remote sensing

MDPI

Article

Remote Sensing Image Stripe Detecting and Destriping Using the Joint Sparsity Constraint with Iterative Support Detection

Yun-Jia Sun [1], Ting-Zhu Huang [1,*], Tian-Hui Ma [2] and Yong Chen [1]

[1] School of Mathematical Sciences/Research Center for Image and Vision Computing, University of Electronic Science and Technology of China, Chengdu 611731, Sichuan, China; sunyunjia_795@163.com (Y.-J.S.); chenyong1872008@163.com (Y.C.)

[2] School of Mathematics and Statistics, Xian Jiaotong University, Xian 710049, Shaanxi, China; nkmth0307@126.com

* Correspondence: tingzhuhuang@126.com; Tel.: +86-28-6183-1016

Received: 18 February 2019; Accepted: 6 March 2019; Published: 13 March 2019

Abstract: Remote sensing images have been applied to a wide range of fields, but they are often degraded by various types of stripes, which affect the image visual quality and limit the subsequent processing tasks. Most existing destriping methods fail to exploit the stripe properties adequately, leading to suboptimal performance. Based on a full consideration of the stripe properties, we propose a new destriping model to achieve stripe detection and stripe removal simultaneously. In this model, we adopt the unidirectional total variation regularization to depict the directional property of stripes and the weighted $\ell_{2,1}$-norm regularization to depict the joint sparsity of stripes. Then, we combine the alternating direction method of multipliers and iterative support detection to solve the proposed model effectively. Comparison results on simulated and real data suggest that the proposed method can remove and detect stripes effectively while preserving image edges and details.

Keywords: alternating direction method of multipliers; destriping; iterative support detection; joint sparsity; remote sensing images

1. Introduction

Remote sensing images have been widely used in many fields, such as urban planning, environment monitoring, and precision farming [1]. Unfortunately, remote sensing images are often contaminated by noises. One common and important noise is the stripe noise, which is mainly due to lines being dropped during scanning, differences between forward and reverse scanning [2], and variations in calibrations across sensor arrays in multisensor instruments. In real applications, stripe noise in remote sensing images not only reduces the image visual quality, but also lowers the accuracy of further processing tasks, such as classification [3], superresolution [4], and unmixing [5,6]. Therefore, the task of destriping is an important problem before further applications. Destriping aims at removing the stripes while maintaining image features, such as edges and details.

1.1. Existing Methods

Existing destriping methods can be categorized into three classes: filter-based, statistics-based, and optimization-based.

Filtering-based methods assume that stripes are periodic and can be distinguished from the power spectrum. Therefore, they suppress the specific frequencies caused by stripe noise in the transform domain, such as low-pass filter [2], wavelet analysis [7–10], and the Fourier transform [2,11]. Although these methods are easy to implement and perform well in specific cases, they generally lead to blurry artifacts in the resulting images due to the shrinkage of the frequencies of image details. To overcome this drawback, Munch et al. [12] proposed a combined Fourier and wavelet filter-based method to exploit the directional property of stripes via wavelet decomposition, achieving results with less blurry artifacts.

Statistics-based methods depend on the statistical characteristic of the digital numbers of each sensor [13–18], such as moment matching [14,17] and histogram matching [15,18]. The histogram matching assumes that the probability distribution of scene radiances seen by each sensor is the same and performs well only for single-type sensors competitively. To solve this problem, Wegner [18] proposed calculating the statistics only over homogeneous image regions; Gorsini et al. [19] applied the radiometric equalization to remove non-periodic stripes. In general, the main problem of statistics-based methods is that these methods greatly depend on the pre-established reference moment or histogram.

The optimization-based methods formulate destriping as an ill-posed problem [20–23], in which the true image and stripes are characterized by prior information [24,25]. For example, Shen and Zhang [22] proposed a maximum a *posteriori* method based on Huber–Markov regularization for image destriping and inpainting. Bouali and Ladjal [20] used the unidirectional total variation (UTV) to exploit the direction features of stripes in Moderate-Resolution Imaging Spectroradiometer (MODIS) images. Subsequently, UTV [26,27] and its variants have been widely used for destriping [21,23,28–31]. For example, Dou et al. [28] proposed a non-convex destriping model by employing the ℓ_0-norm variant of UTV and the global sparsity regularization, which is depicted by the ℓ_1-norm to constrain the stripe noise. The method performs well, but it fails to consider the inner structure among the stripes. Compared with Dou's model, we excavate the directional property of strips more deeply and use a joint sparsity regularization, which is depicted by the weighted $\ell_{2,1}$-norm ($\ell_{w,2,1}$-norm), to constrain the stripe component. Recently, some researchers addressed multispectral and hyperspectral image [32–34] destriping by exploiting the high coherence between different image bands adequately [35,36]. There are several commonly-used sparsity metrics [37] to describe the sparsity of stripes, e.g., ℓ_0-norm, ℓ_1-norm, $\ell_{2,0}$-norm, and $\ell_{2,1}$-norm. Here, ℓ_0-norm and ℓ_1-norm penalize, respectively, the number and the magnitude of non-zero elements, which describes the global sparsity. On the other hand, $\ell_{2,0}$-norm and $\ell_{2,1}$-norm penalize, respectively, the number and the magnitude of non-zero columns, which describes the group sparsity to explore the structural sparsity of stripes. As summarized in Table 1, we discuss the differences and relations among several state-of-the-art optimization-based methods.

Table 1. A comparison of the related optimization-based methods and their properties. UTV, unidirectional total variation.

Method	Stripe Prior			Image Prior	Algorithm
	Sparsity	Low-Rankness	Smoothness	Smoothness	
IRUV[29]	—	—	UTV for factors(ℓ_p norm)	UTV for factors(ℓ_p norm)	an iterative reweighted least squares algorithm
WDSUV [30]	ℓ_0 norm	—	UTV (the combination of the weighted ℓ_1 norm and ℓ_0 norm)	the weighted UTV (ℓ_1 norm)	an alternative direction multiplier method algorithm
LRSID [38]	—	nuclear norm	—	anisotropic TV	an alternative direction multiplier method algorithm
GSLV [39]	ℓ_0 norm	—	UTV (ℓ_1 norm)	UTV (ℓ_1 norm)	an alternative direction multiplier method algorithm
DSM [28]	ℓ_1 norm	—	UTV (ℓ_0 norm)	UTV (ℓ_1 norm)	a proximal alternative direction multiplier method algorithm
GSUTV [40]	$\ell_{2,1}$ norm	—	UTV (ℓ_1 norm)	UTV (ℓ_1 norm)	an alternative direction multiplier method algorithm
JSUTV	$\ell_{w,2,1}$ norm	—	UTV (ℓ_1 norm)	UTV (ℓ_1 norm)	a combined algorithm of the alternative direction multiplier method and iterative support detection

1.2. Motivation

Most existing destriping methods concentrate on estimating the clean image directly from the observed image, while paying little attention to the stripe component. Besides the directional and structural properties of the stripes, the stripe position is also an important feature, and effective exploitation of this feature can significantly improve the destriping performance.

In this paper, we propose a novel destriping model that can detect the stripe position and remove the stripes simultaneously, by fully exploiting the properties of the stripes. Our model exploits both the structural and the directional properties of stripes. On the one hand, we use the $\ell_{w,2,1}$-norm to characterize the joint sparsity (structural property) of stripes. In addition, we use the iterative support detection (ISD) [41,42] to calculate the weight vector in the $\ell_{w,2,1}$-norm, which shows the stripe position in the observed image. On the other hand, we use UTV to characterize the directional property of stripes. We apply the alternating direction method of multipliers (ADMM) [6,43–45] algorithm to solve the proposed model efficiently. In summary, our contributions are as follows:

(1) We propose a non-convex optimization model to characterize the position properties and the joint sparsity of stripes using the $\ell_{w,2,1}$-norm, which can significantly improve the destriping performance.

(2) We combine ADMM and ISD to solve the proposed model effectively. Under specified conditions, the convergence of the proposed method can be guaranteed. In the solving process, we achieve stripe detection by calculating the weight vector. Meanwhile, we design new indices to analyze the accuracy of detection.

The remainder of this paper is organized as follows. Section 2 gives the proposed model with analyses of the stripe properties. Section 3 presents the optimization algorithm. Section 4 shows the experimental results and provides the comparisons with other methods. Section 5 concludes this paper.

2. The Proposed Model

Without loss of generality, we consider each stripe as a column, since we can rotate the image to make the stripe vertical if the stripe is horizontal. We consider the following additive degradation model [20,38]:

$$F = U + S, \tag{1}$$

where $F \in \mathbb{R}^{m \times n}$, $U \in \mathbb{R}^{m \times n}$, and $S \in \mathbb{R}^{m \times n}$ represent the observed image, the ground-truth image, and the stripe noise, respectively. This degradation process is shown in Figure 1.

