A General Destriping Framework for Remote Sensing Images Using Flatness Constraint

Kazuki Naganuma®, Student Member, IEEE, and Shunsuke Ono®, Member, IEEE

Abstract—Removing stripe noise, i.e., destriping, from remote sensing images is an essential task in terms of visual quality and subsequent processing. Most existing destriping methods are designed by combining a particular image regularization with a stripe noise characterization that cooperates with the regularization, which precludes us to examine and activate different regularizations to adapt to various target images. To resolve this, two requirements need to be considered: a general framework that can handle a variety of image regularizations in destriping, and a strong stripe noise characterization that can consistently capture the nature of stripe noise, regardless of the choice of image regularization. To this end, this article proposes a general destriping framework using a newly introduced stripe noise characterization, named flatness constraint (FC), where we can handle various regularization functions in a unified manner. Specifically, we formulate the destriping problem as a nonsmooth convex optimization problem involving a general form of image regularization and the FC. The constraint mathematically models that the intensity of each stripe is constant along one direction, resulting in a strong characterization of stripe noise. For solving the optimization problem, we also develop an efficient algorithm based on a diagonally preconditioned primal-dual splitting algorithm (DP-PDS), which can automatically adjust the step sizes. The effectiveness of our framework is demonstrated through destriping experiments, where we comprehensively compare combinations of a variety of image regularizations and stripe noise characterizations using hyperspectral images (HSIs) and infrared (IR) videos.

Index Terms—Destriping, flatness constraint (FC), hyperspectral images (HSIs), infrared (IR) data, primal-dual splitting.

I. INTRODUCTION

REMOTE sensing images, such as hyperspectral images (HSIs) and infrared (IR) videos, offer various applications, including mineral detection, Earth observation, agriculture, astronomical imaging, automatic target recognition, and video surveillance [1]–[3]. Such data, however, are often contaminated by stripe noise, which is mainly due to differences in the nonuniform response of individual detectors, calibration error, and dark currents [4]–[6]. Stripe noise not only degrades visual quality but also seriously affects subsequent processing, such as hyperspectral unmixing [1], [7], HSI classification [8]–[11], and IR video target recognition [12]. Therefore, stripe noise removal, i.e., destriping, has been an important research topic in remote sensing and related fields.

In the past decades, a large number of destriping methods have been proposed. Filtering-based approaches are widely used due to their simplicity [13]–[15]. They effectively remove periodic stripe noise by truncating the specific stripe components in a Fourier or wavelet data domain. However, these approaches are limited in use since they assume that stripe noise is periodic and can be identified from the power spectrum. Deep learning-based approaches have also been studied [16]–[20]. They can automatically extract the nature of desirable data to remove stripe noise by learned neural networks but have difficulties, such as domain dependence, a lack of a learning dataset and excessive removal of image structures (e.g., textures and singular features) [21], [22].

Among many destriping techniques, optimization-based approaches have received much attention. In these approaches, desirable data and stripe noise are modeled by functions that capture their nature, and then, both are simultaneously estimated by solving an optimization problem involving the functions. These approaches adopt some form of regularization to characterize desirable data, including piecewise smoothness [23]–[28], low-rankness [29]–[33], self-similarity [34], sparse representation [35], [36], and combinations of these regularizations [37], [38].

The characterization of stripe noise is as essential as image regularization in destriping. Existing stripe noise characterizations can be roughly classified into a sparsity-based model [29], [30], [32], [33], [39], a low-rank-based model [40], [41], and a total variation (TV) model [42]–[44]. The first model relies on the fact that the stripe noise in observed data is (group) sparsely distributed. The second model characterizes stripe noise as low rank since stripe noise has a strong low-rank structure [40]. The third model captures the vertical (or horizontal) smoothness of stripe noise using TV regularization.

Many of the existing destriping methods are designed by combining a particular image regularization with a stripe noise characterization that cooperates with the regularization. Since the function used for image regularization is often also used for stripe noise characterization, these methods carefully select the function used for stripe noise characterization so that it does not conflict with the adopted image regularization.
For example, destriping methods using the low-rank-based model employ TV as the image regularization [40], but, in the case of destriping methods with the TV model, only the horizontal TV is used to regularize the image [43], [44] because the vertical TV is used to characterize the stripe noise.

On the other hand, it would be very beneficial to establish a destriping framework that can handle various image regularizations in a unified manner so that we can select a regularization that matches each target image of different nature. In fact, a number of image regularization techniques have been proposed for remote sensing images. Typical examples are HSI regularization techniques based on spatiotemporal smoothness and correlation [23], [24], [29], [30], [33]. In the case of video data, there are also many regularization techniques that consider moving objects [45]–[47]. Combining multiple regularizations is also a promising strategy [38], [48].

In order to achieve the aforementioned unified framework, two requirements need to be considered: 1) a general formulation and algorithm that can handle a variety of image regularizations and 2) a strong stripe noise characterization that can consistently capture the nature of stripe noise, regardless of the choice of image regularization.

Based on the above discussion, this article proposes a general destriping framework for remote sensing images. First, we formulate destriping as a constrained convex optimization problem involving a general form of image regularization and a newly introduced strong stripe noise characterization. Second, we develop an efficient algorithm based on the diagonally preconditioned primal-dual splitting algorithm (DP-PDS) [37]–[39], which can automatically determine the appropriate step sizes for solving this problem.

The main contributions of this article are given as follows.

1) **General Framework:** Our framework incorporates image regularization as a general form represented by a sum of (possibly) nonsmooth convex functions involving linear operators. This enables us to leverage various image regularizations according to target images.

2) **Effective Characterization of Stripe Noise:** The most common type of stripe noise has a strong flat structure in the vertical or horizontal direction. As a typical example, a band of a raw HSI, a frame of a raw IR video, and their vertical and horizontal gradients are shown in Fig. 1, where we can see that the stripe component only exists in the horizontal differences. This implies that stripe noise is flat in the vertical direction. Therefore, we can capture the flatness by constraining its vertical gradient to zero, named the flatness constraint (FC). Moreover, stripe noise in videos is often time-invariant. For example, IR videos are corrupted with time-invariant stripe noise due to focal plane arrays [49], [50]. Some frames of a raw IR video and their differences are shown in Fig. 2, where we can see that the stripe noise is time-invariant because it does not appear in the differences. For such data, we impose the FC along the temporal direction in addition to the spatial constraint. Due to such a strong characterization, our framework has a marked ability of stripe noise removal that does not so much depend on what image regularization is adopted.

