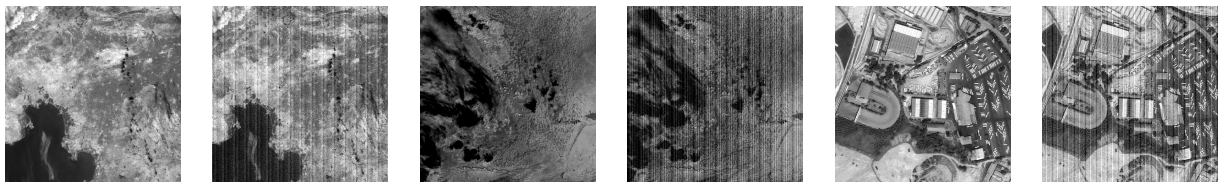
Figure 4. Destriping results of periodic stripes when $(r, m) = (50, 100)$ and $\sigma = 20$. First row: restored $\mathbf{u}$, second row: $\mathbf{u} - \mathbf{u}_*$. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. PSNR/SSIM of $\mathbf{u}$ are presented. Parameter (λ_1, λ_4): $(0.05, 0.4)$.



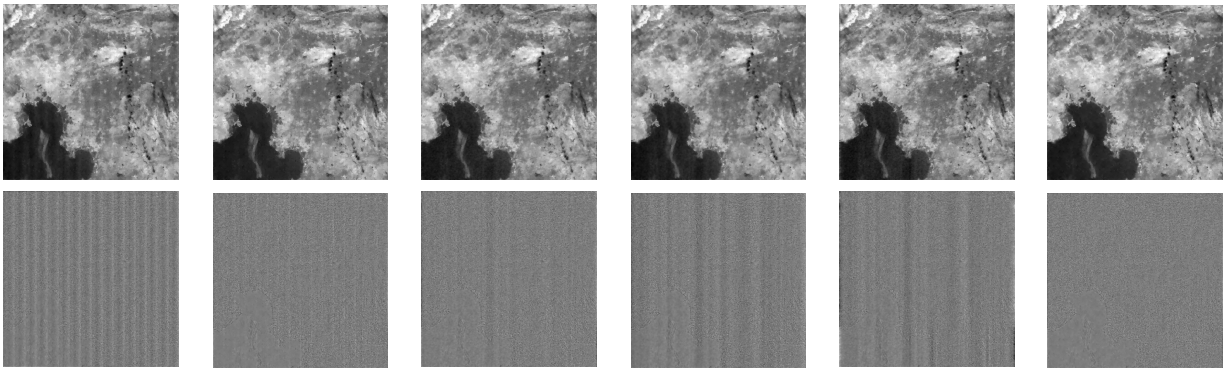
(a) 23.24/0.7240 (b) 26.14/0.8396 (c) 22.96/0.7974 (d) 24.37/0.8354 (e) 23.05/0.8054 (f) **28.79/0.8893**

Figure 5. Destriping results of periodic stripes when $(r, m) = (70, 100)$ and $\sigma = 20$. First row: restored $\mathbf{u}$, second row: $\mathbf{u} - \mathbf{u}_*$. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. PSNR/SSIM of $\mathbf{u}$ are presented. Data $\mathbf{f}$ is given in Figure 3. Parameter (λ_1, λ_4): $(0.05, 0.2)$.

Figures 6 and 7 present the denoising results tested on real SAR images in the presence of periodic stripe noise with $r = 30$, 50, or 70 and $m = 50$ and Gaussian noise with $\sigma = 20$. Similar to the previous results, the LRSID, LRGS+TV, Schatten+TV, and ELRTV+TV models struggle to properly extract stripes from the images compared with our model and TVGS. Also, while TVGS appears to provide similar restored images to our model, our model separates the stripe and image components better than TVGS, which can be seen more clearly in the difference images. Thus, these also show the effective denoising performance of the proposed model for SAR images.

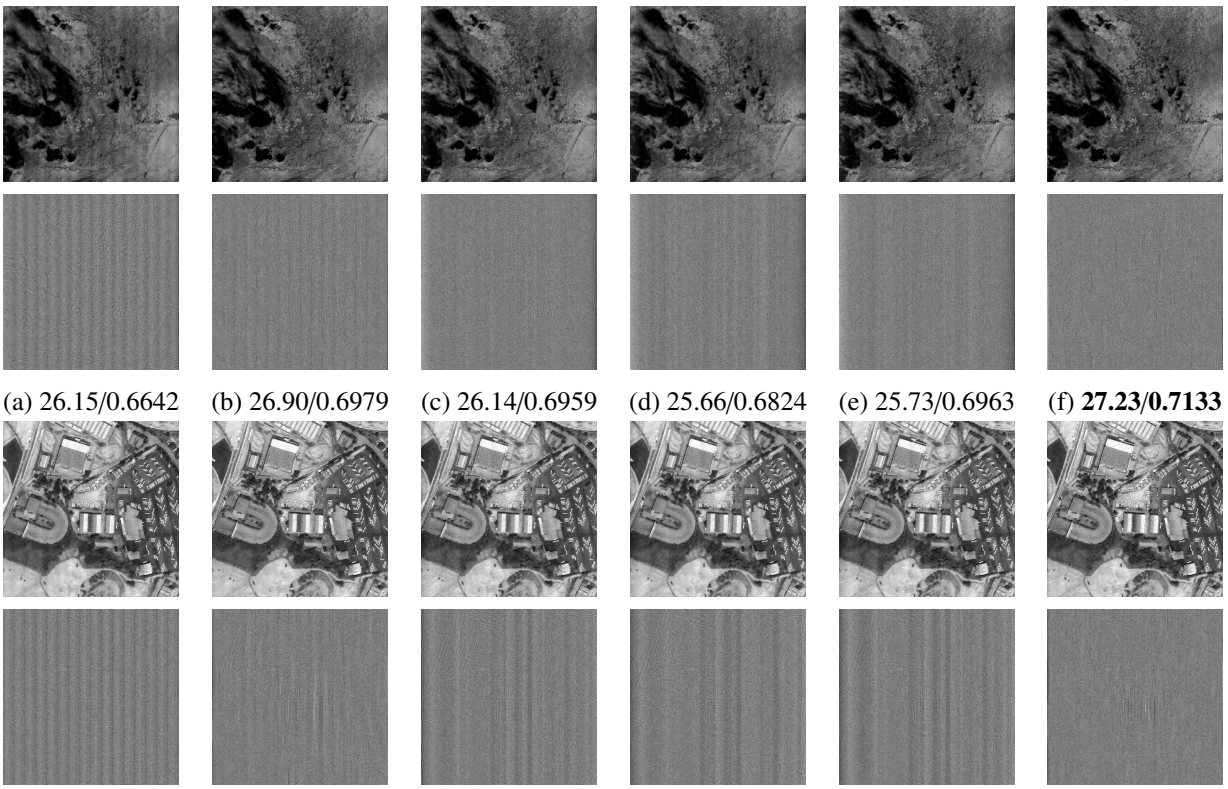


$\mathbf{u}_*$ and $\mathbf{f}$ $(r = 30)$ $\mathbf{u}_*$ and $\mathbf{f}$ $(r = 50)$ $\mathbf{u}_*$ and $\mathbf{f}$ $(r = 70)$



(a) 26.30/0.7053 (b) 26.77/0.7053 (c) 26.76/0.7057 (d) 26.13/0.6872 (e) 26.06/0.7115 (f) **26.95/0.7113**

Figure 6. Destriping results of periodic stripes when $(r, m) = (30, 50)$ and $\sigma = 20$. Second row: restored $\mathbf{u}$, third row: $\mathbf{u} - \mathbf{u}_*$. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. PSNR/SSIM of $\mathbf{u}$ are presented. Parameter (λ_1, λ_4): $(0.05, 0.6)$.



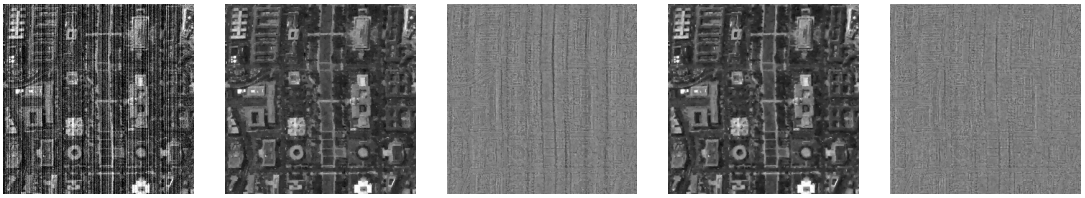
(a) 26.15/0.6642 (b) 26.90/0.6979 (c) 26.14/0.6959 (d) 25.66/0.6824 (e) 25.73/0.6963 (f) **27.23/0.7133**

(a) 25.56/0.7768 (b) 25.83/0.7836 (c) 24.94/0.7795 (d) 24.82/0.7708 (e) 24.72/0.7824 (f) **26.12/0.7936**

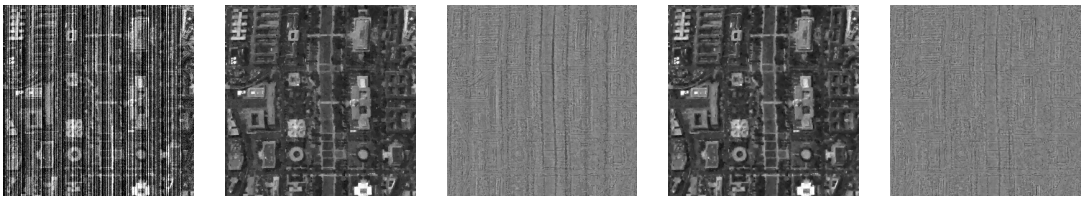
Figure 7. Destriping results of periodic stripes when $r = 50$ (first and second rows), $r = 70$ (third and fourth rows), while $m = 50$ and $\sigma = 20$. First and third rows: restored $\mathbf{u}$, second and fourth rows: $\mathbf{u} - \mathbf{u}_*$. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. PSNR/SSIM of $\mathbf{u}$ are presented. Data $\mathbf{f}$ are given in Figure 6. Parameter (λ_1, λ_4): (top) $(0.04, 0.2)$, (bottom) $(0.04, 0.2)$.