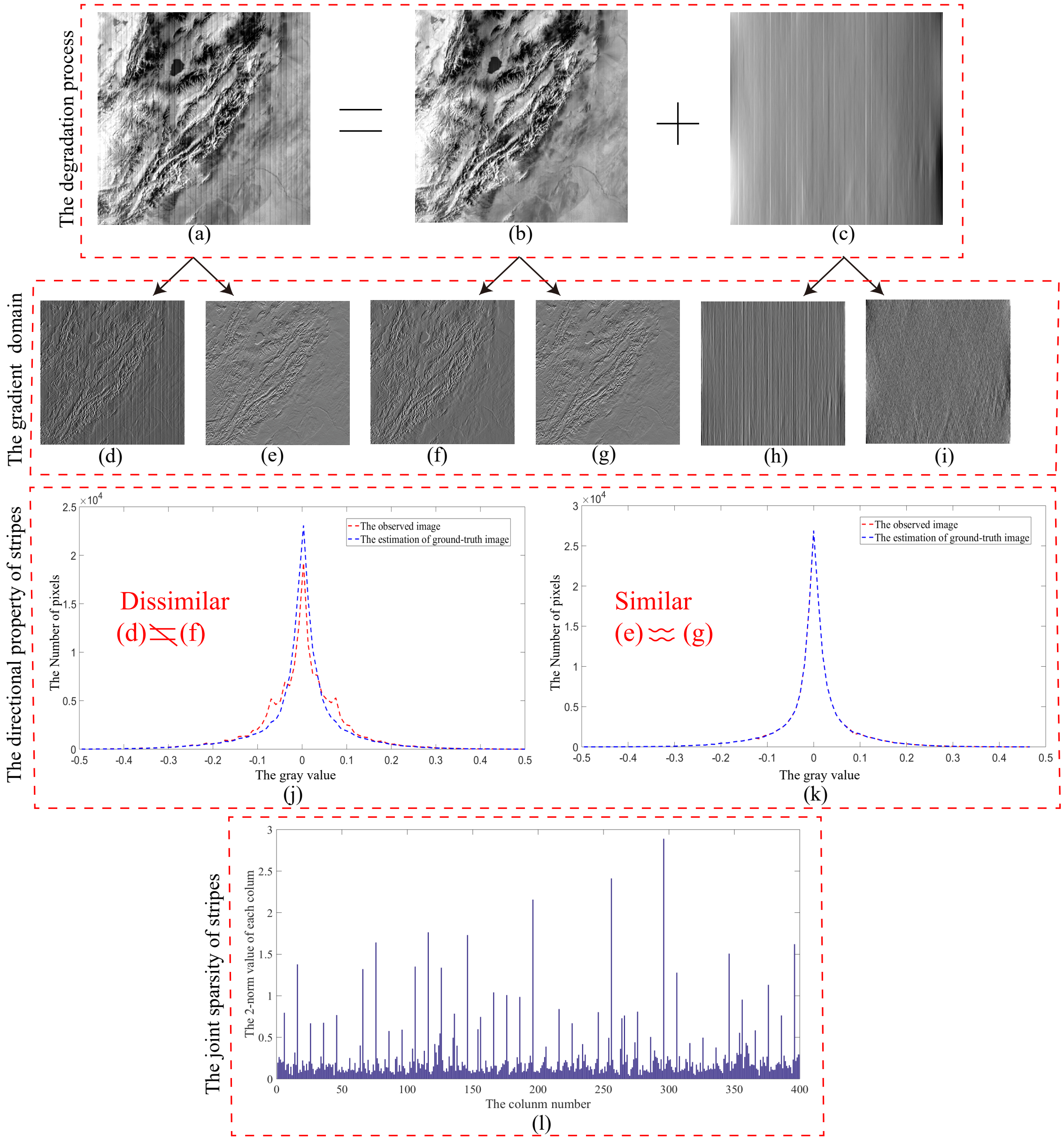


Figure 1. Destriping results in Terra MODIS by GSUTV. The first line: (**a**) Degraded image. (**b**) Destriping image. (**c**) Stripe component. The second line: (**d**) Horizontal gradient of the observed image. (**e**) Vertical gradient of the observed image. (**f**) Horizontal gradient of the destriping image. (**g**) Vertical gradient of the destriping image. (**h**) Horizontal gradient of the stripe component. (**i**) Vertical gradient of the stripe component. The third line: (**j**) Histogram of horizontal gradient. (**k**) Histogram of vertical gradient. The fourth line: (**l**) ℓ_2-norm values of the stripe component.

From a degraded image F, we detect the stripe position and estimate the stripe S simultaneously by solving the following model:

$$\min_{S} \|D_y S\|_{1,1} + \lambda_1 \|S\|_{\omega,2,1} + \lambda_2 \|D_x F - D_x S\|_{1,1}, \tag{2}$$

where $\|D_y S\|_{1,1}$, $\|D_x F - D_x S\|_{1,1}$, and $\|S\|_{\omega,2,1}$ are denoted as:

$$\|D_y S\|_{1,1} := \sum_{j=1}^{n} \|D_y S_{:j}\|_1, \tag{3}$$

$$\|D_x F - D_x U\|_{1,1} := \sum_{j=1}^{n} \|D_x F_{:j} - D_x U_{:j}\|_1, \tag{4}$$

$$\|S\|_{\omega,2,1} := \sum_{j=1}^{n} \omega_j \|S_{:j}\|_2, \tag{5}$$

where the subscript $: j$ denotes the j^{th} column; D_y and D_x are linear second-order difference operators along the vertical direction and the horizontal direction, respectively; $\lambda_1 > 0$ and $\lambda_2 > 0$ are regularization parameters balancing the three terms; $\omega = [\omega_1, \cdots, \omega_n]^\top$ is a weight vector. After estimating the stripe S from (2), the ground-truth image U is obtained by $F - S$.

We explain in detail the motivation of each term in our model. To illustrate the stripe properties appropriately, we use GSUTV [40], which achieves the destriping by estimating the stripe component, to remove stripes in the Terra MODIS image and explore the stripe properties in results. Figure 1a–c presents the degraded image, the destriping result, and the stripe component, respectively. Figure 1d–i shows the gradients of the observed image, the destriping image, and the stripe image in two directions: horizontal (across the stripes) and vertical (along the stripes), respectively.

We analyze the gradient statistical distribution before and after the degradation process firstly. It is easy to see that the horizontal gradient mapping between Figure 1d,f differs greatly in the image domain, while the vertical gradient mapping between Figure 1e,g is similar. In addition, we plot their corresponding gradient histograms in Figure 1j,k. We observe that the gradient distributions of Figure 1d,f are absolutely different, while the gradient distributions of Figure 1e,g are almost the same. Such an observation is not surprising, since the stripe noise has significantly directional feature. As shown in Figure 1c, we observe that the stripe noise possesses a significant structural property. Furthermore, we plot ℓ_2-norm values of the stripe component (shown in Figure 1l).

The smoothness along the vertical direction of the stripe: The first term $\|D_y S\|_{1,1}$ penalizes the gradient of the stripe image along the vertical direction. Figure 1c shows the great smoothness along the vertical direction of the stripe image, and Figure 1i shows that the vertical gradient of the stripe image has significant sparsity. Thus, we chose the ℓ_1-norm of the vertical gradient of stripe image to keep the stripe component smoothness well. Although the ℓ_p-norm $(0 < p < 1)$ [39,46] can depict sparsity more accurately than the ℓ_1-norm [28], here we did not adopt them because they are nonconvex and have a high possibility of affecting the convergence of the algorithm.

Joint sparsity of the stripe: The second term $\|S\|_{\omega,2,1}$ characterizes the joint sparsity of stripes. As shown in Figure 1c, the stripe component has a special column structure. As Chen et al. [40] pointed out, the stripe component is composed by column vectors, and each column can be considered as a group. Moreover, we observe that the structure of the stripe component exhibited joint sparsity in the sense that the non-zeros of $S_{1:}, \cdots, S_{m:}$ absolutely appeared in the same position. This motivated us to use a joint sparsity regularization. As shown in Figure 1l, most of the ℓ_2-norm values of columns are close to zero, while the other values are much bigger than them. This explains again that the joint sparsity regularization is more appropriate to depict the structural property of stripes than the normal group sparsity [47] regularization.

Compared with GSUTV, the main difference of the proposed model is the weight parameter vector ω. We can remove some columns from the regularization by setting the weighted vector ω, if we regard them as true stripe columns. Therefore, the $\ell_{\omega,2,1}$-norm can depict joint sparsity more appropriately than the $\ell_{2,1}$-norm. It is worth mentioning that unlike most of the present weighted models, which acquire $\omega_{:j}$ from the set of positive real numbers, we adopt $\omega_{:j}$ from $\{0,1\}$ due to the specific property of the stripes. This means that if we regard $S_{:j}$ as the true non-zero column (true stripe column), we can forbid it from moving nearer to zero, and set $\omega_{:j}$ as zero. Because the position information about non-zero is unknown, ω_j is not given beforehand. Therefore, we used ISD [42], as a self-learning scheme, to explore partial support information gradually.

The smoothness along the horizontal direction of the ground-truth image: The third term $\|D_xF - D_xS\|_{1,1}$, in the proposed model, penalizes the difference between the gradient of the ground-truth image and the stripe image along the horizontal direction, which enhances the constraint on the smoothness of the ground-truth image along the horizontal direction. Generally speaking, the ground-truth image exhibits smoothness along both horizontal and vertical directions, resulting in sparsity in the gradient domain. However, the stripe noise mainly damages the sparsity along the horizontal direction (Figure 1d), rather than the vertical direction (Figure 1e). Therefore, we chose the ℓ_1-norm of the horizontal gradient of the ground-truth image to make the desired ground-truth image smoother along the horizontal direction.

The framework of the proposed method is illustrated in Figure 2, and the details of the optimization method are shown in the next section.

Remark 1. *The proposed model is non-convex due to the multiplicative nature of ω and S in the regularization term $\|S\|_{\omega,2,1}$. It is worth mentioning that the proposed model is convex with respect to the stripe S.*

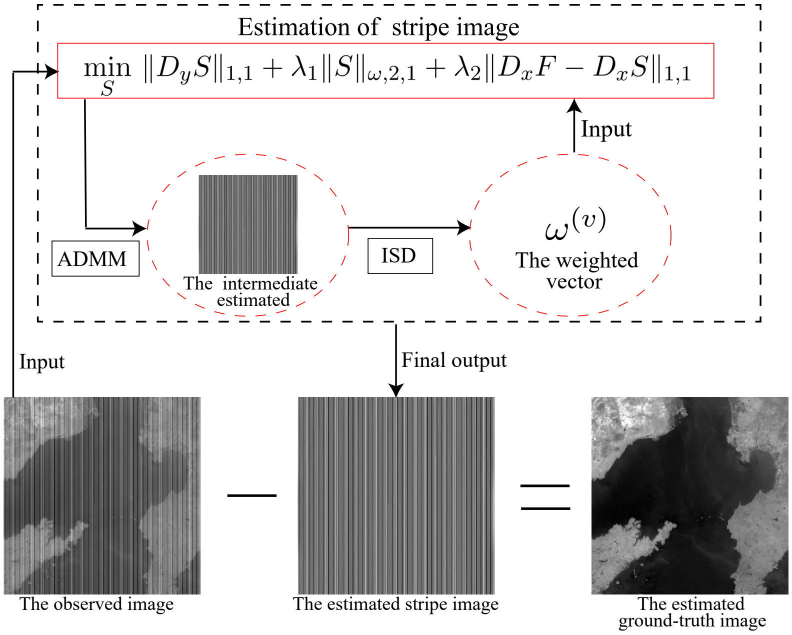

Estimation of stripe image
$\min_S \|D_yS\|_{1,1} + \lambda_1\|S\|_{\omega,2,1} + \lambda_2\|D_xF - D_xS\|_{1,1}$
Input
ADMM
The intermediate estimated
ISD
$\omega^{(v)}$
The weighted vector
Input
Final output
The observed image
The estimated stripe image
The estimated ground-truth image

Figure 2. The framework of the proposed model. ADMM, alternating direction method of multipliers; ISD, iterative support detection.