3) **Automatic Step Size Adjustment:** Our algorithm can automatically adjust the step sizes based on the structure of the optimization problem to be solved. In general, the appropriate step sizes of PDS would be different depending on image regularizations, meaning that we have to manually adjust them many times. Our algorithm is free from such a troublesome task.

We demonstrate the effectiveness of our framework through destriping experiments, where we comprehensively compare combinations of image regularizations and stripe noise characterizations using HSIs and IR videos.

The remainder of this article is organized as follows. The mathematical notations are summarized in Table I. For more detailed and visual understandings of tensor operators, [51], [52] are helpful. Section II gives reviews of the existing sparsity-, low-rank-, and TV-based destriping models. Section III presents the details of the proposed formulation and the solver. Experimental results and discussion are given in Section IV. Finally, we summarize this article in Section V.

The preliminary version of this work, without mathematical details, comprehensive experimental comparison, deeper
TABLE I  
NOTATIONS AND DEFINITIONS

| Line number | Notation | Terminology |
|-------------|----------|-------------|
| 1           | \( \mathbb{R} \) and \( \mathbb{R}_+ \) | Real and positive real numbers |
| 2           | \( \prod_{i=1}^{M} \mathbb{R}^{n_i} \) | \( M \) th-order tensor/positive-element-tensor product space \(^1\) |
| 3           | \( X \), \( (X_1, \ldots, X_M) \) | Elements of tensor product space |
| 4           | \( X_i(i_1, \ldots, i_{N_i}) \) or \( [X_i]_{i_1, \ldots, i_{N_i}} \) | \( (i_1, \ldots, i_{N_i}) \) th element of an ith tensor of \( X \) |
| 5           | \( \|X\|_{1} \), \( (X, Y) \) | \( \ell_1 \)-norm, \( \|X\|_1 = \sum_{i_1, \ldots, i_{N_i}} |X_i(i_1, \ldots, i_{M})| \) |
| 6           | \( X = (X_1, \ldots, X_M) \) in \( \prod_{i=1}^{M} \mathbb{R}^{n_i} \), \( Y = (Y_1, \ldots, Y_M) \) in \( \prod_{i=1}^{M} \mathbb{R}^{n_i} \) | Inner product, \( (X, Y) = \sum_{i_1, \ldots, i_{N_i}} X_i(i_1, \ldots, i_{M}) Y_i(i_1, \ldots, i_{N_i}) \) |
| 7           | \( \|X\|_{F} = \sqrt{(X, X)} \) | Frobenius norm, \( \|X\|_F = \sqrt{(X, X)} \) |
| 8           | \( X \circ Y \) in \( \prod_{i=1}^{M} \mathbb{R}^{n_i} \) | Hadamard product, \( Z_{i}(i_1, \ldots, i_{N_i}) = X_i(i_1, \ldots, i_{N_i}) Y_i(i_1, \ldots, i_{N_i}) \), \( Z = X \circ Y \), \( \forall k \in \{1, \ldots, n_i, N_i \} \), \( \forall i \in \{1, \ldots, M \} \), \( \forall k \in \{1, \ldots, M \} \) |
| 9           | \( I = (I_1, \ldots, I_M) \) | Identity tensor product element with the Hadamard product, \( Z_{i}(i_1, \ldots, i_{N_i}) = 1 \), \( \forall k \in \{1, \ldots, n_i, N_i \} \), \( \forall i \in \{1, \ldots, M \} \), \( \forall k \in \{1, \ldots, M \} \) |
| 10          | \( G^{-1} = (G_1^{-1}, \ldots, G_M^{-1}) \) | Inverse tensor product element of \( G \) with the Hadamard product, \( G \circ G^{-1} = I \) |
| 11          | \( \|X\|_{F,G} = \sqrt{(X, G^{-1} X)} \) | Proximity norm skewed by the metric induced by \( G \), \( \|X\|_{F,G} = \sqrt{(X \circ G^{-1} X)} \) |
| 12          | \( X \) in \( \prod_{i=1}^{M} \mathbb{R}^{n_i} \), \( G \) in \( \prod_{i=1}^{M} \mathbb{R}^{n_i} \) | \( \text{prox}_{G^{-1} f} (X) \), \( f \) is a proper lower semi-continuous convex function |

\(^1\text{If } M = 1, \text{ a tensor product space is equivalent to a tensor space.}\)

discussion, or implementation using DP-PDS, has appeared in conference proceedings [53].

II. REVIEW OF EXISTING APPROACHES

HSI and IR video data can be represented as third-order tensors, where the spatial information lies in the first two dimensions, and the spectral or frame information lies in the third dimension. To estimate desirable data from the observed data contaminated by stripe noise and random noise, we model the observation data as follows:

\[ V = \bar{U} + S + N, \]  

(1)

where \( \bar{U} \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) is a desirable data of interest, \( S \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) is the stripe noise, \( N \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) is the random noise, and \( V \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) is the observed data.

Under the model in (1), the destriping problem is often formulated as convex optimization problems with the following form:

\[ \min_{U \in \mathbb{R}^{n_1 \times n_2 \times n_3}} \sum_{k=1}^{K} R_k(L_k(U)) + \lambda_S J(S) + \lambda_N \left\| V - (U + S) \right\|_F^2, \]

where \( R_k(L_k(\cdot)) : \mathbb{R}^{n_1 \times n_2 \times n_3} \rightarrow (-\infty, \infty] \) is regularization functions for imaging data with a linear operation \( L_k \) and a function \( R_k \) (\( k = 1, \ldots, K \)), and \( J : \mathbb{R}^{n_1 \times n_2 \times n_3} \rightarrow (-\infty, \infty] \) is a function characterizing stripe noise, respectively. The positive scalars \( \lambda_S \) and \( \lambda_N \) are the hyperparameters. Depending on how \( J \) is chosen, destriping models can be classified into the following three categories: the (group-)sparsity-based model, the low-rank-based model, and the TV-based model.

The sparsity-based model has been used in a lot of methods. Among them, the method proposed in [29] is known as a representative work. This method uses the \( \ell_1 \)-norm as \( J \), which is a well-known sparsity measure. As mentioned, this model relies on the fact that stripe noise is sparsely distributed in observed data. The method proposed in [37] sets \( J \) to the mixed \( \ell_{2,1} \)-norm since each column of stripe noise is viewed
as a group. The mixed \( \ell_{2,1} \)-norm is the sum of the \( \ell_2 \)-norm of each column vector, which groups stripe noise by each column, and thus, it is used for the characterization of the stripe noise based on group sparsity. The sparsity-based model results in efficient optimization due to its simple modeling but cannot fully capture the nature of stripe noise. Specifically, its destriping performance strongly depends on image regularization, as will be shown in Section IV-C.