In Figure 8, we present the denoising results of our model with different values of m ($m = 50$ or 100), while $r = 50$ and $\sigma = 20$ are fixed, and we compare our model with TVGS. We can see that our model separates stripes with image structures better than TVGS and that our difference images have much fewer streaks and image edges. On the other hand, despite the use of different intensity values for the stripes, both models supply similar visual quality in the restored images, leading to similar PSNR and SSIM values. Indeed, throughout the experiments, the denoising results of our model are similar when $m = 50$ and $m = 100$, in terms of visual quality of restored images and PSNR and SSIM values.

Table 1 presents the mean PSNR and SSIM values of all methods tested on all images in Figure 2, in the presence of periodic stripe noise and Gaussian noise. The PSNR and SSIM values for all image cases are given in the material at the following link: `https://han.gl/ouTofQ`. As the noise levels σ or r increase, the PSNR and SSIM values decrease. In all cases, the proposed model provides the highest average PSNR and SSIM values. This also verifies the superior denoising performance of our model over other models when both periodic stripes and Gaussian noise exist.



(a) $m = 50$ (b) TVGS: 25.33/0.7734 (c) Proposed: 26.00/0.7947



(a) $m = 100$ (b) TVGS: 25.23/0.7693 (c) Proposed: 25.98/0.7943

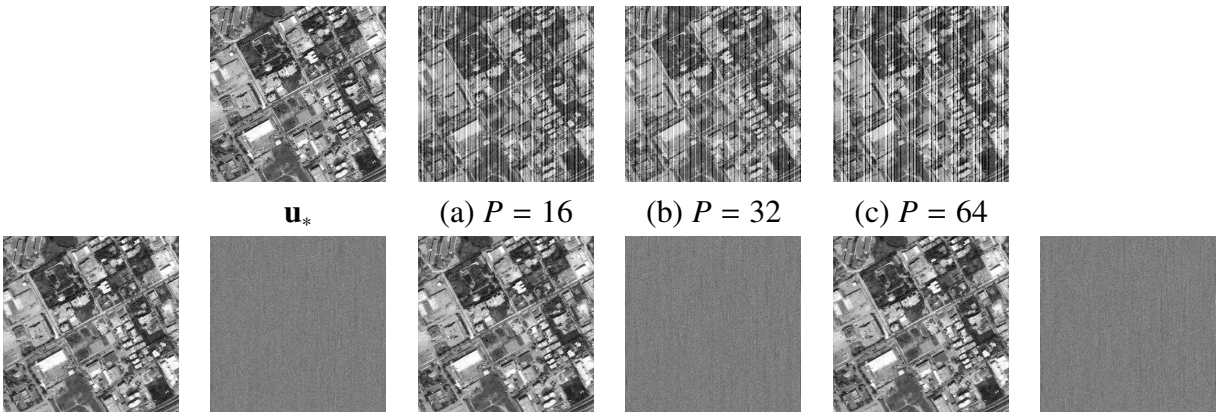
Figure 8. Destriping results of periodic stripes with different $m = 50$ (top), $m = 100$ (bottom), while $r = 50$ and $\sigma = 20$. Second and fourth columns: restored $\mathbf{u}$, third and fifth columns: $\mathbf{u} - \mathbf{u}_*$. PSNR/SSIM of $\mathbf{u}$ are presented. Parameter (λ_1, λ_4): (top) $(0.04, 0.2)$, (bottom) $(0.04, 0.4)$.

Table 1. Average PSNR and SSIM values of all models for periodic stripe noise with (r, m) and Gaussian noise with σ.

	r	30		50		70	
	m	50	100	50	100	50	100
$\sigma = 10$	LRSID	28.20/0.8760	27.27/0.8640	27.86/0.8733	27.35/0.8665	28.02/0.8749	27.41/0.8664
	TVGS	29.06/0.8849	29.02/0.8848	28.78/0.8809	28.74/0.8807	28.59/0.8784	28.51/0.8775
	LRGS	26.94/0.8075	26.93/0.8073	26.86/0.8070	26.84/0.8071	26.72/0.8062	26.70/0.8060
	LRGS+TV	27.09/0.8599	27.09/0.8600	27.01/0.8588	26.98/0.8586	26.85/0.8575	26.83/0.8574
	Shatten	26.69/0.7945	26.31/0.7919	26.43/0.7920	26.59/0.7943	26.56/0.7941	26.44/0.7923
	Shatten+TV	27.51/0.8688	27.04/0.8656	27.09/0.8594	27.37/0.8678	27.34/0.7941	27.11/0.7923
	ELRTV	25.73/0.7918	25.51/0.7896	25.77/0.7923	25.74/0.7921	25.78/0.7923	25.75/0.7921
	ELRTV+TV	25.94/0.8456	25.71/0.8433	25.99/0.8458	25.96/0.8455	25.99/0.8459	25.97/0.8458
	Our	**29.26/0.8878**	**29.26/ 0.8879**	**29.06/0.8847**	**29.07/0.8849**	**28.90/ 0.8825**	**28.87/0.8822**
$\sigma = 20$	LRSID	25.01/0.7595	24.43/0.7480	24.92/0.7595	24.50/0.7525	25.05/0.7614	24.65/0.7552
	TVGS	25.60/0.7783	25.57/0.7784	25.42/0.7742	25.38/0.7732	25.28/0.7702	25.21/0.7695
	LRGS	22.33/0.6149	22.31/0.6147	22.29/0.6148	22.29/0.6146	22.24/0.6145	22.23/0.6139
	LRGS+TV	24.83/0.7625	24.81/0.7630	24.75/0.7619	24.76/0.7619	24.67/0.7607	24.66/0.7608
	Shatten	21.86/0.6022	21.73/0.6001	21.84/0.6024	21.80/0.6010	21.83/0.6016	21.81/0.6015
	Shatten+TV	24.72/0.7564	24.49/0.7547	24.67/0.7566	24.63/0.7566	24.71/0.7570	24.64/0.7566
	ELRTV	21.92/0.6064	21.81/0.6047	21.93/0.6066	21.91/0.6060	21.93/0.6066	21.90/0.6062
	ELRTV+TV	24.29/0.7572	24.11/0.7551	24.30/0.7573	24.27/0.7567	24.30/0.7569	24.27/0.7570
	Our	**25.79/0.7861**	**25.78/0.7864**	**25.63/0.7820**	**25.62/0.7817**	**25.41/0.7766**	**25.45/0.7775**

In Figure 9, we present the denoising results of our model at different periods, such as $P = 16$, 32, and 64. We can observe that our model provides similar denoising performance despite the change of P. In Table 2, we present the PSNR and SSIM values of all models tested on three different images, when $P = 16$, 32, and 64. In the case of LRGS+TV, $P = 32$ provides higher PSNR values than other cases. LRSID and TVGS supply the lowest PSNR values when $P = 64$, whereas Schtten+TV and ELRTV+TV provide the lowest PSNR values when $P = 16$. However, our model provides similar PSNR and SSIM values for different values of P, and our PSNR values are higher than other models

for all cases. This indicates that our model is not sensitive to the value of P in contrast to other models.



$\mathbf{u}_*$ (a) $P = 16$ (b) $P = 32$ (c) $P = 64$

(a) 25.24/0.8095 ($P = 16$) (b) 25.20/0.8076 ($P = 32$) (c) 25.21/0.8092 ($P = 64$)

Figure 9. Destriping results of periodic stripes with different periods $P = 16, 32, 64$, when $(r, m) = (70, 100)$ and $\sigma = 20$. First, third, and fifth columns: restored $\mathbf{u}$. Second, fourth, and sixth columns: $\mathbf{u} - \mathbf{u}_*$. PSNR/SSIM of $\mathbf{u}$ are presented. Parameter (λ_1, λ_4): (a) $(0.04, 0.2)$, (b) $(0.04, 0.4)$, (c) $(0.04, 0.1)$.

Table 2. PSNR and SSIM values of all models for periodic stripe noise with $(r, m) = (70, 100)$ and different periods P and Gaussian noise with $\sigma = 20$.