3. The Optimization Algorithm

To solve the proposed model, we alternately used ISD [42], a self-learning scheme to explore true non-zeros gradually, and ADMM [43,48,49], a useful approach to deal with a non-smooth, but convex optimization model. The optimization method that combines ISD and ADMM is reasonable. The model is hard to solve due to the non-convex regularization term $\|S\|_{\omega,2,1}$ and the nature of the remainder non-smooth regularization terms. To overcome the solving difficulty caused by $\|S\|_{\omega,2,1}$, we extended the idea of ISD in our method from normal sparsity to joint sparsity. We can detect the position of non-zeros and update the value of the weight vector based on ISD. It is easy to see that when the weight parameter vector is a constant, the proposed model degrades to an ordinary convex model in terms of S, and we call it the joint sparsity model. The joint sparsity model, which is non-smooth, but convex, is simple to deal with by ADMM. Therefore, we used ADMM to handle the problem caused by the non-smooth regularizer. Therefore, we solved this model by executing the following two steps:

(1) Step 1: Keep the weight vector fixed, and solve the joint sparsity model by ADMM. In the first iteration, we set the initial value of ω as $[1, \cdots, 1]^\top$ and obtained an estimation S^1 by solving the joint sparsity model.
(2) Step 2: Using the estimated S from Step 1, update ω via a support detection operation.

The details of the two steps are shown as follows.

3.1. Solving the Joint Sparsity Model

The main idea of ADMM is, by introducing auxiliary variables, to transform the unconstrained optimization model into a constrained optimization model with the splitting structure of variables. By introducing three auxiliary variables X, Y, and Z, we rewrite (2) as follows:

$$\begin{aligned} \min_{S,X,Y,Z} \ & \|X\|_{1,1} + \lambda_1 \|Y\|_{\omega,2,1} + \lambda_2 \|Z\|_{1,1} \\ \text{s.t. } & X = D_y S, \\ & Y = S, \\ & Z = D_x F - D_x S. \end{aligned} \tag{6}$$

The augmented Lagrangian function of (6) is as follows:

$$\begin{aligned} L_\beta(S, X, Y, Z, A_1, A_2, A_3) = & \|X\|_{1,1} + \lambda_1 \|Y\|_{\omega,2,1} + \lambda_2 \|Z\|_{1,1} + \langle A_1, D_y S - X \rangle + \langle A_2, S - Y \rangle \\ & + \langle A_3, D_x F - D_x S - Z \rangle + \frac{\beta}{2} \|D_y S - X\|_F^2 + \frac{\beta}{2} \|S - Y\|_F^2 + \frac{\beta}{2} \|D_x F - D_x S - Z\|_F^2, \end{aligned} \tag{7}$$

where A_1, A_2, and A_3 are the Lagrange multipliers and $\beta > 0$ is a penalty scalar. ADMM iterates as follows:

$$\begin{cases} X^{(k+1)} = \underset{X}{\operatorname{argmin}}\, L_\beta(S^{(k)}, X, Y^{(k)}, Z^{(k)}, A_1^{(k)}, A_2^{(k)}, A_3^{(k)}), \\ Y^{(k+1)} = \underset{Y}{\operatorname{argmin}}\, L_\beta(S^{(k)}, X^{(k+1)}, Y, Z^{(k)}, A_1^{(k)}, A_2^{(k)}, A_3^{(k)}), \\ Z^{(k+1)} = \underset{Z}{\operatorname{argmin}}\, L_\beta(S^{(k)}, X^{(k+1)}, Y^{(k+1)}, Z, A_1^{(k)}, A_2^{(k)}, A_3^{(k)}), \\ S^{(k+1)} = \underset{S}{\operatorname{argmin}}\, L_\beta(S, X^{(k+1)}, Y^{(k+1)}, Z^{(k+1)}, A_1^{(k)}, A_2^{(k)}, A_3^{(k)}), \\ A_1^{(k+1)} = A_1^{(k)} + \beta(D_y S^{(k+1)} - X^{(k+1)}), \\ A_2^{(k+1)} = A_2^{(k)} + \beta(S^{(k+1)} - Y^{(k+1)}), \\ A_3^{(k+1)} = A_3^{(k)} + \beta(D_x F - D_x S^{(k+1)} - Z^{(k+1)}). \end{cases} \tag{8}$$

The details of solving the four subproblems are shown as follows:

(1) The X-subproblem can be rewritten by completing the square as follows:

$$\begin{aligned} &\underset{X}{\operatorname{argmin}}\, \|X\|_{1,1} + \langle A_1, D_y S - X\rangle + \frac{\beta}{2}\|D_y S - X\|_F^2 \\ =\, &\underset{X}{\operatorname{argmin}}\, \|X\|_{1,1} + \frac{\beta}{2}\|D_y S - X + \frac{A_1}{\beta}\|_F^2, \end{aligned} \tag{9}$$

whose solution can be given by the following column-wise *vector-soft threshold* function [50,51]:

$$\begin{aligned} X_{ij}^{(k+1)} &= \operatorname{shrink}((D_y S^{(k)} + \frac{A_1^{(k)}}{\beta})_{ij}, \frac{1}{\beta}) \\ &= \max\{|(D_y S^{(k)} + \frac{A_1^{(k)}}{\beta})_{ij}| - \frac{1}{\beta}, 0\} \frac{(D_y S^{(k)} + \frac{A_1^{(k)}}{\beta})_{ij}}{|(D_y S^{(k)} + \frac{A_1^{(k)}}{\beta})_{ij}|}. \end{aligned} \tag{10}$$

Considering the matrix $X \in \mathbb{R}^{m\times n}$, the cost of computing X is $\mathcal{O}(mn)$.

(2) The same as the X-subproblem, the Z-subproblem can be solved by the shrinkage operator:

$$\begin{aligned} &\underset{Z}{\operatorname{argmin}}\, \lambda_2\|Z\|_{1,1} + \langle A_3, D_x F - D_x S - Z\rangle + \frac{\beta}{2}\|D_x F - D_x S - Z\|_F^2 \\ =\, &\underset{Z}{\operatorname{argmin}}\, \lambda_2\|Z\|_{1,1} + \frac{\beta}{2}\|D_x F - D_x S - Z + \frac{A_3}{\beta}\|_F^2, \end{aligned} \tag{11}$$

so:

$$Z_{ij}^{(k+1)} = \max\{|(D_x F - D_x S^{(k)} + \frac{A_3^{(k)}}{\beta})_{ij}| - \frac{\lambda_2}{\beta}, 0\} \frac{(D_x F - D_x S^{(k)} + \frac{A_3^{(k)}}{\beta})_{ij}}{|(D_x F - D_x S^{(k)} + \frac{A_3^{(k)}}{\beta})_{ij}|}. \tag{12}$$

The same as the X-subproblem, the cost of computing Z is also $\mathcal{O}(mn)$.

(3) The Y-subproblem is:

$$\underset{Y}{\operatorname{argmin}}\, \lambda_1\|Y\|_{\omega,2,1} + \langle A_2, S - Y\rangle + \frac{\beta}{2}\|S - Y\|_F^2, \tag{13}$$

which also can be solved by a shrinkage operator [49–51]:

$$Y_{:j}^{(k+1)} = \max\{\|(S^{(k)} + \frac{A_2^{(k)}}{\beta})_{:j}\|_2 - \frac{\lambda_1}{\beta}\omega_j, 0\} \frac{(S^{(k)} + \frac{A_2^{(k)}}{\beta})_{:j}}{\|(S^{(k)} + \frac{A_2^{(k)}}{\beta})_{:j}\|_2}. \tag{14}$$

The cost of computing Y is also $\mathcal{O}(mn)$.

(4) The S-subproblem:

$$\underset{S}{\operatorname{argmin}} \frac{\beta}{2}\|D_y S - X + \frac{A_1}{\beta}\|_F^2 + \frac{\beta}{2}\|S - Y + \frac{A_2}{\beta}\|_F^2 + \frac{\beta}{2}\|D_x F - D_x S - Z + \frac{A_3}{\beta}\|_F^2 \tag{15}$$

is a quadratic optimization, which has the solution:

$$\begin{aligned} S^{(k+1)} = (D_y^\top D_y + I + D_x^\top D_x)^{-1}[D_y^\top (X^{(k+1)} - \frac{A_1^{(k)}}{\beta}) + \\ (Y^{(k+1)} - \frac{A_2^{(k)}}{\beta}) + D_x^\top (D_x F - Z^{(k+1)} + \frac{A_3^{(k)}}{\beta})]. \end{aligned} \tag{16}$$

We assume that the boundary of S is periodic in this paper; thus, $D_y^\top D_y$, $D_x^\top D_x$ are both block cyclic system matrices. Therefore, (16) can be efficiently solved by fast Fourier transforms [42].

$$S^{(k+1)} = \mathcal{F}^{-1} \frac{\xi}{\overline{\mathcal{F}(D_y)} \otimes \mathcal{F}(D_y) + 1 + \overline{\mathcal{F}(D_x)} \otimes \mathcal{F}(D_x)}, \tag{17}$$

where:

$$\xi = \overline{\mathcal{F}(D_y)} \otimes \mathcal{F}(X^{(k+1)} - \frac{A_1^{(k)}}{\beta}) + \mathcal{F}(Y^{(k+1)} - \frac{A_2^{(k)}}{\beta}) + \overline{\mathcal{F}(D_x)} \otimes \mathcal{F}(D_x F - Z^{(k+1)} + \frac{A_3^{(k)}}{\beta}),$$

$\mathcal{F}(\cdot)$ and $\mathcal{F}(\cdot)^{-1}$ denote the fast Fourier transform and its reverse transform, respectively. The symbol $\otimes$ means component-wise multiplication, the the same as the division, and the symbol $\overline{(\cdot)}$ means component-wise complex conjugate. The cost of computing S is $\mathcal{O}(mn \log mn)$ by using fast Fourier transform. At each iteration, the cost of computing all the variables X, Y, Z, and S is $\mathcal{O}(mn \log mn)$.