The low-rank-based model has been proposed in [40]. In [40], the authors revealed that the stripe noise only exists in the horizontal gradient component, and the rank of stripe noise is one. Based on this observation, they adopted the nuclear norm for \( J \), which is a reasonable convex function that can evaluate the low-rankness of a matrix. In general, this model outperforms the sparsity-based model. However, it conflicts with low-rank image regularizations where the nuclear norm is employed [29]–[31], [33].

The TV-based model [43], [44] adopted a TV term and a sparse term to capture the one-directional smoothness of stripe noise. This model is also superior to the sparsity-based model. However, the TV-based model weakens the TV regularization ability to capture the vertical smoothness, as will be shown in Section IV-C.

We summarize the stripe noise characterizations in Table II.

### III. PROPOSED FRAMEWORK

The proposed framework involves a general form of regularization term and two types of FC. The choice of the specific image regularization and the removal of the temporal FC are required to fit the nature of an observed image. Depending on image regularization and the temporal FC, the DP-PDS-based solver needs to be implemented. We illustrate a whole workflow for the proposed framework in Fig. 3.

#### A. General Destriping Model With Flatness Constraint

In this section, we propose a general destriping model using the FC. As mentioned, stripe noise \( S \) has the characteristic that the vertical/temporal gradient is zero, that is,

\[
\begin{align*}
\mathcal{D}_v(S) &= \mathcal{O}, \\
\mathcal{D}_t(S) &= \mathcal{O},
\end{align*}
\]

(2)

where \( \mathcal{O} \in \mathbb{R}^{n_1 \times n_2 \times n_3} \) is a zero tensor, i.e., \( \mathcal{O}(i, j, k) = 0, \forall i \in \{1, \ldots, n_1\}, \forall j \in \{1, \ldots, n_2\}, \text{ and } \forall k \in \{1, \ldots, n_3\} \). Moreover, \( \mathcal{D}_v: \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{(n_1 - 1) \times n_2 \times n_3} \) and \( \mathcal{D}_t: \mathbb{R}^{n_1 \times n_2 \times n_3} \to \mathbb{R}^{n_1 \times (n_2 - 1) \times n_3} \).

Using the FCs in (2), we newly formulate destriping as the following convex optimization problem:

\[
\begin{align*}
\min_{\mathcal{U}, \mathcal{S}} \sum_{k=1}^{K} R_k(\mathcal{L}_k(\mathcal{U})) + \lambda \|S\|_1 \\
\text{s.t.} \quad \mathcal{D}_v(S) &= \mathcal{O}, \\
\mathcal{D}_t(S) &= \mathcal{O}, \\
\|V - (\mathcal{U} + S)\|_F &\leq \varepsilon,
\end{align*}
\]

(5)

where \( \lambda > 0 \) is a hyperparameter and \( R_k(\mathcal{L}_k(\cdot)) \) is a regularization term with a proper semicontinuous convex proximable function \( R_k \) and a linear operator \( \mathcal{L}_k \). The vertical and temporal gradients of stripe noise are constrained to zero by the first and second constraints, which captures the vertical/temporal flatness of the stripe noise. In addition, we impose the \( \ell_1 \)-norm on \( S \) to exploit the sparsity of the stripe noise. The third constraint is a Frobenius norm constraint with the radius \( \varepsilon \) for data fidelity to \( V \) given in (1). The data-fidelity constraint has an important advantage over the standard additive data fidelity in terms of facilitating hyperparameter settings, as addressed in [54]–[58]. If stripe

\[\|r_{n_1 \times n_2 \times (n_1 - 1)} \] are the vertical/temporal difference operators with the Neumann boundary, which are defined as

\[
\begin{align*}
[\mathcal{D}_v(X)]_{i,j,k} := X(i, j, k) - X(i + 1, j, k), \\
&\quad \forall i \in \{1, \ldots, n_1 - 1\}, \\
&\quad \forall j \in \{1, \ldots, n_2\},
\end{align*}
\]

(3)

\[
\begin{align*}
[\mathcal{D}_t(X)]_{i,j,k} := X(i, j, k) - X(i, j, k + 1), \\
&\quad \forall i \in \{1, \ldots, n_1\}, \\
&\quad \forall j \in \{1, \ldots, n_2\},
\end{align*}
\]

(4)

If an efficient computation of the skewed proximity operator of \( f \) is available, we call \( f \) skew proximable.

![Proposed general destriping framework](image)

**TABLE II**

| Model                  | \( J(S) \)                                      |
|------------------------|------------------------------------------------|
| Sparsity-based model   | \( \lambda \|S\|_1 \)                           |
| Group sparsity-based model | \( \lambda \sum_{i=1}^{n_1} \|S(:,j,k)\|_2 \) |
| Low-rank-based model   | \( \lambda \sum_{i=1}^{n_3} \|S(:,:,i)\|_2 \) |
| TV-based model         | \( \mu \|\mathcal{D}_v(S)\|_{\text{Frobenius}} + \lambda \|S\|_1 \) |

![Whole workflow of the proposed general destriping framework](image)
noise is variant in the third direction, such as HSIs, we remove the second constraint.

For data with horizontally featured stripe noise, as in images acquired by whiskbroom scanning [43], we rotate the data $90^\circ$ in the spatial direction before optimization.

B. Diagonally Preconditioned Primal-Dual Splitting Algorithm for Solving the General Destriping Model

In this part, we introduce DP-PDS [59] to solve Problem (5). DP-PDS (see Appendix), which is a diagonally preconditioned version of the PDS [60], [61], frees us from tedious step size settings. Moreover, the convergence speed of DP-PDS is much faster in general than that of the original PDS algorithm.

To solve Problem (5) with DP-PDS, we rewrite it into the following equivalent problem:

$$\min_{U,S,Y_1,Y_2,Y_3} \frac{1}{2} \| S \|_1 + \sum_{k=1}^K R_k(Y_{1,k}) + t_{I(0)}(Y_2)$$

subject to:

$$Y_{1,1} = \mathcal{L}_1(U),$$
$$Y_{1,k} = \mathcal{L}_k(U),$$
$$Y_{2} = \mathcal{D}_b(S),$$
$$Y_{3} = \mathcal{D}_i(S),$$
$$Y_{4} = U + S,$$

where $t_{I(0)}$ and $R_{k,Y_{2,v}}$ are the indicator functions of $\{0\}$ and $B_{Y_{2,v}} := \{ X \in \mathbb{R}^{n_1 \times n_2 \times n_3} \mid \| Y - X \|_F \leq \epsilon \}$, respectively. DP-PDS computes the solution of (6) by updating primal variables $(U$ and $S$) and dual variables $(Y_{1,1}, \ldots, Y_{1,k}, Y_{2}, Y_{3},$ and $Y_{4})$ alternately.