Image	P	LRSID	TVGS	LRGS+TV	Shatten+TV	ELRTV+TV	Our
(a)	16	24.80/0.6388	26.72/0.6921	25.69/0.6921	24.81/0.6635	24.81/0.6774	**27.15/0.7072**
	32	25.38/0.6660	26.43/0.6919	25.99/0.6966	25.69/0.6843	25.69/0.6984	**27.07/0.7088**
	64	23.54/0.6157	26.32/0.6628	25.81/0.6876	25.60/0.6801	25.57/0.6905	**27.04/0.7066**
(b)	16	25.35/0.6999	26.45/0.6963	26.34/0.7034	25.08/0.6797	25.09/**0.7084**	**26.77**/0.7032
	32	26.23/0.7042	26.41/0.6944	26.64/0.7038	25.83/0.6826	26.04/**0.7109**	**26.70**/0.7028
	64	23.10/0.6851	26.00/0.6926	26.32/**0.7031**	25.95/0.6834	25.76/0.7096	**26.70**/0.7029
(g)	16	23.91/0.7935	25.05/0.8025	24.67/0.8044	24.04/0.7968	24.09/0.8066	**25.24/0.8095**
	32	24.00/0.7805	25.01/0.8014	24.93/0.7780	24.69/0.7972	24.81/0.8069	**25.20/0.8076**
	64	21.40/0.7714	24.92/0.8016	24.85/0.8045	24.70/0.7982	24.77/0.8082	**25.21/0.8092**

Lastly, Figure 10 depicts the impact of parameters λ_1, α, λ_3, and λ_4. As mentioned earlier, we set the same values for λ_1 and λ_2. First, λ_1 and α control the smoothness of the recovered images. Specifically, as the value of λ_1 increases, the restored image becomes smoother. Although $\lambda_1 = 0.03$ provides the highest PSNR value, the restored image with $\lambda_1 = 0.03$ retains some Gaussian noise. Thus, we choose the restored image with $\lambda_1 = 0.04$ as the best image. In the whole test, we selected the best restored images, considering both their visual quality and PSNR and SSIM values. Second, as the value of α increases, the staircase effect, which occurs when $\alpha = 1$, becomes alleviated, but the restored images becomes smoother, which also leads to a loss of details. In this case, we choose the restored image with $\alpha = 1.3$ as the best image since it provides the highest PSNR value. Throughout the experiments, we select $\alpha = 1.3$ or 1.5. Finally, λ_3 and λ_4 control the separation of stripes from the image. For λ_3, we test four different values, such as $0.05, 0.1, 0.6, 1$. If the value of λ_3 is too small, the stripes are not extracted properly and the restored image is over-smoothed. But a large enough value of λ_3 enables a successful extraction of stripes. Indeed, for $\lambda_3 = 0.6$ or higher, the denoising results do not change

much, so we fix the value of λ_3 to 0.6 throughout the experiment. For λ_4, we test four values, such as 0.01, 0.05, 0.2, 0.4. It can be observed that using $\lambda_1 = 0.01$ or 0.4 fails to properly extract stripes from the image, while using $\lambda_4 = 0.05$ or 0.2 provides better decomposition of the stripe and image components than the others. Besides, $\lambda_4 = 0.05$ and 0.2 provide very similar PSNR values. Although the parameter λ_4 is more sensitive than the other parameters, λ_4 is selected from {0.05, 0.08, 0.2, 0.4} in many cases.

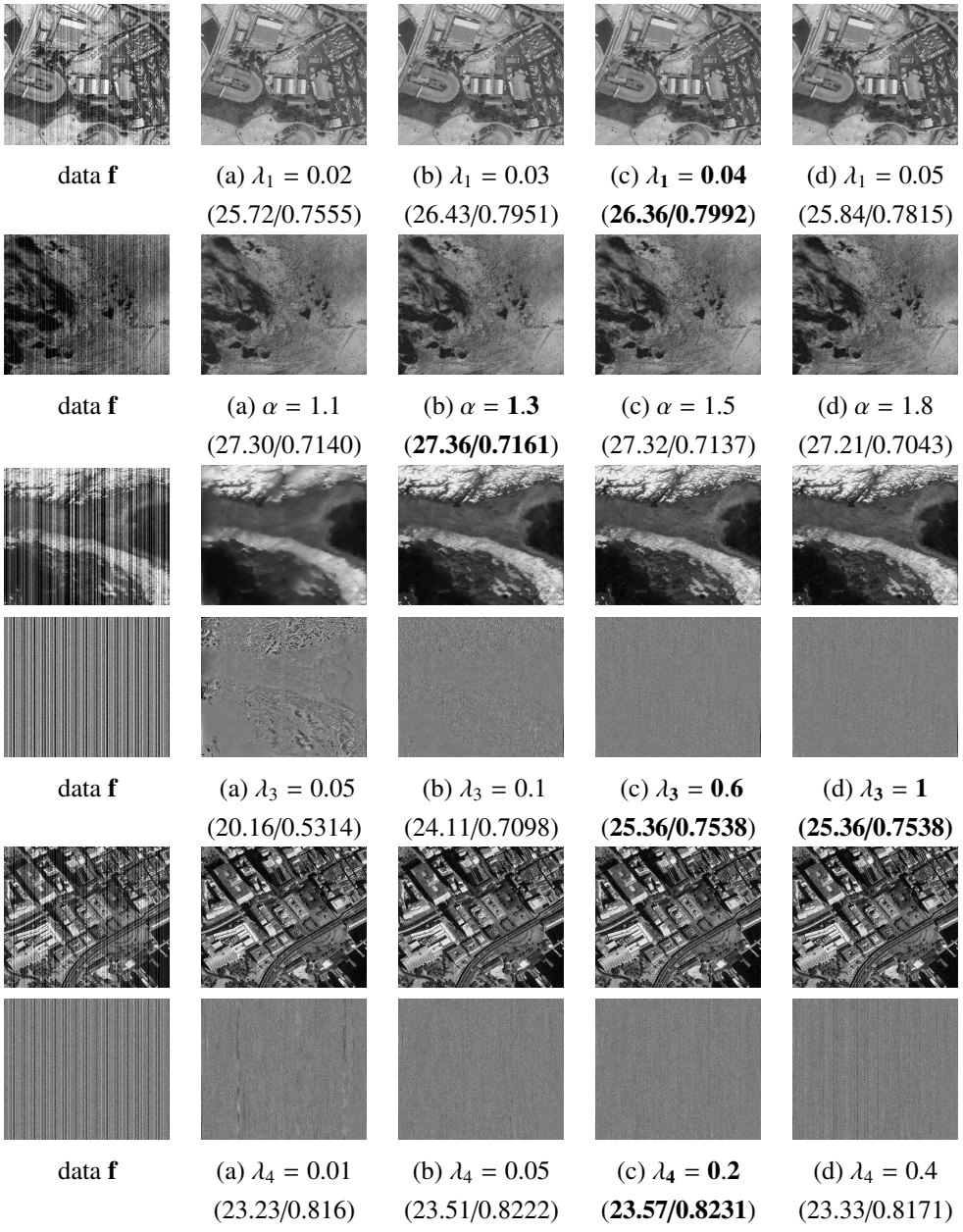


Figure 10. Effect of parameters λ_1, α, λ_3, and λ_4 in the proposed model in the presence of periodic stripes and Gaussian noise. First and second rows: $(r, m) = (30, 50)$ and $\sigma = 20$, third and fourth rows: $(r, m) = (70, 100)$ and $\sigma = 20$, fifth and sixth rows : $(r, m) = (70, 50)$ and $\sigma = 20$. PSNR/SSIM of **u** are presented. Parameter: (top to bottom) $\lambda_4 = 0.4$, $(\lambda_1, \lambda_4) = (0.04, 0.2)$, $(\lambda_1, \lambda_4) = (0.05, 0.4)$, $\lambda_1 = 0.4$.

4.3. Denoising results with non-periodic stripes and Gaussian noise

This section presents the denoising results in the presence of non-periodic stripe noise and Gaussian noise.

First, Figure 11 presents the denoising results tested on the vatican image in the presence of non-periodic stripes with $r = 50$ or 70, $m = 100$, and Gaussian noise with $\sigma = 10$. The noisy data images are provided in the first row. We can see that all the models except our model fail to properly separate the stripes from the image, which leads to some traces of stripes in the restored images. This is also visible in the difference images between $\mathbf{u}$ and $\mathbf{u}_*$. For all cases, our model effectively eliminates both stripes and Gaussian noise, resulting in the highest PSNR and SSIM values. These show the efficiency of our directional term and nonconvex group sparsity term of stripes to extract non-periodic stripes in the presence of Gaussian noise.