3.2. Update the Weight Vector ω via ISD

In this step, we used ISD to detect the position of stripes and update the weight vector ω based on the current intermediate solution S.

Firstly, we give the definition of the set whose elements are indices of detected nonzero columns as follows:

$$I^{(v+1)} := \{j : |s_j^{(v)}| > \epsilon^{(v)}\}, v = 1, 2, \cdots, \tag{18}$$

where $s_j = \|S_{:j}\|_2$, s is a sequence containing $|s_1|, \cdots, |s_n|$.

Secondly, we chose threshold $\epsilon^{(v)}$ by the "first significant jump". In detail, after sorting sequence $s^{(v)}$ from small to large, the "first significant jump" scheme finds the smallest j satisfying the inequality:

$$|s_{k_{j+1^{(v)}}}| - |s_{k_{j^{(v)}}}| > \tau^{(v)}, \tag{19}$$

where s_{k_j} is the j^{th} number of $\overline{s^{(v)}}$, which is sorted. There are several easy methods [42] to define $\tau^{(v)}$; in this case, we set it as the average value of $|s_1|, \cdots, |s_n|$.

Thirdly, we set $\epsilon^{(v)} = |s_{k_{j^{(v)}}}|$. It is easy to see that the true stripe entries of $|s^{(v)}|$ are large in magnitude and small in number, and the false ones are large in number and small in magnitude. Therefore, the "first significant jump" scheme spread out the magnitudes of the true stripe entries and clustered those of the false ones.

Finally, we can obtain the index set $I^{(v+1)}$ and give the definition of the set whose elements are indices of detected nonzero columns as follows:

$$T^{(v+1)} := (I^{(v+1)})^c = \{\{1, 2, \cdots, n\} \backslash I^{(v+1)}\}. \tag{20}$$

We update the weight vector $\omega^{(v+1)}$ according to the two index sets as follows:

$$\begin{cases} \omega_j^{(v+1)} = 1, j \in T^{(v+1)}, \\ \omega_j^{(v+1)} = 0, j \in I^{(v+1)}. \end{cases} \tag{21}$$

The main computational cost of the weight vector ω stems from computing the S in the inner loop. Moreover, the iteration number is small empirically, as per the following discussion in Section 4.1. Therefore, the total complexity of the proposed algorithm is $\mathcal{O}(mn \log mn)$.

The pseudocode of the algorithm combining ADMM and ISD is summarized in Algorithm 1.

Algorithm 1 The optimization algorithm combining ADMM and ISD.

```
Input: The observed image $F$, the parameters $\lambda_1$, $\lambda_2$, the penalty parameter $\beta$, the
          outer iteration number $V$, and the inner iteration number $K_{max}$.
Initialize: $\omega_j^{(0)} = 1, (j = 1, \cdots, n)$.
Iteration:
    for $v = 0$ to $V$ do
        for $k = 0$ to $K_{max}$ do
        Initialize for inner loop: $\omega = \omega^{(v)}$, $S^{(0)} = 0$, $X^{(0)} = 0$, $Y^{(0)} = 0$, $Z^{(0)} = D_x F$,
        $A_i^{(0)} = 0 (i = 1, 2, 3)$, and $\varepsilon = 10^{-4}$.
        Update $X^{(k+1)}$, $Y^{(k+1)}$, $Z^{(k+1)}$ by column-wise vector-soft threshold via (10)
            (14), and (12).
        Update $S^{(k+1)}$ by (16).
        Update the multiplier $A_i^{(k+1)} (i = 1, 2, 3)$ by (8).
        Check the convergence condition:
            $\|(F - S^{(k+1)}) - (F - S^{(k)})\| / \|(F - S^{(k)})\| \leq \varepsilon$.
        end for
    Estimate the intermediate ground-truth image $U^{(v)} = F - S^{(k)}$.
    Calculate the index set $I^{(v+1)}$ by the method of threshold choice for $S = S^{(k)}$.
    Update the weight vector $\omega^{(v+1)}$ by (21).
    end for
Output: The final weight vector $\omega^{(V)}$ and the estimation of the ground-truth image U.
```

Remark 2. *Although we set ω from the set of $\{0, 1\}$, we can consider ω from real continuous values as an alternative.*

4. Experiments

In this section, we conduct experiments on both simulated and real data to test the efficiency of the proposed method. Five state-of-the-art destriping methods were chosen for comparison: the filtering-based method [12] (WAFT), the statistics-based method [13] (SLD), and optimization-based methods such as the weighted UTV-based method [29] (WDSUV), the low-rank decomposition-based method [38] (LRSID), and the group sparsity regularization-based method [40] (GSUTV). In the simulated experiments, the original MODIS Image Band 32 of size 400 × 400, which is available at https://ladsweb.nascom.nasa.gov/, and the IKONOS image of size 377 × 331, which can be downloaded from https://openremotesensing.net/, were used by adding synthetic stripe noises to the clean image. In the real experiments, we used two real MODIS images, which are available at https://ladsweb.nascom.nasa.gov/, degraded by different types of stripes. For the convenience of quantitative evaluation and parameter selection, the gray value of the test images was normalized to [0,1]. We used a desktop of 16 GB RAM, Intel (R) Core (TM) i5-4590U CPU, @3.30 GHz to run all codes in MATLAB (R2017a).

To evaluate the results of both simulated and real experiments, we chose several qualitative and quantitative assessments. The qualitative assessments include the visual effect of destriping image and the mean cross-track profile [38]. In the simulated experiments, we chose full-reference evaluation indices: the peak signal-to-noise ratio (PSNR) and the structural similarity (SSIM) [52]. Moreover, to evaluate the correctness of stripe detection, we defined two indices called the detection error rate (DER) and the detection missing rate (DMR) as follows:

$$\begin{aligned} DER(S,W) &= \frac{a}{n}, \\ DMR(S,W) &= \frac{b}{n}, \end{aligned} \tag{22}$$

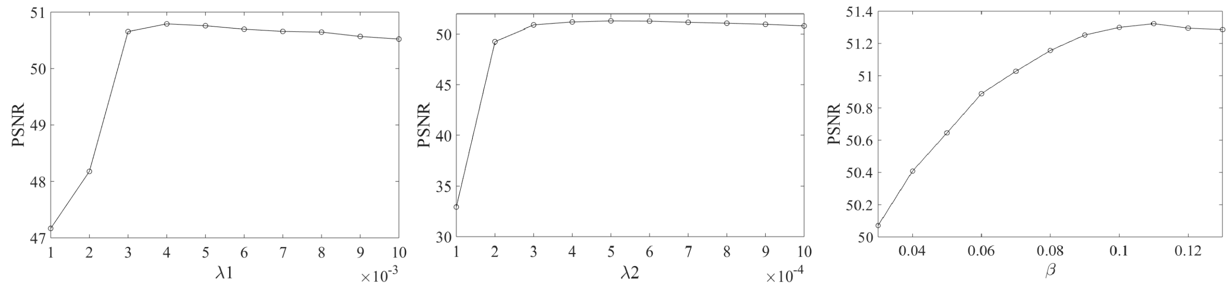
where S is the synthetic stripe component, W is the last weight vector, a is the number of error detected stripe columns, and b is the number of missing detected stripe columns. In real experiments, we selected two no-reference indices including noise reduction (NR) [16,22,36] and mean relative deviation (MRD) [16,36], since the ground-truth image was unavailable. NR depicts the ratio of stripes reduction in the frequency domain; MRD evaluates the ability of the method to retain the pixel value in stripe-free regions. In short, larger values of PSNR, SSIM, and NR indicate better destriping performance, and smaller values of MRD indicate better destriping performance.

4.1. Parameter Setting

In the inner loop, there are two regularization parameters λ_1, λ_2 and the penalty parameter β. We empirically found that the optimal values of λ_1 and λ_2 varied for different test images. Therefore, we used the hand-tuning strategy to tune them. To show the effects of the three parameters on the destriping performance, we performed an analysis by using a simulated experiment (Case 3) as an example. Figure 3 shows the corresponding PSNR index for different λ_1, λ_2, and β, respectively. From Figure 3, we find that the PSNR curves reached the highest point at $\lambda_1 = 0.004$ ($\lambda_2 = 0.001$, $\beta = 0.1$), $\lambda_2 = 0.0005$ ($\lambda_1 = 0.004$, $\beta = 0.1$), and $\beta = 0.1$ ($\lambda_1 = 0.004$, $\lambda_2 = 0.0005$). In our experiments, we set the $\lambda_1 \in [0.001, 0.01]$ with increments of 0.001 and $\lambda_2 \in [0.0001, 0.001]$ with increments of 0.0001. In theory [43], the penalty parameter β does not impact the convergence of ADMM greatly, so we set $\beta \in [0.1, 1]$ with increment of 0.1 by experience.

In the outer loop, the key index is the iterative number that decides the threshold. Fan et al. [41] pointed out that the threshold value will not change much, even if the iterative number goes to infinity.

Empirically, we set the iterative number as five to stabilize the threshold. For the compared methods, we set the parameters as the suggested values in the references.



(a) (b) (c)

Figure 3. Parameter analysis of Case 3. The PSNR curve as a function of: (**a**) λ_1; (**b**) λ_2; (**c**) β.