The primal variables are updated as follows:

$$U^{(n+1)} \leftarrow U^{(n)} - \mathcal{G}_U \odot \left( \sum_{k=1}^K \mathcal{L}_k(N_{1,k} + Y_4) \right),$$

$$S^{(n+1)} \leftarrow \text{prox}_{G_3^{(1)}} \left[ \left( S^{(n)} - \mathcal{G}_S \odot \left( \mathcal{D}_b^*(\mathcal{Y}_{2}^{(n)}) + \mathcal{D}_i^*(\mathcal{Y}_{3}^{(n)}) + \mathcal{Y}_{4}^{(n)} \right) \right) \right],$$

where $\mathcal{L}_1$, $\mathcal{L}_k$, $\mathcal{D}_b$, and $\mathcal{D}_i$ are the adjoint operators of $\mathcal{L}_1$, $\mathcal{L}_k$, $\mathcal{D}_b$, and $\mathcal{D}_i$, respectively. The constants $\mathcal{G}_U$ and $\mathcal{G}_S$ are step size parameters that are called preconditioners, and $G_3^{(1)}$ is the inverse tensor of $G_3$ (see line 10 of Table I). The preconditioners are given by the coefficients of the linear operations $\mathcal{L}$ and $\mathcal{D}_b$ (see (34) in Appendix for the detailed definitions). The skewed proximity operator (see line 12 of Table I for the definition) of $\| \cdot \|_1$ in (8) is given by

$$\text{prox}_{G_3^{(1)}}(X) = \text{sign}(X) \circ \max\{ |X| - \lambda G_S, 0 \},$$

where $\text{sign}(X)$, $\max\{X, 0\}$, and $|X|$, respectively, denote the sign, positive part, and magnitude of $X$. Their definitions are given as follows:

$$\text{[sign}(X)]_{i,j,k} = \begin{cases} 1, & \text{if } X(i,j,k) \geq 0, \\ -1, & \text{if } X(i,j,k) < 0, \end{cases}$$

$$\text{[max}(X, 0)]_{i,j,k} = \begin{cases} 0, & \text{if } X(i,j,k) < 0, \\ x, & \text{if } X(i,j,k) \geq 0, \end{cases}$$

Then, the dual variables are updated as follows:

$$Y_{1,k}^{(n+1)} \leftarrow \text{prox}_{G_3^{(1)}}(Y_1^{(n)} + \mathcal{G}_y Y_{1,k} \odot (\mathcal{L}_k(2U^{(n+1)} - U^{(n)}))),$$

$$Y_2^{(n+1)} \leftarrow \text{prox}_{G_3^{(1)}}(Y_2^{(n)} + \mathcal{G}_y Y_{2} \odot (2S^{(n+1)} - S^{(n)})),$$

where $\mathcal{G}_y Y_{1,k}$ and $\mathcal{G}_y Y_{2}$ are the Fenchel–Rockafellar conjugate functions of $R_k$ and $R_{k,Y_{2,v}}$, respectively.

Algorithm 1 DP-PDS Algorithm for Solving Problem (5)

Input: An observed image $V$, a balancing parameter $\lambda$, and a data fidelity parameter $\varepsilon$

Output: $U^{(n)}$, $S^{(n)}$

1. Initialize $U^{(0)}$, $S^{(0)}$, $Y_{1,i}^{(0)}$ ($i = 1, 2, 3, 4$);
2. $n = 0$;
3. while A stopping criterion is not satisfied do
4. $U^{(n+1)} \leftarrow U^{(n)} - G_{U} \odot \left( \sum_{k=1}^{K} L_k(N_{1,k} + Y_4) \right)$;
5. $S^{(n+1)} \leftarrow S^{(n)} - G_{S} \odot (D_b^*(Y_2^{(n)}) + D_i^*(Y_3^{(n)}) + Y_4^{(n)})$;
6. $S^{(n+1)} \leftarrow \text{prox}_{G_3^{(1)}}(S^{(n)})$ by (9);
7. for $i = 1, \ldots, K$ do
8. $Y_{1,i}^{(n+1)} \leftarrow Y_{1,i}^{(n)} + \mathcal{G}_y Y_{1,k} \odot (2U^{(n+1)} - U^{(n)})$;
9. $Y_{1,i}^{(n+1)} \leftarrow Y_{1,i}^{(n)} - \mathcal{G}_y Y_{1,k} \odot \text{prox}_{G_3^{(1)}}(Y_1^{(n)} + Y_4^{(n)})$;
10. end for
11. $Y_2^{(n+1)} \leftarrow Y_2^{(n)} + \mathcal{G}_y Y_{2} \odot (2S^{(n+1)} - S^{(n)})$;
12. $Y_3^{(n+1)} \leftarrow Y_3^{(n)} + \mathcal{G}_y Y_{3} \odot (2S^{(n+1)} - S^{(n)})$;
13. $Y_4^{(n+1)} \leftarrow Y_4^{(n)} + \mathcal{G}_y Y_{4} \odot (2(L^{(n+1)} - L^{(n)}) - S^{(n)})$;
14. $Y_4^{(n+1)} \leftarrow \text{prox}_{G_3^{(1)}}(Y_4^{(n)} + Y_4^{(n)})$ by (18);
15. $n \leftarrow n + 1$;
16. end while
The skewed proximity operator has the following useful property [62, Corollary 6]:
\[
\text{prox}_{g^{-1},f}(X) = X - G \odot \text{prox}_{g,f}(G^{-1} \odot X),
\]
(17)
so that the skewed proximity operator of a Fenchel–Rockafellar conjugate function \(f^*\) can be easily calculated if \(f\) is skew proximizable. The skewed proximity operators in (13) are efficiently computed because \(R_k\) is a skew proximable function. The skewed proximity operator of \(t_{\psi(V)}\) in (14) and (15) is calculated as \(\text{prox}_{g_{t_{\psi(V)}}}(X) = O\) for any \(X \in \mathbb{R}^{(n-1) \times (n+1) \times n^2}\) and \(G \in \mathbb{R}^{(n+1) \times (n+1) \times n^2}\). The skewed proximity operator of \(t_{\psi(V)}\) in (16) is not proximable in general. In our method, all entries of the preconditioner \(G\) are \((1/2)\). Hence, the operator \(\text{prox}_{g_{t_{\psi(V)}}}\) is easily calculated as
\[
\begin{align*}
\text{prox}_{g_{t_{\psi(V)}}}(X) &= \text{prox}_{X,2t_{\psi(V)}}(X) \\
&= \mathcal{P}_{t_{\psi(V)}}(X) = \begin{cases} 
X, & \text{if } X \in B_{\psi(V)}, \\
V + \frac{\epsilon(X - V)}{\|X - V\|_F}, & \text{otherwise},
\end{cases}
\end{align*}
\]
(18)
Through these update steps, we obtain the solution of Problem (5). We show the detailed algorithms in Algorithm 1. We note that this algorithm can handle a nonconvex optimization problem that contains the proximal nonconvex function, such as the \(\ell_0\)-norm and the rank function. However, its convergence, in this case, is not guaranteed.