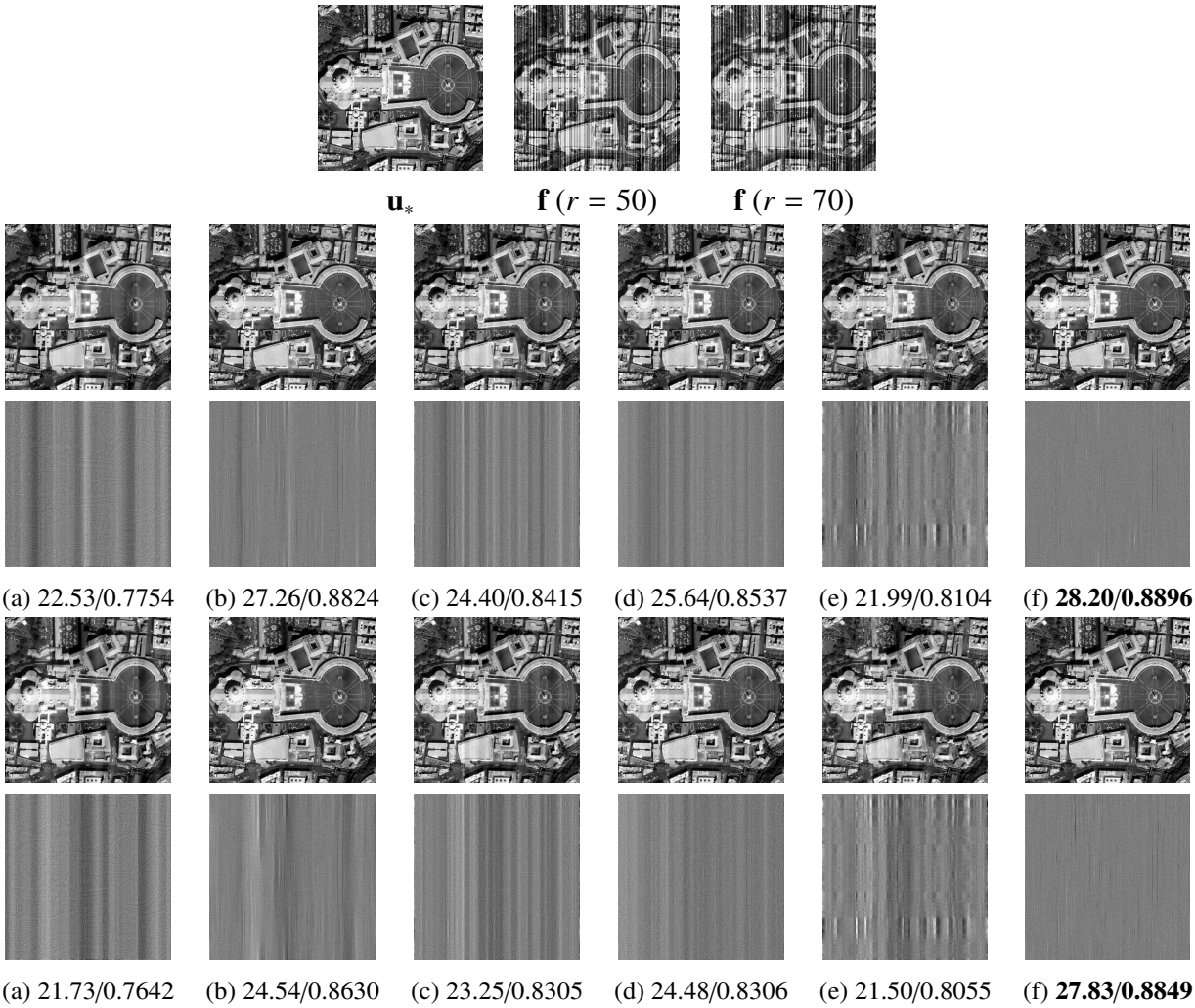


Figure 11. Destriping results of non-periodic stripes when $r = 50$ (second and third rows), $r = 70$ (fourth and fifth rows), while $m = 100$ and $\sigma = 10$. Second and fourth rows: restored $\mathbf{u}$, third and fifth rows: $\mathbf{u} - \mathbf{u}_*$. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. PSNR/SSIM of $\mathbf{u}$ are presented. Parameter (λ_1, λ_4): (top) $(0.02, 0.1)$, (bottom) $(0.02, 0.05)$.

In Figure 12, we present the denoising results tested on two other different images in the presence of non-periodic stripes with $r = 50$ or 70, $m = 50$, and Gaussian noise with $\sigma = 20$. Similarly, TVGS eliminates stripes from the images better than LRSID, LRGS+TV, Schatten+TV, and ELRTV+TV, bringing better denoised images. But there are traces of steaks in both the restored and difference images of TVGS. In contrast, our model removes both stripe noise and Gaussian noise sufficiently, yielding cleaner restored images than other models. Furthermore, our model mitigates the staircase artifacts that appeared in the restored images of TVGS. Therefore, these examples also confirm the effectiveness of the proposed model for removing both non-periodic stripes and Gaussian noise.

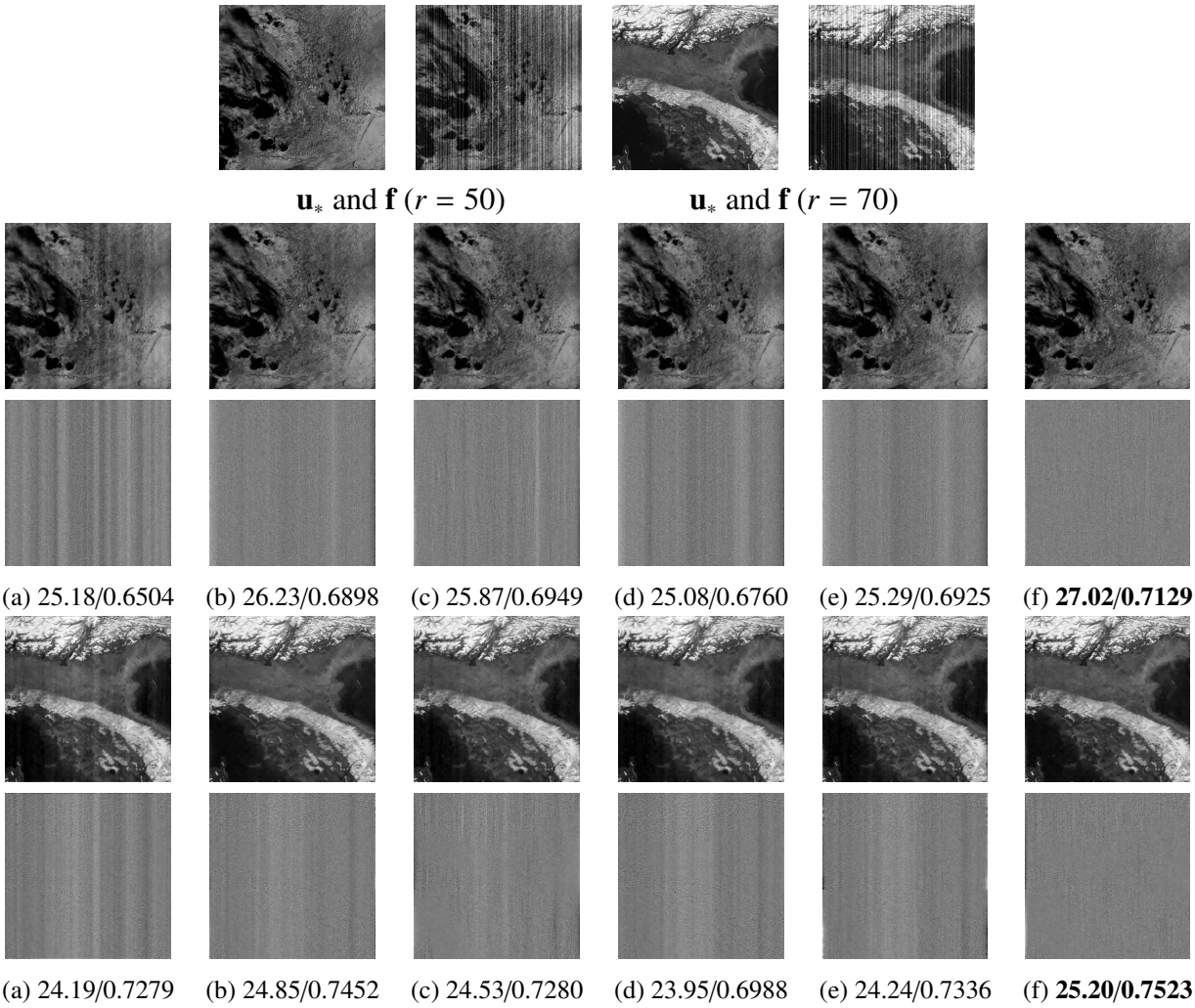
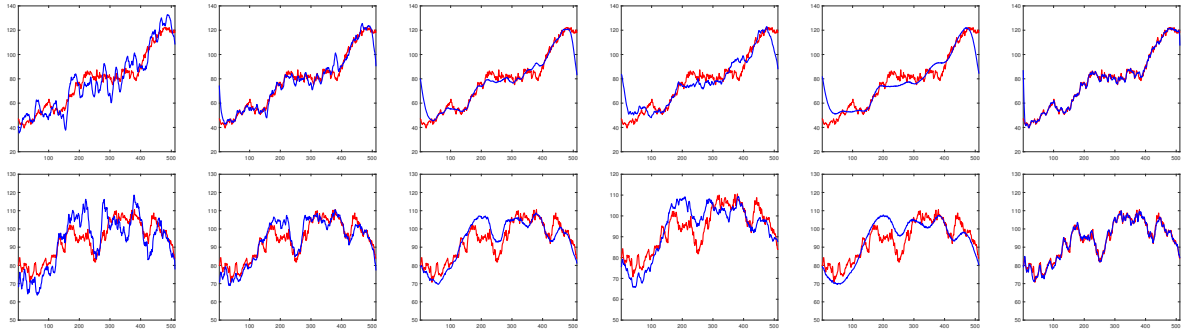


Figure 12. Destriping results of non-periodic stripes when $r = 50$ (second and third rows), $r = 70$ (fourth and fifth rows), while $m = 50$ and $\sigma = 20$. Second and fourth rows: restored $\mathbf{u}$, third and fifth rows: $\mathbf{u} - \mathbf{u}_*$. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. PSNR/SSIM of $\mathbf{u}$ are presented. Parameter (λ_1, λ_4): (top) $(0.04, 0.1)$, (bottom) $(0.05, 0.1)$.