4.2. Simulated Experiments

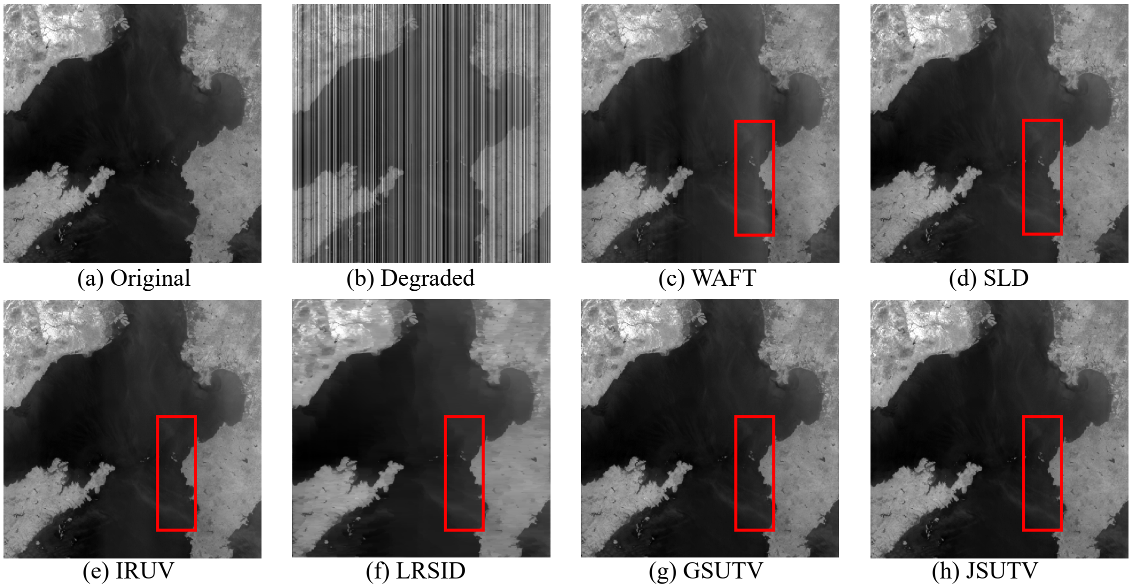
In the simulated experiments, we added both periodical and non-periodical types of synthetic stripe based on the observed model (1) to two different remote images. We denoted the periodic stripe level by using a vector with three elements, e.g., (period, intensity, and rate), where "period" denotes the number of columns between each stripe column (denoted p), "intensity" denotes the absolution value of pixels of each stripe column (denoted I), and "rate" denotes the rate of the stripe area within the ground-truth image (denoted r). Similar to the periodic stripe, we denoted the non-periodic stripe level by using a vector with two elements, e.g., (intensity and rate), where "intensity" and "rate" have the same definition as the elements listed above. In our simulation experiments, p is in the set of $\{10, 15\}$, I is in the set of $\{50, 100\}$, and r is in the set of $\{0.2, 0.5, 0.8\}$. For both periodic and nonperiodic cases, the stripe locations were randomly distributed on the image, and the absolute value of the stripe column pixels was the same. From Case 1–3, we gradually increased the rate of the stripes with $p = 10, I = 50$, leading to some different striping effects. From Case 4–6, we gradually increased the rate of the stripes with $p = 15, I = 50$. From Case 7–9, the rate also has been increased with $I = 50$. From Case 10–12, the rate also has been increased with $I = 100$.

As shown in Figures 4 and 5, the visual results of Case 9 and Case 3 are displayed as examples. In Figure 4, we observe that the WAFT method blurred the boundaries and many residual stripes still existed; the IRUV method lost the detailed information; the LRSID method blurred the whole image. Although the stripes were well removed by the SLD and GSUTV methods, some blurred regions are shown in the red circle. As a comparison, the proposed method achieved better visual performance.

Figures 6 and 7 show the estimated stripe images of different methods and the corresponding error images. The error images of the five comparative methods show that they failed to estimate the stripe component precisely. In contrast, the error image of the proposed method had fewer stripe detail information, which suggests that the proposed method had a better destriping performance.

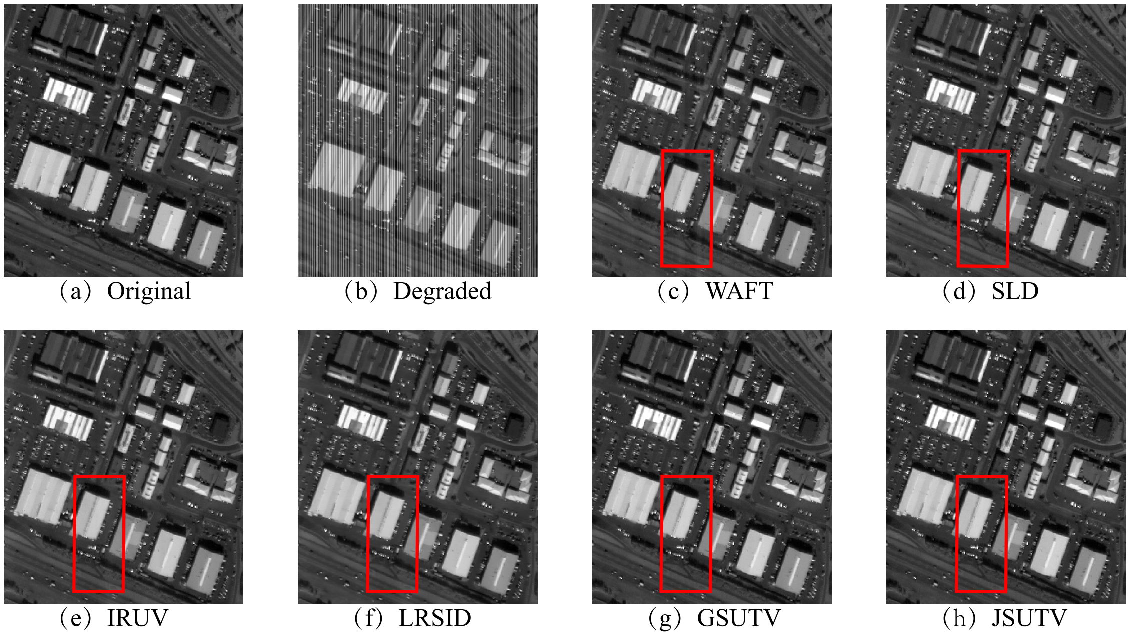
The quantitative results of all the cases are listed in Table 2. Moreover, to test the effect of stripe position detecting, we show the results of DER and DMR in Table 3. From Table 2, we observe that the proposed method obtained better PSNR and SSIM values in the simulation experiments. From Table 3, with the striping effect increasing, the DER values increased gradually, and most of the DMR values were zero. In general, it is clear that the proposed method could detect the stripe position almost precisely. In summary, the quantitative comparison conformed with the above visual results presented.

In addition, Figures 8 and 9 show the mean cross-track profiles of Figures 4 and 5, respectively. Figure 8 shows that the proposed method kept a right curve tendency, and the column mean values of the destriping image were almost as the same as the true image. This illustrates that the proposed method can preserve the edges and details of the image, while it can accurately estimate the stripe component.


(a) Original (b) Degraded (c) WAFT (d) SLD

(e) IRUV (f) LRSID (g) GSUTV (h) JSUTV

Figure 4. Simulated destriping results of Case 9. (**a**) Original image; (**b**) degraded image; destriping images of: (**c**) WAFT; (**d**) SLD; (**e**) IRUV; (**f**) LRSID; (**g**) GSUTV; (**h**) JSUTV.


(a) Original (b) Degraded (c) WAFT (d) SLD

(e) IRUV (f) LRSID (g) GSUTV (h) JSUTV

Figure 5. Simulated destriping results of Case 3. (**a**) Original image; (**b**) degraded image; destriping images of: (**c**) WAFT; (**d**) SLD; (**e**) IRUV; (**f**) LRSID; (**g**) GSUTV; (**h**) JSUTV.

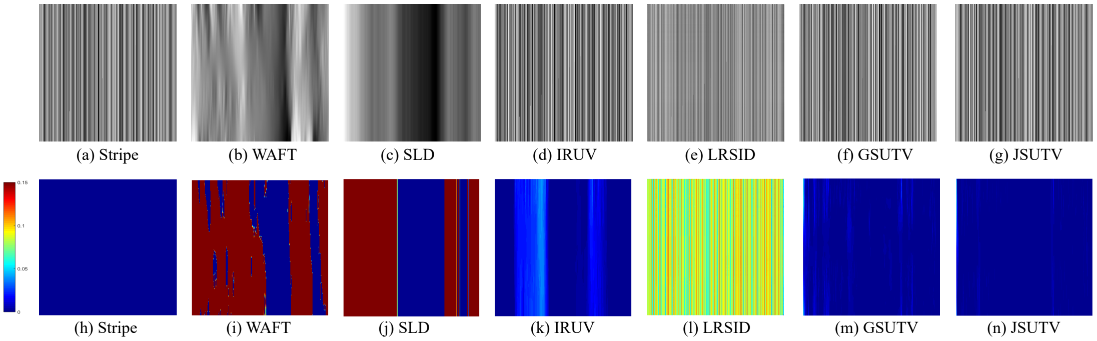


Figure 6. The estimated stripe of Case 9. (**a**) The true stripe. Estimated stripe images by (**b**) WAFT, (**c**) SLD, (**d**) IRUV, (**e**) LRSID, (**f**) GSUTV, and (**g**) JSUTV. The corresponding error images of (**h**) the true stripe, (**i**) WAFT, (**j**) SLD, (**k**) IRUV, (**l**) LRSID, (**m**) GSUTV, and (**n**) JSUTV.

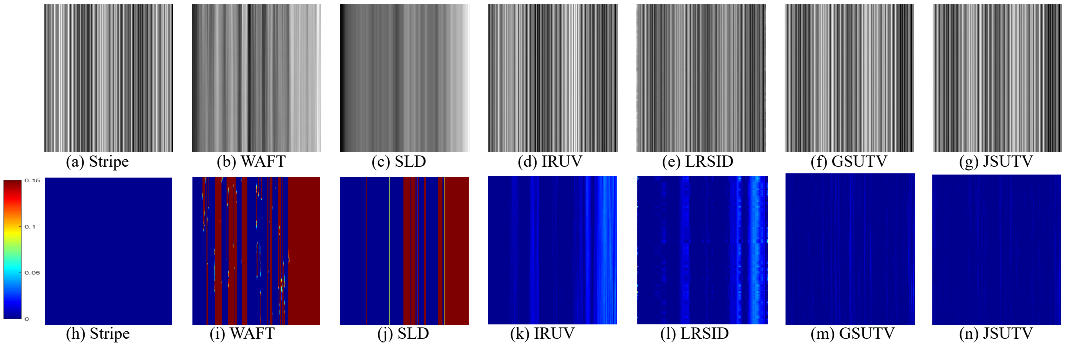


Figure 7. The estimated stripe of Case 3. (**a**) The true stripe. Estimated stripe images by (**b**) WAFT, (**c**) SLD, (**d**) IRUV, (**e**) LRSID, (**f**) GSUTV, and (**g**) JSUTV. The corresponding error images of (**h**) the true stripe, (**i**) WAFT, (**j**) SLD, (**k**) IRUV, (**l**) LRSID, (**m**) GSUTV, and (**n**) JSUTV.