In temporally variant stripe noise cases, such as an HSI, the temporal constraint is removed. Following the change, the update step in (8) will be given as follows:
\[
S^{(n+1)} \leftarrow \text{prox}_{g_{s_{\psi(V)}}} \left( S^{(n)} - G_S \odot \left( D_v \left( Y_2^{(n)} \right) + Y_4^{(n)} \right) \right).
\]
(19)
Then, we remove the update step of \(Y_3\) (line 9 of Algorithm 1).

### C. Examples of Image Regularizations

We give some examples of image regularization \(\sum_{k=1}^{K} R_k(\mathcal{L}_k(\mathcal{U}))\) in (5). First, let us consider hyperspectral total variation (HTV) [23]. Since the HTV is an image regularization for HSIs, we adopt the formulation that does not involve the temporal FC. The definition of HTV is
\[
\|\mathcal{U}\|_{\text{HTV}} := \sum_{i,j} \sqrt{D_1(i,j,k)^2 + D_2(i,j,k)^2},
\]
(20)
where \(D_1 = \mathcal{D}_v(\mathcal{U})\) and \(D_2 = \mathcal{D}_h(\mathcal{U})\). Therefore, by letting \(K = 1, \mathcal{L}_1(\mathcal{U}) = (\mathcal{D}_v(\mathcal{U}), \mathcal{D}_h(\mathcal{U}))\), and \(R_1 = \|\mathcal{U}_1, \mathcal{U}_2\|_{1,2} = \sum_{i,j} \left( \mathcal{D}_1(i,j,k)^2 + \mathcal{U}_1(i,j,k)^2 \right)^{1/2}\), we can apply HTV to Problem (5). The update of \(\mathcal{U}\) is given as follows:
\[
\begin{align*}
\mathcal{U}^{(n+1)} &\leftarrow \mathcal{U}^{(n)} - H_T \odot \left( \mathcal{D}_v^{-1} \left( Y_1^{(n)} \right) + \mathcal{D}_h^{-1} \left( Y_2^{(n)} \right) \right), \\
&= \mathcal{U}^{(n)} - H_T \odot \left( \mathcal{D}_v^{-1} \left( Y_1^{(n)} \right) + \mathcal{D}_h^{-1} \left( Y_2^{(n)} \right) \right),
\end{align*}
\]
(21)
where \(Y_1^{(n)} = (Y_1^{(n)}), Y_2^{(n)} = (Y_2^{(n)}), Y_3^{(n)} = (Y_3^{(n)}), Y_4^{(n)} = (Y_4^{(n)})\). The proximity operator of \(\|\cdot\|_{1,2}\) is calculated as follows:
\[
Z_1(i,j,k) = \max \left\{ 1 - \frac{\mathcal{G}_2 Y_1(i,j,k)}{\sqrt{\mathcal{S}_1(i,j,k)^2 + Y_1(i,j,k)^2}}, 0 \right\},
\]
(22)
where \(Z_1, Z_2 = \text{prox}_{g_2(\|\cdot\|_{1,2}), Y_2}\). Preconditioners are determined as \(\mathcal{G}_2(i,j,k) = 1/(G_2(i,j,k)) + G_2^{-1}(i,j,k) + 1\) and \(\mathcal{G}_3(i,j,k) = 1/(G_3(i,j,k) + 1)\), where
\[
\begin{align*}
G_2^{-1}(i,j,k) &= \begin{cases} 1, & \text{if } i = 1, n_1, \\
2, & \text{otherwise},
\end{cases} \\
G_3^{-1}(i,j,k) &= \begin{cases} 1, & \text{if } j = 1, n_2, \\
2, & \text{otherwise},
\end{cases}
\end{align*}
\]
(23)
and \(G_2 Y_1(i,j,k) = 1/2, G_2 Y_2(i,j,k) = 1/2, \forall I \in \{1, \ldots, n_1\}, \forall J \in \{1, \ldots, n_2\}, \forall K \in \{1, \ldots, n_3\}\). Finally, we obtain a solver for Problem (5) with HTV.

As another example for an IR video case, we consider ATV [28]. ATV is defined as
\[
\|\mathcal{U}\|_{\text{ATV}} := \|\mathcal{D}_v(\mathcal{U})\|_1 + \|\mathcal{D}_h(\mathcal{U})\|_1 + \|\mathcal{D}_v(\mathcal{U})\|_1.
\]
(25)

TABLE III

| Characterization of stripe noise | Image regularization | HTV (HSI) | SSTV (HSI) | ATSTV (HSI) | TNN (HSI) | SSTV+TNN (HSI) | \(k_{\psi-1}\)HTV (HSI) |
|--------------------------------|----------------------|----------|------------|------------|-----------|----------------|------------------|
| S [29] | GS [37] | LR [40] | TV [44] | PC | Ours |
| HTV (IR video) | [28] | None | None | None | None | None | None |
| ATV (IR video) | [28] | None | None | None | None | None | None |
| ATV+N (IR video) | [48] | None | None | None | None | None | None |

Fig. 5. Salinas destriping result of \(S^{(n)}\) in each iteration with HTV (R: 140, G: 101, B: 30).
The proximity operator in line 11 of Algorithm 1 is calculated by (9). Preconditioners are set as $G_{D_1}(i, j, k) = 1/(G_{D_1}(i, j, k) + G_{D_2}(i, j, k) + 1)$ and $G_{S}(i, j, k) = 1/(G_{D_1}(i, j, k) + G_{D_2}^{-1}(i, j, k) + 1)$, where $G_{D_1}$ and $G_{D_2}$ are already defined in the HTV example and

$$G_{D_1}(i, j, k) = \begin{cases} 1, & \text{if } k = 1, n_3, \\ 2, & \text{otherwise}, \end{cases} \quad G_{D_2}(i, j, k) = \begin{cases} 2, & \text{if } k = 1, n_3, \\ 1/2, & \text{otherwise}. \end{cases}$$

The complexities of lines 4, 8, and 9 of Algorithm 1 depend on what image regularization is adopted. When a specific image regularization is not given, we cannot have explicit complexities. All operations of lines 5, 6, and 11–14 of Algorithm 1 have the complexity of $O(n_1 n_2 n_3)$. Thus, the complexity for each iteration of the algorithm is larger of $O(n_1 n_2 n_3)$ or the one for the image regularization term.