Figure 13 shows the column mean cross-track profiles of the restored images (blue curve) in Figure 12 and original images (red curve). The horizontal axis represents the column number, and

the vertical axis represents the mean value of the intensities in each column. It can be seen that the curves of our model are similar to the original ones. Meanwhile, there are large gaps between the curves of the other models and the original ones. Hence, these examples also show better denoising performance of our model than the others.



(a) (b) (c) (d) (e) (f)

Figure 13. Column mean cross-track profiles of Figure 12. (top) MODIS-BAND20 image, (bottom) image02172021 image. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. Red curves: original images, blue curves: restored images.

In Table 3, we record the average of the PSNR and SSIM values of all methods tested on all images in Figure 2, in the presence of non-periodic stripes and Gaussin noise. We can see that the proposed model supplies the highest PSNR and SSIM values for all cases. The quantitative assessment is also consistent with visual results. This illustrates that the proposed model is superior to the existing models in terms of visual quality and image quality evaluation.

Table 3. Average PSNR and SSIM values of all models for non-periodic stripe noise with (r, m) and Gaussian noise with σ.

	r	30		50		70	
	m	50	100	50	100	50	100
σ = 10	LRSID	25.17/0.7666	23.60/0.7457	24.76/0.7612	22.86/0.7367	24.38/0.7590	21.94/0.7245
	TVGS	28.93/0.8831	28.87/0.8827	28.45/0.8782	28.23/0.8764	27.50/0.8704	25.96/0.8613
	LRGS	26.93/0.8077	26.74/0.8065	26.63/0.8058	26.08/0.8005	25.95/0.8010	24.51/0.7914
	LRGS+TV	27.08/0.8598	26.89/0.8586	26.78/0.8578	26.21/0.8490	26.07/0.8514	24.59/0.8371
	Shatten	26.51/0.7940	25.97/0.7919	26.02/0.7904	25.57/0.7824	26.13/0.7911	24.14/0.7403
	Shatten+TV	27.28/0.8674	26.69/0.8643	26.63/0.8576	26.33/0.8561	26.83/0.8634	25.06/0.8218
	ELRTV	25.59/0.7918	24.90/0.8375	25.34/0.7902	24.55/0.7822	25.11/0.7880	21.73/0.7775
	ELRTV+TV	25.79/0.8450	25.08/0.8375	25.54/0.8433	24.69/0.8305	25.29/0.8405	23.83/0.8245
	Our	**29.19/0.8867**	**29.19/0.8869**	**29.01/0.8843**	**28.89/0.8828**	**28.40/0.8788**	**28.54/0.8798**
σ = 20	LRSID	24.57/0.7557	23.11/0.7343	24.13/0.7498	22.49/0.7246	24.05/0.7495	21.52/0.7146
	TVGS	25.53/0.7770	25.45/0.7763	25.05/0.7692	25.06/0.7683	24.58/0.7613	24.15/0.7484
	LRGS	22.33/0.6149	22.31/0.6147	22.29/0.6148	22.29/0.6146	22.24/0.6145	22.23/0.6139
	LRGS+TV	24.83/0.7625	24.81/0.7630	24.75/0.7619	24.76/0.7619	24.67/0.7607	24.66/0.7608
	Shatten	21.73/0.6000	21.73/0.6001	21.41/0.5992	21.11/0.5885	21.58/0.5994	20.61/0.5624
	Shatten+TV	24.52/0.7545	24.49/0.7547	24.01/0.7525	23.78/0.7453	24.24/0.7528	23.55/0.7350
	ELRTV	21.92/0.6064	21.81/0.6047	21.93/0.6066	21.91/0.6060	21.93/0.6066	21.90/0.6062
	ELRTV+TV	24.29/0.7572	24.11/0.7551	24.30/0.7573	24.27/0.7567	24.30/0.7569	24.27/0.7570
	Our	**25.75/0.7851**	**25.74/0.7851**	**25.29/0.7771**	**25.46/0.7794**	**25.05/0.7717**	**25.08/0.7633**

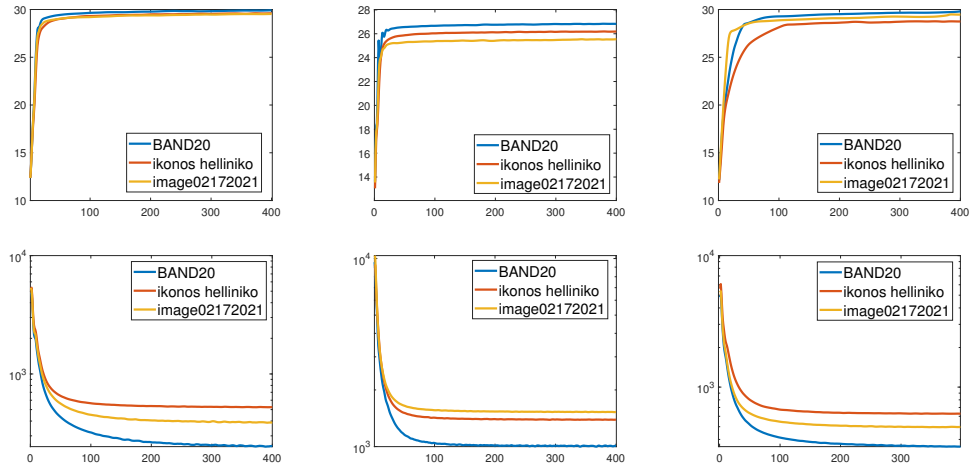
Table 4 presents the computing time of all models, in the case of the non-periodic stripe noise

with $(r, m) = (70, 50)$ and Gaussian noise with $\sigma = 20$. It can be observed that LRSID, TVGS, LRGS(+TV), and ELRTV(+TV) models are faster than Schatten(+TV) and our model. Despite the high computational cost, the proposed model provides better restoration results than other models.

Table 4. Computing time (in seconds) of all models for non-periodic stripe noise with $(r, m) = (70, 100)$ and Gaussian noise $\sigma = 20$.

Image	LRSID	TVGS	LRGS (+TV)	Shatten (+TV)	ELRTV (+TV)	Our
MODISBAND20	17.36	20.24	24.37 (25.06)	24.62 (25.47)	17.51 (18.31)	26.96
BAND20	15.54	19.34	20.40 (20.94)	28.48 (28.95)	17.97 (18.59)	26.95
Original band30	7.17	7.05	10.11 (10.40)	10.54 (10.82)	8.78 (9.15)	9.54
rio	15.36	19.25	21.12 (21.67)	29.06 (29.51)	17.35 (17.98)	26.79
rio1	15.47	17.20	25.48 (26.18)	30.18 (30.88)	19.55 (20.18)	28.14
helliniko	16.17	19.45	22.15 (22.68)	32.24 (32.87)	17.98 (18.56)	25.83
helliniko1	16.82	19.29	20.70 (21.26)	23.74 (24.17)	17.60 (18.17)	26.54
vatican	16.64	20.21	19.69 (20.21)	29.50 (30.14)	18.14 (18.72)	27.13
image02162021	19.38	23.84	25.81 (26.73)	32.28 (33.29)	20.18 (21.26)	32.89
image02172021	16.85	17.56	17.11 (17.71)	30.54 (31.25)	16.85 (17.55)	26.65

Figure 14 presents the plots of the PSNR and energy functional values of our model via the outer iteration number. As the outer iteration number increases, we can see that the PSNR values gradually increase and converge to some constant values, while the energy values gradually decrease. These plots justify the numerical convergence of the proposed algorithm.



(a) (b) (c)

Figure 14. Plots of PSNR and energy functional values of the proposed model via outer iteration number ℓ. (top) PSNR; (bottom) Energy functional value. (a) periodic stripes with $(r, m) = (50, 50)$ and $\sigma = 10$, (b) periodic stripes with $(r, m) = (50, 50)$ and $\sigma = 20$, (c) non-periodic stripes with $(r, m) = (50, 50)$ and $\sigma = 10$.

In Figure 15, we present the extracted stripe noise components, $\mathbf{s}$, of all models from Figures 6, 7, 11, and 12. We also record the PSNR and SSIM values between the extracted stripe noise and the originally added stripe noise. It can be observed that the extracted stripes of our model are very close to the originally added stripes, which contributes the highest PSNR and SSIM values of our stripe noise

component. This confirms the effectiveness of our model for extracting stripe noise in the presence of Gaussian noise.