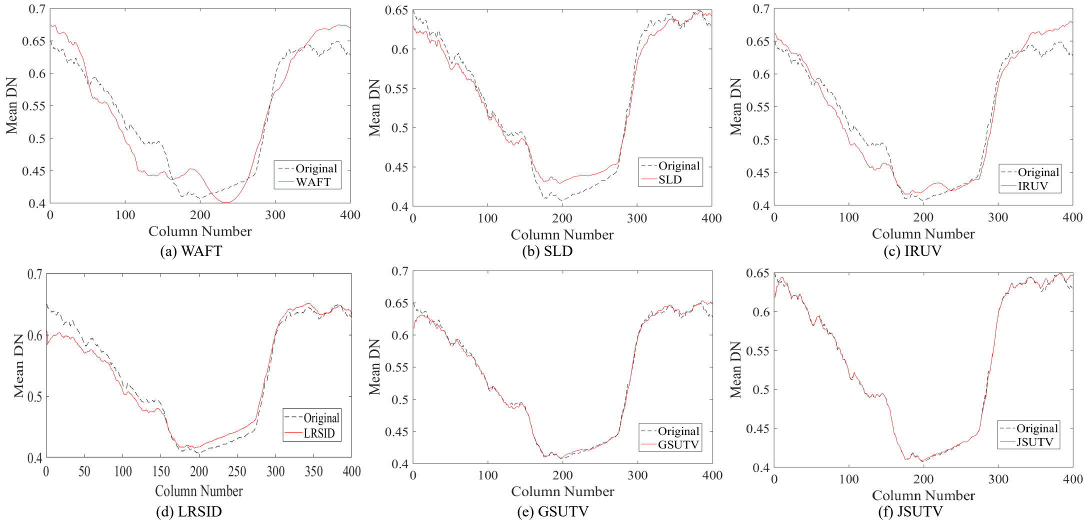
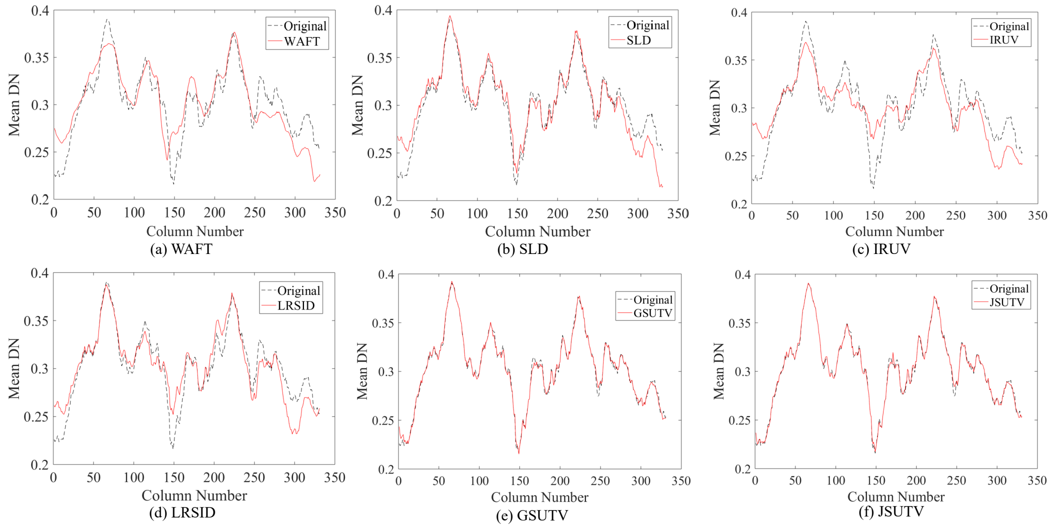


Figure 8. Column mean cross-track profiles of Figure 4. Column mean cross-track profiles' comparison between the original image and the estimated stripe by: (**a**) WAFT; (**b**) SLD; (**c**) IRUV; (**d**) LRSID; (**e**) GSUTV; and (**f**) JSUTV.



(a) WAFT (b) SLD (c) IRUV (d) LRSID (e) GSUTV (f) JSUTV

Figure 9. Column mean cross-track profiles of Figure 5. Column mean cross-track profiles' comparison between the original image and the estimated stripe by: (**a**) WAFT; (**b**) SLD; (**c**) IRUV; (**d**) LRSID; (**e**) GSUTV; and (**f**) JSUTV.

Table 2. Quantitative results (PSNR (dB), SSIM values, and CPU time) of the simulated experiments.

Stripe Type	Case	Index	Degrade	WAFT	SLD	IRUV	LRSID	GSUTV	JSUTV
Periodical (image: IKONOS)	Case 1	PSNR	17.62	38.07	41.27	40.58	40.26	53.50	**54.78**
	(p= 10, r = 0.2, I = 100)	SSIM	0.403	0.991	0.998	0.997	0.994	**0.999**	**0.999**
		CPU time	—	0.433	**0.139**	2.877	6.890	4.507	5.142
	Case 2	PSNR	13.57	33.22	30.31	30.28	29.76	53.27	**54.26**
	(p = 10, r = 0.5, I = 100)	SSIM	0.183	0.980	0.980	0.978	0.968	**0.999**	**0.999**
		CPU time	—	0.372	**0.322**	18.260	7.916	5.070	5.343
	Case 3	PSNR	11.57	34.23	37.45	34.86	34.53	49.39	**51.30**
	(p = 10, r = 0.8, I = 100)	SSIM	0.137	0.986	0.997	0.992	0.983	0.998	**0.999**
		CPU time	—	0.523	**0.278**	2.783	6.655	4.996	5.503
	Case 4	PSNR	17.52	32.48	33.05	32.96	32.87	52.33	**54.73**
	(p = 15, r = 0.2, I = 100)	SSIM	0.409	0.981	0.989	0.988	0.982	**0.999**	**0.999**
		CPU time	—	0.441	**0.288**	3.083	7.685	2.690	4.382
	Case 5	PSNR	13.90	33.22	34.72	34.41	33.86	51.39	**53.93**
	(p = 15, r = 0.5, I = 100)	SSIM	0.194	0.981	0.992	0.989	0.979	0.998	**0.999**
		CPU time	—	0.457	**0.449**	2.824	7.692	3.599	4.996
	Case 6	PSNR	11.50	34.38	40.07	35.88	34.25	48.81	**50.12**
	(p = 15, r = 0.8, I = 100)	SSIM	0.134	0.983	0.998	0.993	0.975	**0.998**	**0.998**
		CPU time	—	0.288	**0.180**	2.782	7.059	5.498	5.298
Non-periodical (image: MODIS)	Case 7	PSNR	19.79	36.39	44.20	37.59	38.97	51.00	**52.60**
	(r = 0.2, I = 50)	SSIM	0.398	0.986	0.998	**0.999**	0.993	**0.999**	**0.999**
		CPU time	—	0.222	**0.318**	11.030	7.499	3.490	3.678
	Case 8	PSNR	16.10	31.00	34.40	36.55	35.08	50.57	**51.89**
	(r = 0.5, I = 50)	SSIM	0.223	0.975	0.991	0.996	0.983	**0.999**	**0.999**
		CPU time	—	0.273	**0.200**	8.056	9.837	5.311	5.765
	Case 9	PSNR	14.21	30.78	37.78	34.33	33.90	45.17	**47.78**
	(r = 0.8, I = 50)	SSIM	0.182	0.975	0.998	0.994	0.978	0.997	**0.999**
		CPU time	—	**0.287**	0.316	7.250	8.147	5.558	5.982
	Case 10	PSNR	17.35	33.21	43.93	38.88	38.04	50.75	**51.61**
	(r = 0.2, I = 100)	SSIM	0.266	0.982	0.997	0.995	0.991	**0.999**	**0.999**
		CPU time	—	0.232	**0.078**	7.250	8.837	3.909	4.092
	Case 11	PSNR	13.55	30.07	33.62	33.33	32.45	46.45	**49.231**
	(r = 0.5, I = 100)	SSIM	0.154	0.976	0.993	0.996	0.988	**0.999**	**0.999**
		CPU time	—	0.241	**0.076**	8.906	9,873	5.121	5.738
	Case 12	PSNR	11.57	29.35	32.25	32.14	31.32	42.67	**46.76**
	(r = 0.8, I = 100)	SSIM	0.117	0.974	0.997	0.997	0.973	0.998	**0.999**
		CPU time	—	0.289	**0.248**	8.376	8.983	6.004	6.839

Table 3. The detection error rate (DER) and detection missing rate (DMR) results of simulated experiments.

Case	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8	Case 9	Case 10	Case 11	Case 12
DER	0.0000	0.0000	0.0030	0.0000	0.0000	0.0000	0.0025	0.0050	0.0250	0.0075	0.0125	0.0075
DMR	0.0000	0.0000	0.0000	0.0000	0.0000	0.0030	0.0025	0.0125	0.0100	0.0000	0.0000	0.0000
a	0	0	1	0	0	0	1	2	10	3	5	3
b	0	0	0	0	0	1	1	5	4	0	0	0

4.3. Real Experiments

In this subsection, Figures 10 and 11 show the results from all six methods on two real remote sensing images. The real images of different stripe noise were selected from Terra MODIS and Aqua MODIS, as shown in Figures 10a and 11a. The Terra MODIS image was highly influenced by non-periodical stripe noise, and the Aqua MODIS image was gravely contaminated by a periodical one.