We measured the actual running times using MATLAB (R2021a) on a Windows 10 computer with an Intel Core i9-10900 3.7-GHz processor, 32 GB of RAM, and NVIDIA GeForce RTX 3090. The actual running times [s] and total iteration numbers were 13.47 and 932, 5.123 and 191 for Moffett Field destriping using HTV, Salinas destriping using HTV, and Bats1 destriping using ATTV, respectively. For the experimental settings, see Section IV-C.

**D. Computational Cost and Running Time**

The complexity of each iteration of the algorithm is larger of $O(n_1 n_2 n_3)$ or the one for the image regularization term.

We measured the actual running times using MATLAB (R2021a) on a Windows 10 computer with an Intel Core i9-10900 3.7-GHz processor, 32 GB of RAM, and NVIDIA GeForce RTX 3090. The actual running times [s] and total iteration numbers were 13.47 and 932, 5.123 and 191 for Moffett Field destriping using HTV, Salinas destriping using HTV, and Bats1 destriping using ATTV, respectively. For the experimental settings, see Section IV-C.

**E. Convergence Analysis**

The convergence property of Algorithm 1 is given in Appendix B. Moreover, we experimentally confirm the convergence properties. We plotted the objective function values $\sum_{k=1}^{K} R_k(\mathcal{U}(n)) + \lambda \|S(n)\|_1$ versus iterations $n$ on the experiments using HTV and ATTV in Fig. 4, where our algorithm minimizes the objective function. Fig. 5 shows Salinas destriping results of $S(n)$ in each iteration. From these results, we can see that the stripe noise becomes flat along the vertical direction as the number of iterations is large. The convergence speed of the stripe noise component depends on what image regularization is adopted.

**IV. EXPERIMENTS**

In this section, we illustrate the effectiveness of our framework through comprehensive experiments. Specifically, these experiments aim to show the following.

1) Our FC accurately separates stripe noise from striped images.
2) Our framework achieves good destriping performance on average, whatever image regularizations are used.
3) Set some parameters, such as the weight of image regularization, the gradient regularization weight $\mu$ of the TV-based model, the data-fidelity parameter $\epsilon$, and the parameter of the sparse term $\lambda$. (Their detailed settings are given in each experimental section.)
4) Conduct destriping experiments using these solvers and parameters.

**A. Image Regularizations and Stripe Noise Characterizations**

In HSI experiments, we adopted HTV [23], spatiotemporal total variation (SSTV) [24], anisotropic spectral–spatial total variation (ASSSTV) [25], tensor nuclear norm (TNN) [30], spatial–spectral total variation with TNN (SSTV+TNN) [38], and spatial–spectral total variation with TNN (SSTV+TNN) [38].

---

The code is available at https://drive.google.com/file/d/1krce7Tv09rkPQ BiosMb6tStMLAmc_UL/view?usp=sharing
and \(l_0-l_1\) hybrid total variation (\(l_0-l_1\)HTV) [27], which are often used for HSI regularization. The parameters of ASSTV were experimentally determined as the values that can achieve the best performance. The parameter of SSTV+TNN was set to the values recommended in [38]. In IR video experiments, we adopted ATV, isotropic total variation (ITV) [28], and ATV with nuclear norm (ATV+NN) [48], which are known as video regularization. We compared the proposed FC with the sparsity-based model (S), the group-sparsity-based model (GS), the low-rank-based model (LR), and the TV-based model (TV). For convenience, we denote each method that combines a particular stripe noise characterization and a particular image regularization by connecting each name with a hyphen. For example, the destriping method using the sparsity-based model and HTV is denoted as S-HTV.

Table III summarizes all combinations of stripe noise characterizations and image regularizations examined in our experiments, where we indicate reference numbers for specific combinations that have been proposed in existing studies (“None” means that the combination has not been considered yet).

### B. Dataset Descriptions

We employed three HSI datasets and two IR datasets for experiments in simulated and real noise cases. All images were normalized between \([0, 1]\).

The **Moffett Field** [63] was acquired by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) over the urban and rural area in Moffett Field, CA, USA, with a spatial resolution of 20 m. This image consists of 224 spectral bands in the range of 400–2500 nm. After removing noisy bands, we used a subimage of size \(395 \times 185 \times 176\) [see Fig. 6(a)] for experiments in simulated noise cases.

The **Salinas** [64] was collected by AVIRIS over the field area in Salinas Valley, CA, USA, with a spatial resolution of 3.7 m. This image consists of 224 spectral bands in the range of 400–2500 nm. After removing noisy bands, we used a subimage of size \(360 \times 217 \times 190\) [see Fig. 8(a)] for experiments in simulated noise cases.

The **Suwannee** [65] acquired by AVIRIS over National Wildlife Reserves in the Gulf of Mexico with a spatial resolution of 2 m. This image consists of 360 spectral bands in the range of 395–2450 nm. We used a subimage of size \(256 \times 256 \times 360\) [see Fig. 13(a)] for experiments in real noise cases.

The **Bats1** and **Bats2** [66], which include hundreds of bats, were collected with three forward looking infrared (FLIR) SC6000 thermal IR cameras at a frame rate of 125 Hz. For more detailed descriptions, see [3], [67], [68]. We used denoised and raw subimages of size \(256 \times 256 \times 50\) [see Figs. 7(a) and 13(b)] for experiments in simulated and real noise cases, respectively.