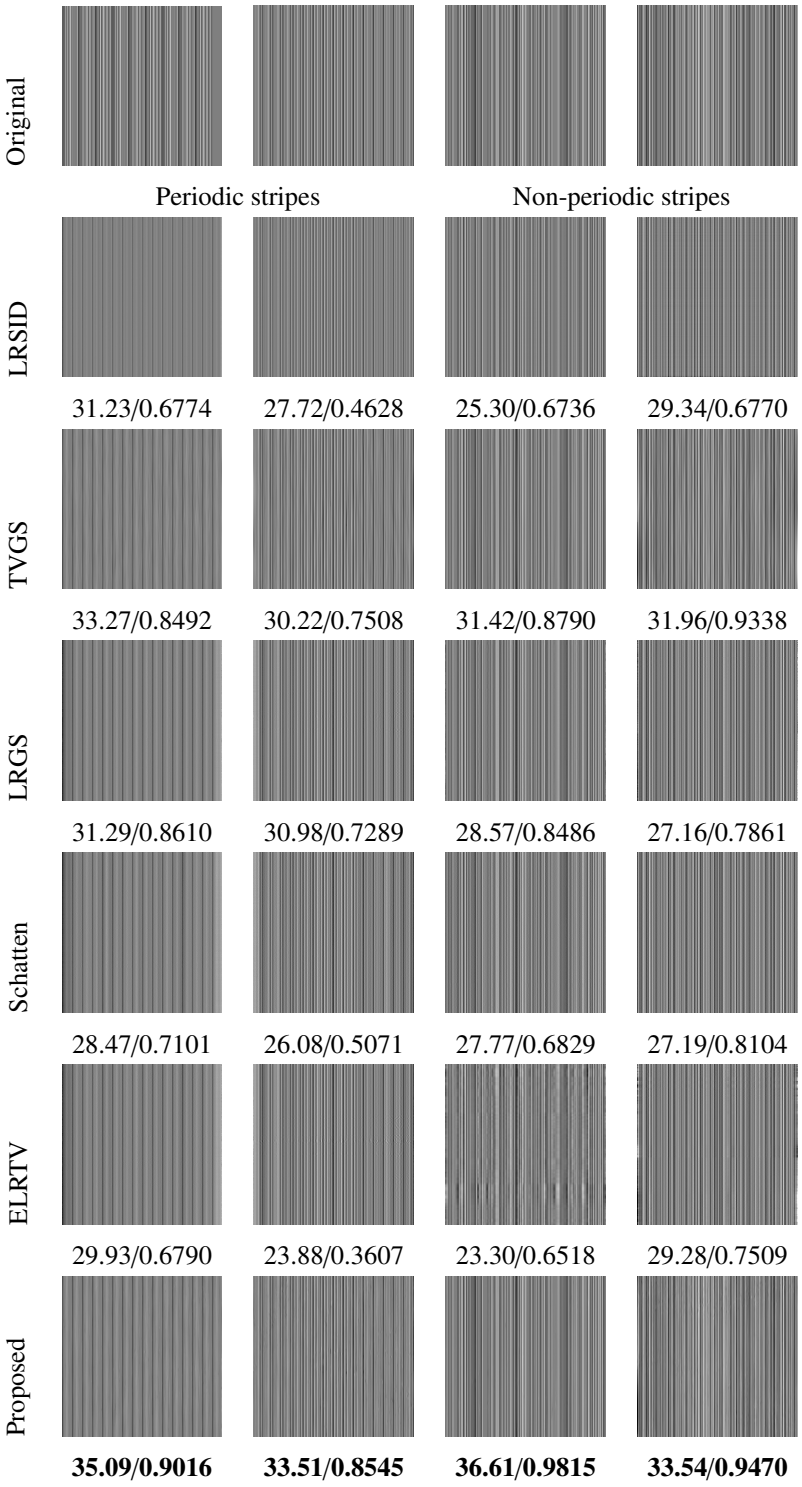
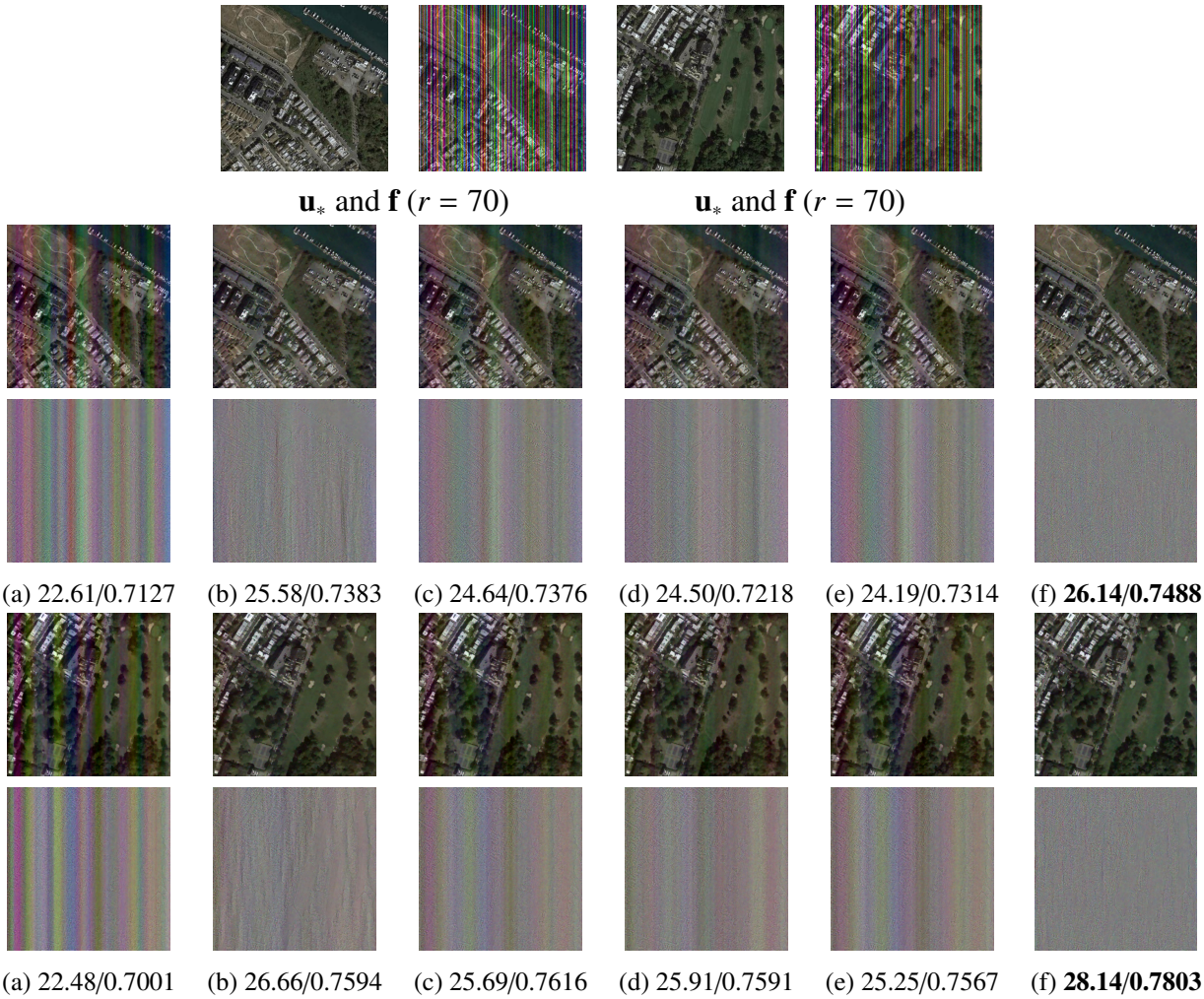


Figure 15. Extracted stripe component **s** of all models. (First column) MODIS-BAND20 in Figure 6, (second column) ikonos helliniko in Figure 7, (third column) vatican ($r = 50$) in Figure 11, (fourth column) image02172021 in Figure 12. PSNR/SSIM of **s** are presented.

Finally, in Figure 16, we present the denoising results for color SAR images in the presence of non-periodic stripes with $(r, m) = (70, 100)$ and Gaussian noise with $\sigma = 20$. Stripes and Gaussian noise are added to each color channel independently, and all models are applied to each channel. We can see that our model not only appropriately separates the stripes from the images but also preserves edges and details well compared with other models. These examples also validate the effectiveness of the proposed model on color SAR images with a mixture of stripes and Gaussian noise. More experimental results on color SAR images can be found in the material at the following link: `https://url.kr/kdgrcz`.



(a) 22.61/0.7127 (b) 25.58/0.7383 (c) 24.64/0.7376 (d) 24.50/0.7218 (e) 24.19/0.7314 (f) **26.14/0.7488**

(a) 22.48/0.7001 (b) 26.66/0.7594 (c) 25.69/0.7616 (d) 25.91/0.7591 (e) 25.25/0.7567 (f) **28.14/0.7803**

Figure 16. Destriping results of non-periodic stripes when $(r, m) = (70, 100)$ and $\sigma = 20$, tested on color SAR images. Second and fourth rows: restored $\mathbf{u}$, third and fifth rows: $\mathbf{u}-\mathbf{u}_*$. (a) LRSID, (b) TVGS, (c) LRGS+TV, (d) Schatten+TV, (e) ELRTV+TV, (f) Proposed. PSNR/SSIM of $\mathbf{u}$ are presented. Parameter (λ_1, λ_4): (top) $(0.05, 0.2)$, (bottom) $(0.05, 0.1)$.