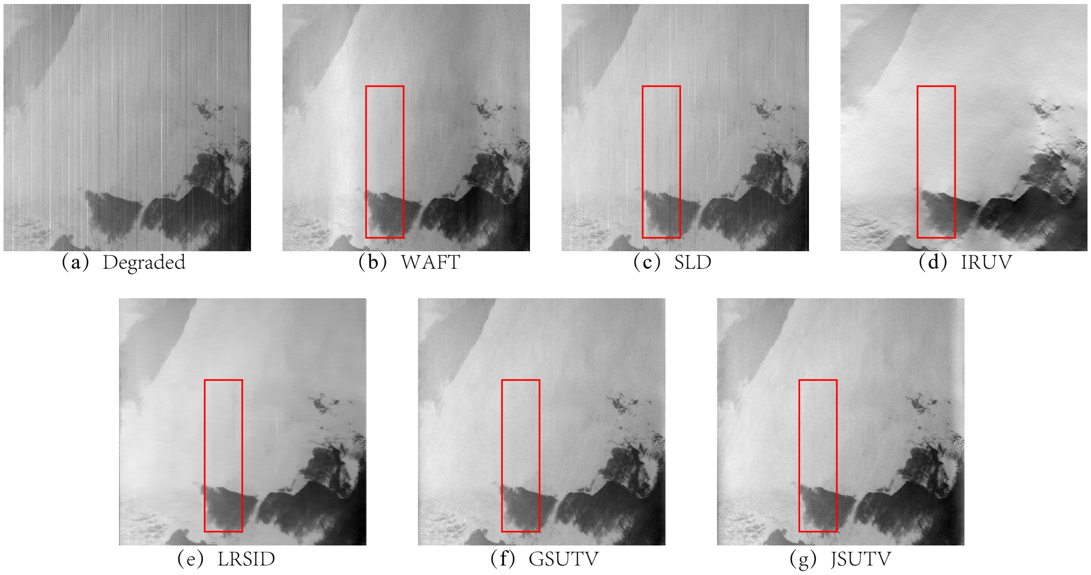


Figure 10. Destriping results of the Terra MODIS image. (**a**) Degraded image; destriping images by: (**b**) WAFT; (**c**) SLD; (**d**) IRUV; (**e**) LRSID; (**f**) GSUTV; (**g**) JSUTV.

As shown in Figure 10, the WAFT and SLD methods blurred the boundaries, and many residual stripes still existed; the IRUV method removed the stripes completely, while blurring some regions; the LRSID method had a bad performance in preserving the stripe-free information, and some residual stripes still existed. For the GSUTV method, all stripes could be removed completely, but some edge regions are shallower in Figure 10f. The proposed method successfully suppressed the stripe noise with fewer artifacts. Figures 10 and 11 show that the JSUTV method was better than the five compared methods in terms of removing stripes and preserving the stripe-free information.

Moreover, in Table 4, we list the quantitative NR and MRD indices' values of the real experiments. In order to avoid the affect of external factors, five 10×10 homogeneous regions were selected to calculate MRD, then to obtain the mean value of MRD. For the NR and MRD indices, we can see that the proposed method achieved better results. In summary, the proposed method can obtain both a better visual effect and quantitative indices.

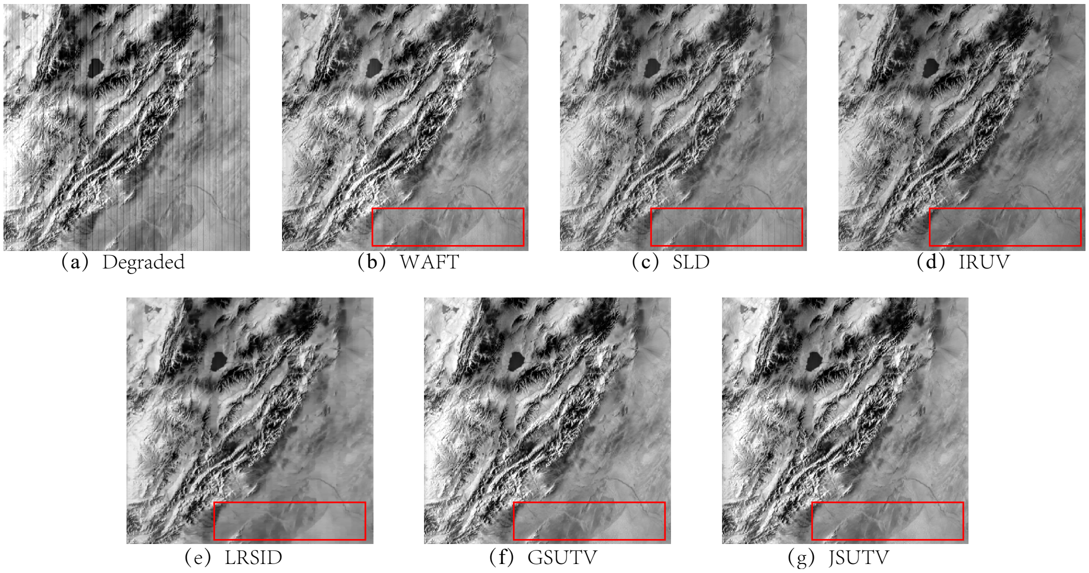


Figure 11. Destriping results of the Aqua MODIS image. (**a**) Degraded image; destriping images by: (**b**) WAFT; (**c**) SLD; (**d**) IRUV; (**e**) LRSID; (**f**) GSUTV; (**g**) JSUTV.

Table 4. Quantitative results on real data using NR and mean relative deviation (MRD).

Image	Index	WAFT	SLD	IRUV	LRSID	GSUTV	JSUTV
Terra MODIS image	NR	2.94	2.98	4.82	**4.90**	3.44	3.71
	MRD	0.1101	0.0475	0.0153	0.9961	0.0758	**0.0101**
Aqua MODIS image	NR	3.45	2.44	3.53	**3.73**	3.56	3.60
	MRD	0.1087	**0.0076**	0.1140	0.9960	0.0101	0.0387

4.4. Numerical Convergence of the Proposed Algorithm

Since our model was convex for a fixed ω, and the convergence of the S subproblem could be guaranteed according to the convergence result of ADMM. We give the following Figure 12 to numerically illustrate the convergence of our algorithm. Figure 12 shows the relative error curves of the successive estimated stripes $S^{(k)}$ and $S^{(k+1)}$, and the relative error kept decreasing as the iteration number increases, which illustrates that the proposed algorithm was convergent numerically.

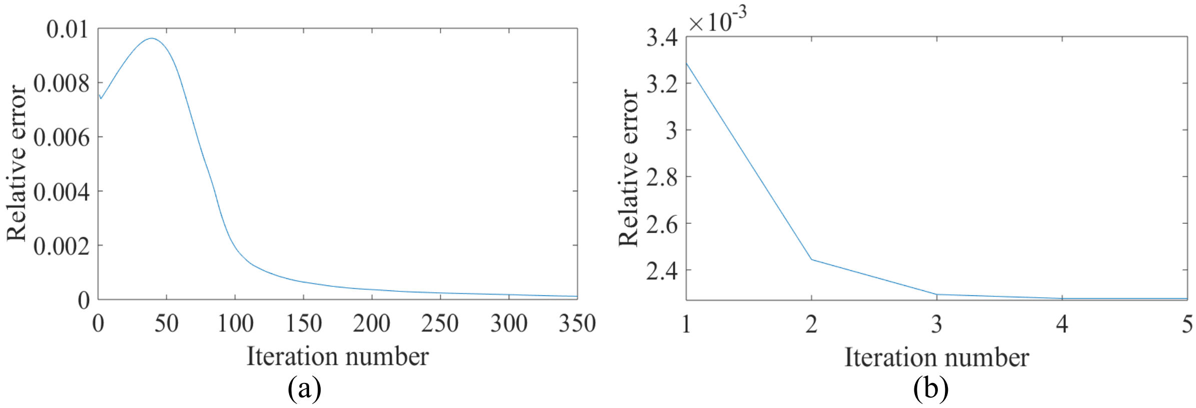


Figure 12. Curves of relative error values versus: (**a**) inner iterations; (**b**) outer iterations.

5. Conclusions

In this paper, we have proposed a new destriping model to achieve the stripe detection and stripe removal simultaneously. The proposed model exploits both the stripe directional property and the stripe structural property and utilizes the joint sparsity prior to describe the structural property. We have proposed a multi-stage alternative optimization method, which combines ISD and ADMM, to solve the minimization problem. The proposed method has been compared with WAFT, SLD, IRUV, LRSID, and GSUTV, and experimental results show the superiority of the proposed method in terms of both quantitative and qualitative assessments. Moreover, simulated experimental results show the accuracy of the stripe detection.

Although the proposed method removes both periodic and nonperiodic stripes well in the horizontal or vertical direction, it still has a shortcoming: it cannot remove oblique stripes. Thus, we will focus on improving our method to overcome the mentioned challenge.

Author Contributions: All authors contributed to the design of the methodology and the validation of experimental exercise: Y.-J.S. and T.-Z.H. wrote the draft; T.-H.M. and Y.C. reviewed and revised the paper.