![Fig. 7. Bats1 destriping results in Case (ii) with ATV. The MPSNR and MSSIM of our FC are highlighted in bold.](image-url)

![Fig. 8. Salinas destriping results in Case (iii) with TNN (R: 140, G: 101, B: 30). The MPSNR and MSSIM of our FC are highlighted in bold.](image-url)
TABLE IV

| Image data | Range of | MPSNR | MSSIM |
|------------|----------|-------|-------|
|            | stripe noise |      |       |
|            | function    | S [29] | GS [37] | LR [40] | TV [44] | FC |
| HSI        |            |       |       |       |       |    |
| HSI+HTV    | 28.70      | 32.00 | 38.29 | 37.65 | 37.35 | 0.7601 | 0.8980 | 0.9835 | 0.9937 | 0.9929 |
| HSI+HTV   | 38.69      | 39.92 | 41.64 | 39.29 | 39.29 | 0.9344 | 0.9411 | 0.9628 | 0.9266 | 0.9787 |
| HSI+TNN    | 32.95      | 34.54 | 39.04 | 37.58 | 37.61 | 0.9285 | 0.9276 | 0.9889 | 0.9735 | 0.9799 |
| l_{o1}=HTV | 36.66      | 35.96 | 41.53 | 38.52 | 39.92 | 0.9515 | 0.9429 | 0.9877 | 0.9655 | 0.9837 |
|            |            |       |       |       |       |    |
|            |           |       |       |       |       |    |
|            |           |       |       |       |       |    |
|            |           |       |       |       |       |    |
|            |           |       |       |       |       |    |

C. Experiments in Simulated Noise Cases

For the HSI destriping experiments, the parameter \( \lambda \) of each stripe noise characterization model summarized in Table II was set to a hand-optimized value, so as to achieve the best mean peak signal-to-noise ratio (MPSNR). For fair comparison, we set \( \varepsilon \) to the oracle value, i.e., \( \varepsilon = \|N\|_F \). As quantitative evaluations, we employed the MPSNR

\[
\text{MPSNR} = \frac{1}{N_3} \sum_{k=1}^{n_3} 10 \log_{10} \frac{1}{\|U_k - \bar{U}_k\|^2}. \tag{28}
\]

and the mean structural similarity overall bands (MSSIM) [69]

\[
\text{MSSIM} = \frac{1}{N} \sum_{k=1}^{N} \text{SSIM}(U_k, \bar{U}_k), \tag{29}
\]

where \( U_k \) is the \( k \)-th band of \( U \). The larger these values are, the better the destriping results are. The stopping criterion of Algorithm 1 was set as \( \|U_k(n+1) - U_k(n)\|_F / \|U_k(n)\|_F < 1.0 \times 10^{-4} \). We generated the three types of degraded images:

1) HSIs with vertical stripe noise.
2) HSIs with horizontal stripe noise.
3) HSIs with diagonal stripe noise.
TABLE V

| IR video data | Range of stripe noise | Regularization   | MPSNR | MSSIM | MPSNR | MSSIM |
|---------------|-----------------------|------------------|-------|-------|-------|-------|
|               |                       | function        | S [29] | GS [37] | LR [40] | TV [44] | FC | S [29] | GS [37] | LR [40] | TV [44] | FC |
| Moffett field | [−0.2, 0.2]           | ATV              | 30.15 | 30.48 | 34.45 | 29.97 | 36.53 | 0.9532 | 0.9540 | 0.9955 | 0.9400 | 0.9956 |
|               |                       | ITV              | 30.15 | 30.53 | 34.56 | 29.98 | 36.53 | 0.9532 | 0.9524 | 0.9935 | 0.9414 | 0.9957 |
|               |                       | ATV+NN           | 30.18 | 30.50 | 34.76 | 29.98 | 35.28 | 0.9531 | 0.9530 | 0.9951 | 0.9486 | 0.9951 |
| Bats1         | [−0.25, 0.25]         | ATV              | 30.02 | 30.44 | 32.09 | 29.98 | 36.82 | 0.9526 | 0.9540 | 0.9771 | 0.9399 | 0.9956 |
|               |                       | ITV              | 30.02 | 30.64 | 33.45 | 29.99 | 36.87 | 0.9525 | 0.9540 | 0.9736 | 0.9410 | 0.9959 |
|               |                       | ATV+NN           | 30.06 | 30.64 | 33.87 | 29.99 | 35.36 | 0.9528 | 0.9538 | 0.9950 | 0.9485 | 0.9951 |
|               | [−0.3, 0.3]           | ATV              | 31.89 | 32.36 | 38.67 | 31.76 | 41.94 | 0.9541 | 0.9552 | 0.9945 | 0.9420 | 0.9953 |
|               |                       | ITV              | 31.90 | 32.37 | 38.85 | 31.82 | 42.13 | 0.9539 | 0.9551 | 0.9939 | 0.9430 | 0.9954 |
|               |                       | ATV+NN           | 31.90 | 32.37 | 38.76 | 31.91 | 41.73 | 0.9539 | 0.9551 | 0.9775 | 0.9504 | 0.9953 |
|               | [−0.35, 0.35]         | ATV              | 31.78 | 32.23 | 35.64 | 31.63 | 40.70 | 0.9539 | 0.9548 | 0.9914 | 0.9422 | 0.9955 |
|               |                       | ITV              | 31.78 | 32.37 | 35.43 | 31.68 | 40.54 | 0.9539 | 0.9553 | 0.9844 | 0.9424 | 0.9958 |
|               |                       | ATV+NN           | 31.82 | 32.39 | 35.69 | 31.98 | 40.69 | 0.9540 | 0.9554 | 0.9909 | 0.9503 | 0.9956 |
|               | [−0.4, 0.4]           | ATV              | 31.43 | 31.91 | 34.58 | 31.19 | 39.10 | 0.9537 | 0.9541 | 0.9899 | 0.9378 | 0.9954 |
|               |                       | ITV              | 31.43 | 31.91 | 34.99 | 31.27 | 39.38 | 0.9537 | 0.9541 | 0.9840 | 0.9393 | 0.9956 |
|               |                       | ATV+NN           | 31.37 | 31.85 | 34.45 | 31.36 | 38.98 | 0.9536 | 0.9540 | 0.9900 | 0.9467 | 0.9953 |

![Figures 9, 10, and 11](images)

- 2) IR videos with time-invariant vertical stripe noise;
- 3) HSIs with vertical stripe noise and white Gaussian noise.

In the IR video experiments, we only consider stripe noise because Gaussian-like random noise does not appear in raw IR video data [14], [15]. For the variety of experiments, we considered the following five types of the intensity range of stripe noise: [−0.2, 0.2], [−0.25, 0.25], [−0.3, 0.3], [−0.35, 0.35], and [−0.4, 0.4]. The standard deviation of white Gaussian noise was set to 0.05.