5. Conclusions

In this paper, we introduce an image decomposition model to extract stripe noise and image components from a noisy image that is corrupted by a mixture of stripe noise and Gaussian noise. We

considered various types of periodic or non-periodic stripe noise and also assumed a relatively high level of Gaussian noise unlike the previous works. For the stripe noise component, a unidirectional TV and a nonconvex group sparsity term were exploited, and they enabled the proper separation of periodic or non-periodic stripes from images with high levels of Gaussian noise. Furthermore, for the image component, we made use of a nonconvex FTV regularization, which not only ameliorated the staircase effect appearing in images recovered from TV-based models but also enabled the conservation of edges and details. To handle the nonconvex and nonsmooth problem, we adopted IRL1 and ADMM. This led to an efficient iterative algorithm capable of satisfactorily solving the proposed model, and we also proved its global convergence. The numerical results validated that the proposed model generated superior denoising results than other existing models in terms of visual and image quality assessments. Despite the effective performance of the proposed model, issues remain regarding high computational time and many parameters, which need to be investigated in future work.

Author contributions

All authors developed the theoretical formalism, performed the analytic calculations and performed the numerical simulations. All authors contributed to the final version of the manuscript.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

Myeongmin Kang was supported by the National Research Foundation (NRF) of Korea grant (No. 2019R1I1A3A01055168). Miyoun Jung was supported by the Hankuk University of Foreign Studies Research Fund and the National Research Foundation (NRF) of Korea grant (No. RS-2023-00241770).

Conflict of interest

The authors declare no conflicts of interest in this paper.

References

1. J. A. Richards, *Remote sensing digital image analysis: an introduction*, Springer Berlin, Heidelberg, 2013, 343–380. https://doi.org/10.1007/978-3-642-30062-2
2. I. Makki, R. Younes, C. Francis, T. Bianchi, M. Zucchetti, A survey of landmine detection using hyperspectral imaging, *ISPRS J. Photogramm. Remote Sens.*, **124** (2017), 40–53. https://doi.org/10.1016/j.isprsjprs.2016.12.009
3. H. Zhang, J. Li, Y. Huang, L. Zhang, A nonlocal weighted joint sparse representation classification method for hyperspectral imagery, *IEEE J-STARS*, **7** (2017), 2056–2065. https://doi.org/10.1109/JSTARS.2013.2264720

4. Y. Tarabalka, J. Chanussot, J. A. Benediktssons, Segmentation and classification of hyperspectral images using watershed transformation, *Pattern Recogn.*, **43** (2010), 2367–2379. https://doi.org/10.1016/j.patcog.2010.01.016
5. D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, A. D. Stocker, Anomaly detection from hyperspectral imagery, *IEEE Signal Proc. Mag.*, **19** (2002), 58–69. https://doi.org/10.1109/79.974730
6. M. D. Iordache, J. M. Bioucas-Dias, A. Plaza, Collaborative sparse regression for hyperspectral unmixing, *IEEE Trans. Geosci. Remote Sens.*, **52** (2014), 341–354. https://doi.org/10.1109/TGRS.2013.2240001
7. J. Chen, Y. Shao, H. Guo, W. Wang, B. Zhu, Destriping CMODIS data by power filtering, *IEEE Trans. Geosci. Remote Sens.*, **41** (2003), 2119–2124. https://doi.org/10.1109/TGRS.2003.817206
8. J. Chen, H. Lin, Y. Shao, L. Yang, Oblique striping removal in remote sensing imagery based on wavelet transform, *Int. J. Remote Sens.*, **27** (2006), 1717–1723. https://doi.org/10.1080/01431160500185516
9. R. Pande-Chhetri, A. Abd-Elrahman, De-striping hyperspectral imagery using wavelet transform and adaptive frequency domain filtering, *ISPRS J. Photogramm. Remote Sens.*, **66** (2011), 620–636. https://doi.org/10.1016/j.isprsjprs.2011.04.003
10. L. Sun, R. Neville, K. Staenz, H. P. White, Automatic destriping of Hyperion imagery based on spectral moment matching, *J. Can. Remote Sens.*, **34** (2008), S68–S81. https://doi.org/10.5589/m07-067
11. M. Wegener, Destriping multiple sensor imagery by improved histogram matching, *Int. J. Remote Sens.*, **11** (1990), 859–875. https://doi.org/10.1080/01431169008955060
12. H. Shen, L. Zhang, A MAP-based algorithm for destriping and inpainting of remotely sensed images, *IEEE Trans. Geosci. Remote Sens.*, **47** (2009), 1492–1502. https://doi.org/10.1109/TGRS.2008.2005780
13. M. Bouali, S. Ladjal, Toward optimal destriping of MODIS data using a unidirectional variational model, *IEEE Trans. Geosci. Remote Sens.*, **49** (2011), 2924–2935. https://doi.org/10.1109/TGRS.2011.2119399
14. Y. Chang, H. Fang, L. Yan, H. Liu, Robust destriping method with unidirectional total variation and framelet regularization, *Opt. Express*, **21** (2013), 23307–23323. https://doi.org/10.1364/OE.21.023307
15. Y. Chang, L. Yan, H. Fang, H. Liu, Simultaneous destriping and denoising for remote sensing images with unidirectional total variation and sparse representation, *IEEE Geosci. Remote Sens. Lett.*, **11** (2014), 1051–1055. https://doi.org/10.1109/LGRS.2013.2285124
16. Y. Zhang, G. Zhou, L. Yan, T. Zhang, A destriping algorithm based on TV-Stokes and unidirectional total variation model, *Optik*, **127** (2016), 428–439. https://doi.org/10.1016/j.ijleo.2015.09.246
17. M. Wang, X. Zheng, J. Pan, B. Wang, Unidirectional total variation destriping using difference curvature in MODIS emissive bands, *Infrared Phys. Technol.*, **75** (2016), 1–11. https://doi.org/10.1016/j.infrared.2015.12.004
18. X. Liu, X. Lu, H. Shen, Q. Yuan, Y. Jiao, L. Zhang, Stripe noise separation and removal in remote sensing images by consideration of the global sparsity and local variational properties, *IEEE Trans. Geosci. Remote Sens.*, **54** (2016), 3049-3060. https://doi.org/10.1109/TGRS.2015.2510418

19. Y. Chang, L. Yan, T. Wu, S. Zhong, Remote sensing image stripe noise removal: from image decomposition perspective, *IEEE Trans. Geosci. Remote Sens.*, **54** (2016), 7018–7031. https://doi.org/10.1109/TGRS.2016.2594080
20. Y. Chen, T. Z. Huang, X. Zhao, L. J. Deng, J. Huang, Stripe noise removal of remote sensing images by total variation regularization and group sparsity constraint, *Remote sens.*, **9** (2017), 559. https://doi.org/10.3390/rs9060559
21. Y. Chen, T. Z. Huang, L. J. Deng, X. L. Zhao, M. Wang, Group sparsity based regularization model for remote sensing image stripe noise removal, *Neurocomputing*, **267** (2017), 95–106. https://doi.org/10.1016/j.neucom.2017.05.018
22. Y. Chen, T. Z. Huang, X. L. Zhao, Destriping of multispectral remote sensing image using low-rank tensor decomposition, *IEEE J-STARS*, **11** (2018), 4950–4967. https://doi.org/10.1109/JSTARS.2018.2877722
23. H. X. Dou, T. Z. Huang, L. J. Deng, X. L. Zhao, J. Huang, Directional ℓ_0 sparse modeling for image stripe noise removal, *Remote Sens.*, **10** (2018), 361. https://doi.org/10.3390/rs10030361
24. S. Qiong, Y. Wang, X. Yan, H. Gu, Remote sensing images stripe noise removal by double sparse regulation and region separation, *Remote Sens.*, **10** (2018), 998. https://doi.org/10.3390/rs10070998
25. J. Wang, T. Z. Huang, T. H. Ma, X. L. Zhao, Y. Chen, A sheared low-rank model for oblique stripe removal, *Appl. Math. Comput.*, **360** (2019), 167–180. https://doi.org/10.1016/j.amc.2019.03.066
26. J. L. Wang, T. Z. Huang, X. L. Zhao, J. Huang, T. H. Ma, Y. B. Zheng, Reweighted block sparsity regularization for remote sensing images destriping, *IEEE J-STARS*, **12** (2019), 4951–4963. https://doi.org/10.1109/JSTARS.2019.2940065
27. J. H. Yang, X. L. Zhao, T. H. Ma, Y. Chen, T. Z. Huang, M. Ding, Remote sensing images destriping using unidirectional hybrid total variation and nonconvex low-rank regularization, *J. Comput. Appl. Math.*, **363** (2020), 124–144. https://doi.org/10.1016/j.cam.2019.06.004
28. X. Wu, H. Qu, L. Zheng, T. Gao, A remote sensing image destriping model based on low-rank and directional sparse constraint, *Remote Sens.*, **13** (2021), 5126. https://doi.org/10.3390/rs13245126
29. X. Liu, X. Lu, H. Shen, Q. Yuan, L. Zhang, Oblique stripe removal in remote sensing images via oriented variation, *arXiv*, 2018. https://doi.org/10.48550/arXiv.1809.02043
30. Q. Song, Z. Huang, H. Ni, K. Bai, Z. Li, Remote sensing images destriping with an enhanced low-rank prior and total variation regulation, *Signal Image Video Process.*, **16** (2022), 1895–1903. https://doi.org/10.1007/s11760-022-02149-8
31. L. Song, H. Huang, Simultaneous destriping and image denoising using a nonparametric model with the EM algorithm, *IEEE Trans. Image Process.*, **32** (2023), 1065–1077. https://doi.org/10.1109/TIP.2023.3239193
32. N. Kim, S. S. Han, C. S. Jeong, ADOM: ADMM-Based optimization model for stripe noise removal in remote sensing image, *IEEE Access*, **11** (2023), 106587–106606. https://doi.org/10.1109/ACCESS.2023.3319268
33. F. Yan, S. Wu, Q. Zhang, Y. Liu, H. Sun, Destriping of remote sensing images by an optimized variational model, *Sensors*, **23** (2023), 7529. https://doi.org/10.3390/s23177529
34. C. Wang, X. Zhao, Q. Wang, Z. Ma, P. Tang, An inexact proximal majorization-minimization algorithm for remote sensing image stripe noise removal, *Numer. Algor.*, 2024. https://doi.org/10.1007/s11075-023-01743-2