Funding: This research is supported by NSFC (61772003), the Fundamental Research Funds for the Central Universities (ZYGX2016J132), the National Postdoctoral Program for Innovative Talents (BX20180252), and the Project funded by China Postdoctoral Science Foundation (2018M643611).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chen, Y.; Huang, T.Z.; Zhao, X.L. Destriping of multispectral remote sensing image using low-rank tensor decomposition. *IEEE J. STARS* **2018**, *11*, 4950–4967. [CrossRef]
2. Chen, J.; Shao, Y.; Guo, H.; Wang, W. Destriping CMODIS data by power filtering. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 2119–2124. [CrossRef]
3. Zhu, Z.; Yin, H.; Chai, Y.; Li, Y.; Qi, G. A novel multi-modality image fusion method based on image decomposition. and sparse representation. *Inf. Sci.* **2018**, *432*, 516–529. [CrossRef]
4. Chappalli, M.B.; Bose, N.K. Simultaneous noise filtering and super-resolution with second-generation wavelets. *IEEE Signal Process. Lett.* **2005**, *12*, 772–775. [CrossRef]
5. Iordache, M.D.; Bioucas-Dias, J.M.; Plaza, A. Total variation spatial regularization for sparse hyperspectral unmixing. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 4484–4502. [CrossRef]
6. Zhao, X.L.; Wang, F.; Huang, T.Z.; Ng, M.K.; Plemmons, R.J. Deblurring and sparse unmixing for hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4045–4058. [CrossRef]
7. Mann, B.L.; Koger, C.H.; Li, J. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 2331–2338.
8. Murphy, J.M.; Le, M.J.; Harding, D.J. Automatic Image Registration of Multi-Modal Remotely Sensed Data with Global Shearlet Features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 1685–1704. [CrossRef] [PubMed]
9. Pande-Chhetri, R.; Abd-Elrahman, A. De-striping hyperspectral imagery using wavelet transform and adaptive frequency domain filtering. *ISPRS J. Photogramm. Remote Sens.* **2011**, *66*, 620–636. [CrossRef]
10. Torres, J.; Infante, S.O. Wavelet analysis for the elimination of striping noise in satellite images. *Opt. Eng.* **2001**, *40*, 1309–1314.
11. Pan, J.J.; Chang, C.I. Destriping of landsat MSS images by filtering techniques. *Photogramm. Eng. Remote Sens.* **1992**, *58*, 1417–1423.
12. Munch, B.; Marone, F.; Stampanoni, M.; Trtik, P. Stripe and ring artifact removal with combined wavelet—Fourier filtering. *Opt. Express* **2009**, *17*, 8567–8591. [CrossRef]
13. Carfantan, H.; Idier, J. Statistical linear destriping of satellite-based pushbroom-type images. *IEEE Trans. Geosci. Remote Sens.* **2010**, 48, 1860–1871. [CrossRef]

14. Gadallah, F.L.; Csillag, F.; Smith, E.J.M. Destriping multisensor imagery with moment matching. *Int. J. Remote Sens.* **2000**, *21*, 2505–2511. [CrossRef]
15. Horn, B.K.P.; Woodham, R.J. Destriping LANDSAT MSS images by histogram modification. *Comput. Graph. Image Process.* **1979**, *10*, 69–83. [CrossRef]
16. Ma, N.; Zhou, Z.; Luo, L.; Wang, M. Stripe noise reduction in MODIS data: a variational approach. *Proc. SPIE Int. Soc. Opt. Eng.* **2011**, *8193*, 393–403.
17. Sun, L.; Neville, R.; Staenz, K.; White, H.P. Automatic destriping of Hyperion imagery based on spectral moment matching. *Can. J. Remote Sens.* **2008**, *34* (Supp. 1), S68–S81. [CrossRef]
18. Wegener, M. Destriping multiple sensor imagery by improved histogram matching. *Int. J. Remote Sens.* **1990**, *11*, 859–875. [CrossRef]
19. Corsini, G.; Diani, M.; Walzel, T. Striping removal in MOS-B data. *IEEE Trans. Geosci. Remote Sens.* **2000**, *38*, 1439–1446. [CrossRef]
20. Bouali, M.; Ladjal, S. Toward optimal destriping of MODIS data using a unidirectional variational model. *IEEE Trans. Geosci. Remote Sens.* **2011**, 49, 2924–2935. [CrossRef]
21. Chang, Y.; Fang, H.; Yan, L.; Liu, H. Robust destriping method with unidirectional total variation and framelet regularization. *Opt. Express* **2013**, 21, 23307–23323. [CrossRef] [PubMed]
22. Shen, H.; Zhang, L. A MAP-based algorithm for destriping and inpainting of remotely sensed images. *IEEE Trans. Geosci. Remote Sens.* **2009**, *47*, 1492–1502. [CrossRef]
23. Zhang, Y.; Zhou, G.; Yan, L.; Zhang, T. A destriping algorithm based on TV-Stokes and unidirectional total variation model. *Opt. Int. J. Light Electron Opt.* **2016**, *127*, 428–439. [CrossRef]
24. Zorzi, M.; Chiuso, A. Sparse plus Low rank Network Identification: A Nonparametric Approach. *Automatica* **2017**, *76*, 355-366. [CrossRef]
25. Zorzi, M.; Sepulchre, R. AR Identification of Latent-Variable Graphical Models. *Trans. Autom. Control* **2016**, *61*, 2327–2340. [CrossRef]
26. Jiang, T.X.; Huang, T.Z.; Zhao, X.L.; Deng, L.J.; Wang, Y. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Trans. Image Process.* **2019**, *28*, 2089–2102. [CrossRef] [PubMed]
27. Zheng, Y.B.; Huang, T.Z.; Ji, T.Y.; Zhao, X.L.; Jiang, T.X.; Ma, T.H. Low-rank tensor completion via smooth matrix factorization. *Appl. Math. Model.* **2019**, *70*, 677–695. [CrossRef]
28. Dou, H.X.; Huang, T.Z.; Deng, L.J.; Zhao, X.L.; Huang, J. Directional l0 sparse modeling for image stripe noise removal. *Remote Sens.* **2018**, *10*, 361. [CrossRef]
29. Huang, Y.; Cong, H.; Fang, H.; Wang, X. Iteratively reweighted unidirectional variational model for stripe non-uniformity correction. *Infrared Phys. Technol.* **2016**, *75*, 107–116. [CrossRef]
30. Song, Q.; Wang, Y.; Yan, X.; Gu, H. Remote sensing images stripe noise removal by double sparse regulation and region separation. *Remote Sens.* **2018**, *10*, 998. [CrossRef]
31. Zhou, G.; Fang, H.; Lu, C.; Wang, S.; Zuo, Z.; Hu, J. Robust destriping of MODIS and hyperspectral data using a hybrid unidirectional total variation model. *Opt. Int. J. Light Electron Opt.* **2015**, *126*, 838–845. [CrossRef]
32. Chen, Y.; Huang, T.Z.; Zhao, X.L.; Deng, L.J. Hyperspectral image restoration using framelet-regularized low-rank nonnegative matrix factorization. *Appl. Math. Model.* **2018**, *63*, 128–147. [CrossRef]
33. Prasad, S.; Labate, D.; Cui, M.; Zhang, Y. Morphologically Decoupled Structured Sparsity for Rotation-Invariant Hyperspectral Image Analysis. *IEEE Trans. Geosci. Remote Sens.* **2017**, *99*, 1–12. [CrossRef]
34. Wang, Y.; Peng, J.J.; Leung, Y.; Zhao, X.L.; Meng, D.Y. Hyperspectral image restoration via total variation regularized low-rank tensor decomposition. *IEEE J. STARS* **2018**, *11*, 1227–1243. [CrossRef]
35. Chang, Y.; Yan, L.; Fang, H.; Luo, C. Anisotropic spectral-spatial total variation model for multispectral remote sensing image destriping. *IEEE Trans. Image Process.* **2015**, *24*, 1852–1866. [CrossRef] [PubMed]
36. Lu, X.; Wang, Y.; Yuan, Y. Graph-regularized low-rank representation for destriping of hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4009–4018. [CrossRef]
37. Qi, G.; Zhu, Z.; Chen, Y.; Wang, J.; Zhang, Q.; Zeng, F. Morphology-based visible-infrared image fusion framework for smart city. *Int. J. Simul. Process Model.* **2018**, *13*, 523–536. [CrossRef]

38. Chang, Y.; Yan, L.; Wu, T.; Sheng, Z. Remote sensing image stripe noise removal: From image decomposition perspective. *IEEE Trans. Geosci. Remote Sens.* **2016**, 54, 7018–7031. [CrossRef]
39. Liu, X.; Lu, X.; Shen, H.; Yuan, Q.; Jiao, Y.; Zhang, L. Stripe noise separation and removal in remote sensing images by consideration of the global sparsity and local variational properties. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 3049–3060. [CrossRef]
40. Chen, Y.; Huang, T.Z.; Deng, L.J.; Zhao, X.L.; Wang, M. Group sparsity based regularization model for remote sensing image stripe noise removal. *Neurocomputing* **2017**, *267*, 95–106. [CrossRef]
41. Fan, Y.R.; Wang, Y.; Huang, T.Z. Enhanced joint sparsity via iterative support detection. *Inf. Sci.* **2017**, *415–416*, 298–318. [CrossRef]
42. Wang, Y.; Yin, W. Sparse Signal Reconstruction via Iterative Support Detection. *SIAM J. Imaging Sci.* **2010**, *3*, 462–491. [CrossRef]
43. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **2010**, 3, 1–122. [CrossRef]
44. Mei, J.J.; Dong, Y.Q.; Huang, T.Z.; Yin, W.T. Cauchy noise removal by nonconvex admm with convergence guarantees. *J. Sci. Comput.* **2017**, *74*, 743–766. [CrossRef]
45. Zhao, X.L.; Wang, W.; Zeng, T.Y.; Huang, T.Z.; Ng, M.K. Total variation structured total least squares method for image restoration. *SIAM J. Sci. Comput.* **2013**, *35*, B1304–B1320. [CrossRef]
46. Zuo, W.; Meng, D.; Zhang, L.; Feng, X. A generalized iterated shrinkage algorithm for non-convex sparse coding. In *IEEE Int. Conf. Comput. Vis.* **2013**, 217–224. [CrossRef]
47. Wang, Y.T.; Zhao, X.L.; Jiang, T.X.; Deng, L.J.; Zhang, Y.T. A total variation and group sparsity based tensor optimization model for video rain streak removal. *Signal Process. Image Commun.* **2018**. [CrossRef]
48. Eckstein, J.; Bertsekas, D.P. On the douglas-rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Programm.* **1992**, *55*, 293–318. [CrossRef]
49. Glowinski, R. *Lectures on Numerical Methods for Nonlinear Variational Problems*; Springer: Berlin/Heidelberg, Germany, 1980.
50. Donoho, D. De-noising by soft-thresholding. *IEEE Trans. Inf. Theory* **1995**, *41*, 613–627. [CrossRef]
51. Donoho, D.; Johnstone, I. Adapting to unknown smoothness via Wavelet Shrinkage. *Publ. Am. Stat. Assoc.* **1995**, *90*, 1200–1224. [CrossRef]
52. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]