35. L. I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithm, *Phys. D*, **60** (1992), 259–268. https://doi.org/10.1016/0167-2789(92)90242-F
36. T. Chan, A. Marquina, P. Mulet, High-order total variation-based image restoration, *SIAM J. Sci. Comput.*, **22** (2000), 503–516. https://doi.org/10.1137/S1064827598344169
37. M. Lysaker, A. Lundervold, X. C. Tai, Noise removal using fourth-order partial differential equation with application to medical magnetic resonance images in space and time, *IEEE Trans. Image Process.*, **12** (2003), 1579–1590. https://doi.org/10.1109/TIP.2003.819229
38. K. Bredies, K. Kunisch, T. Pock, Total generalized variation, *SIAM J. Imaging Sci.*, **3** (2010), 492–526. https://doi.org/10.1137/090769521
39. F. Li, C. Shen, J. Fan, C. Shen, Image restoration combining a total variational filter and a fourth-order filter, *J. Vis. Commun. Image Represent.*, **18** (2007), 322–330. https://doi.org/10.1016/j.jvcir.2007.04.005
40. K. Papafitsoros, C. B. Schönlieb, A combined first and second order variational approach for image restoration, *J. Math. Imaging Vis.*, **48** (2014), 308–338. https://doi.org/10.1007/s10851-013-0445-4
41. J. Bai, X. C. Feng, Fractional-order anisotropic diffusion for image denoising, *IEEE Trans. Image Process.*, **16** (2007), 2492–2502. https://doi.org/10.1109/TIP.2007.904971
42. J. Zhang, Z. Wei, L. Xiao, Adaptive fractional-order multi-scale method for image denoising, *SIAM J. Imaging Sci.*, **43** (2012), 39–49. https://doi.org/10.1007/s10851-011-0285-z
43. R. H. Chan, A. Lanza, S. Morigi, F. Sgallari, An adaptive strategy for the restoration of textured images using fractional order regularization, *Numer. Math. Theor. Meth. Appl.*, **6** (2013), 276–296. https://doi.org/10.4208/nmtma.2013.mssvm15
44. J. Zhang, Z. Hui, L. Xiao, A fast adaptive reweighted residual-feedback iterative algorithm for fractional order total variation regularized multiplicative noise removal of partly-textured images, *Signal Process.*, **98** (2014), 381–395. https://doi.org/10.1016/j.sigpro.2013.12.009
45. J. Zhang, K. Chen, A total fractional-order variation model for image restoration with nonhomogeneous boundary conditions and its numerical solution, *SIAM J. Imaging Sci.*, **8** (2015), 2487–2518. https://doi.org/10.1137/14097121X
46. A. Ullah, W. Chen, M. A. Khan, A new variational approach for restoring images with multiplicative noise, *Comput. Math. Appl.*, **71** (2016), 2034–2050. https://doi.org/10.1016/j.camwa.2016.03.024
47. F. Dong, Y. Chen, A fractional-order derivative based variational framework for image denoising, *Inverse Probl. Imag.*, **10** (2016), 27–50. https://doi.org/10.3934/ipi.2016.10.27
48. M. R. Chowdhury, J. Zhang, J. Qin, Y. Lou, Poisson image denoising based on fractional-order total variation, *Inverse Probl. Imag.*, **14** (2020), 77–96. https://doi.org/10.3934/ipi.2019064
49. Y. F. Pu, J. L. Zhou, X. Yuan, Fractional differential mask: a fractional differential-based approach for multiscale texture enhancement, *IEEE Trans. Image Process.*, **19** (2010), 491–511. https://doi.org/10.1109/TIP.2009.2035980
50. Z. Ren, C. He, Q. Zhang, Fractional order total variation regularization for image super-resolution, *Signal Process.*, **93** (2013), 2408–2421. https://doi.org/10.1016/j.sigpro.2013.02.015
51. D. Geman, G. Reynolds, Constrained restoration and recovery of discontinuities, *IEEE Trans. Pattern Anal. Mach. Intell.*, **14** (1992), 367–383. https://doi.org/10.1109/34.120331

52. D. Geman, C. Yang, Nonlinear image recovery with half-quadratic regularization, *IEEE Trans. Image Process.*, **4** (1995), 932–946. https://doi.org/10.1109/83.392335
53. M. Nikolova, M. K. Ng, C. P. Tam, Fast nonconvex nonsmooth minimization methods for image restoration and reconstruction, *IEEE Trans. Image Process.*, **19** (2010), 3073–3088. https://doi.org/10.1109/TIP.2010.2052275
54. S. Oh, H. Woo, S. Yun, M. Kang, Non-convex hybrid total variation for image denoising, *J. Vis. Commun. Image Represent.*, **24** (2013), 332–344. https://doi.org/10.1016/j.jvcir.2013.01.010
55. M. Kang, M. Kang, M. Jung, Nonconvex higher-order regularization based Rician noise removal with spatially adaptive parameters, *J. Visual Commun. Image Represent.*, **32** (2015), 180–193. https://doi.org/10.1016/j.jvcir.2015.08.006
56. T. Adam, R. Paramesran, Hybrid non-convex second-order total variation with applications to non-blind image deblurring, *Signal Image Video Process.*, **14** (2020), 115–123. https://doi.org/10.1007/s11760-019-01531-3
57. Y. Sun, L. Lei, D. Guan, X. Li, G. Xiao, SAR image speckle reduction based on nonconvex hybrid total variation model, *IEEE Trans. Geosci. Remote Sens.*, **59** (2020), 1231–1249. https://doi.org/10.1109/TGRS.2020.3002561
58. P. Ochs, A. Dosovitskiy, T. Brox, T. Pock, On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision, *SIAM J. Imaging Sci.*, **8** (2015), 331–372. https://doi.org/10.1137/140971518
59. E. J. Candés, M. B. Wakin, S. P. Boyd, Enhancing sparsity by reweighted ℓ_1 minimization, *J. Fourier Anal. Appl.*, **14** (2008), 877–905. https://doi.org/10.1007/s00041-008-9045-x
60. J. Eckstein, D. P. Bertsekas, On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators, *Math. Program.*, **55** (1992), 293–318. https://doi.org/10.1007/BF01581204
61. R. Glowinski, *Numerical methods for nonlinear variational problems*, Springer Berlin, Heidelberg, 1984. https://doi.org/10.1007/978-3-662-12613-4
62. S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.*, **3** (2011), 1–122. https://doi.org/10.1561/2200000016
63. H. Carfantan, J. Idier, Statistical linear destriping of satellite-based pushbroom-type images, *IEEE Trans. Geosci. Remote Sens.*, **48** (2010), 1860–1871. https://doi.org/10.1109/TGRS.2009.2033587
64. K. Miller, B. Ross, *An introduction to the fractional calculus and fractional differential equations*, New York, USA: John Wiley & Sons, 1993.
65. K. B. Oldham, J. Spanier, *The fractional calculus: theory and applications of differentiation and integration to arbitrary order*, New York, USA: Academic Press, 1974. https://doi.org/10.1016/s0076-5392(09)x6012-1
66. I. Podlubny, *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, London, UK: Academic Press, 1999. https://doi.org/10.1016/s0076-5392(99)x8001-5
67. L. Vese, T. F. Chan, *Reduced non-convex functional approximations for image restoration & segmentation*, UCLA CAM Report, 1997.

68. H. Attouch, J. Bolte, B. F. 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), 91–129. https://doi.org/10.1007/s10107-011-0484-9
69. L. P. D. Van den Dries, *Tame topology and o-minimal structures*, New York, NY, USA: Cambridge University Press, 1998. https://doi.org/10.1017/CBO9780511525919
70. A. Chambolle, T. Pock, A first-order primal-dual algorithm for convex problems with applications to imaging, *J. Math. Imaging Vis.*, **40** (2011), 120–145. https://doi.org/10.1007/s10851-010-0251-1
71. T. Goldstein, S. Osher, The split Bregman method for L1-regularized problems, *SIAM J. Imaging Sci.*, **2** (2009), 323–343. https://doi.org/10.1137/080725891
72. Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.*, **13** (2004), 600–612. https://doi.org/10.1109/TIP.2003.819861

AIMS Press