Tables IV, V, and VI list the resulting MPSNR and MSSIM values in Case (i), Case (ii), and Case (iii), respectively. The best and second-best values are highlighted in bold and underlined, respectively. The proposed FC achieved the best/second-best MPSNR and MSSIM values in most cases. S and GS performed worse overall. LR and TV performed better than S and GS. However, the performance of LR and TV is significantly degraded in the cases where they are combined with a low-rank image regularization (LR-TNN) and TV image regularizations (TV-SSTV and TV-ASSTV), respectively.

Figs. 6, 7, and 8 depict the Moffett field destriping results in Case (i) using SSTV, the Bats1 destriping results in Case (ii) using ATV, and the Salinas destriping results in Case (iii) using TNN, respectively. Fig. 9 plots their bandwise or framewise peak signal to noise ratios (PSNRs) and structural similarities (SSIMs).
TABLE VI

| HSI          | Range of stripe noise | Regularization function | MPSNR | MSSIM |
|--------------|-----------------------|-------------------------|-------|-------|
|              |                       | S [29] | GS [37] | LR [40] | TV [44] | PC       | S [29] | GS [37] | LR [40] | TV [44] | PC |
| ITV          | [0.25, 0.25]         | -0.31 | -0.57 | -0.67 | -0.77 | -0.87 | 30.19 | 30.74 | 30.94 | 31.04 | 31.14 |
| SSTV         | [0.3, 0.3]           | 30.84 | 30.98 | 31.00 | 31.02 | 31.04 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| ASSTV        | [0.35, 0.35]         | 30.79 | 30.95 | 31.00 | 31.03 | 31.05 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| TNN          | [0.4, 0.4]           | 30.64 | 30.90 | 31.00 | 31.04 | 31.06 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| SSTV+TNN     | [0.3, 0.3]           | 30.84 | 30.98 | 31.00 | 31.02 | 31.04 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| l−1/4-ITV    | [0.25, 0.25]         | -0.31 | -0.57 | -0.67 | -0.77 | -0.87 | 30.19 | 30.74 | 30.94 | 31.04 | 31.14 |
| SSTV         | [0.3, 0.3]           | 30.84 | 30.98 | 31.00 | 31.02 | 31.04 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| ASSTV        | [0.35, 0.35]         | 30.79 | 30.95 | 31.00 | 31.03 | 31.05 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| TNN          | [0.4, 0.4]           | 30.64 | 30.90 | 31.00 | 31.04 | 31.06 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| SSTV+TNN     | [0.3, 0.3]           | 30.84 | 30.98 | 31.00 | 31.02 | 31.04 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| l−1/4-ITV    | [0.25, 0.25]         | -0.31 | -0.57 | -0.67 | -0.77 | -0.87 | 30.19 | 30.74 | 30.94 | 31.04 | 31.14 |
| SSTV         | [0.3, 0.3]           | 30.84 | 30.98 | 31.00 | 31.02 | 31.04 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| ASSTV        | [0.35, 0.35]         | 30.79 | 30.95 | 31.00 | 31.03 | 31.05 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| TNN          | [0.4, 0.4]           | 30.64 | 30.90 | 31.00 | 31.04 | 31.06 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| SSTV+TNN     | [0.3, 0.3]           | 30.84 | 30.98 | 31.00 | 31.02 | 31.04 | 0.90 | 0.90 | 0.90 | 0.90 | 0.90 |
| l−1/4-ITV    | [0.25, 0.25]         | -0.31 | -0.57 | -0.67 | -0.77 | -0.87 | 30.19 | 30.74 | 30.94 | 31.04 | 31.14 |

PSNRs and SSIMs of S-SSTV, GS-SSTV, and LR-SSTV dropped to 30 [dB] and 0.7, respectively. This is because S-SSTV, GS-SSTV, and LR-SSTV excessively smoothened the spectral signatures around the band. In the magnified areas of Fig. 6(c)–(e), we see that the land shapes of the red and green bands are removed as Gaussian and stripe noise. TV-SSTV also resulted in the low PSNRs and SSIMs of the band 95 and eliminated some edges in addition to the stripe noise [see Fig. 6(f)]. S-ATV, GS-ATV, LR-ATV, and TV-ATV removed bats as the stripe noise, resulting in poor performance [see Fig. 7(c)–(f)]. Fig. 9(c) and (d) shows that the PSNRs and SSIMs of S-ATV, GS-ATV, LR-ATV, and TV-ATV vary according to frame numbers. The reason is that the results are worse as the number of unrestored bats increases. In contrast, FC-SSTV recovers the land shapes and edges [see Fig. 6(f)], and FC-ATV accurately removed stripe noise, leading to high PSNRs and SSIMs. The SSIM results for Fig. 9(e) and (f) were better for LR than FC and TV, but the PSNRs were better for FC and TV than LR. In particular, from 30 to 150 bands, FC and TV achieved 10 [dB] better PSNRs and 0.01 worse SSIMs than LR. In the magnified area of the stripe noise by LR-TNN [see Fig. 8(e)], the yellow line appears along with
a field shape. This indicates that LR-TNN restores the image structure but does not recover the contrast. The three results verify that FC consistently achieves high performance due to its accurate capturing ability for stripe noise.

Fig. 10 shows the means of MPSNRs and MSSIMs in each noise case. In Case (i), LR and FC accurately captured stripe noise, leading to better performances than TV. In Case (ii), FC achieved the best performance. This is because FC captures the temporal flatness, while the other characterizations do not. In Case (iii), LR captured horizontal lines as a stripe noise component to remove Gaussian noise by the intersections in the overall luminance level so that it does not prevent spectral oversmoothing caused by the image regularizations. On the other hand, TV and FC obtained better results than LR without capturing the horizontal lines.

Fig. 11 plots the means of MPSNRs and MSSIMs in each stripe noise intensity. LR dropped its MPSNRs as the stripe noise intensities increased. This is due to the fact that LR removes the meaningful image components as stripe noise components if stripe noise intensity is high. The MPSNRs and MSSIMs of TV did not decrease depending on the stripe noise intensities but were lower than FC overall. Compared with these existing stripe noise characterizations, FC accurately eliminated stripe noise, resulting in high destriping performances regardless of the stripe noise intensity.

Fig. 12 shows the means of MPSNRs and MSSIMs in each image regularization. FC resulted in 0.5 [dB] worse MPSNRs than LR for the ASSTV and SSTV+TNN cases. This is because FC-ASSTV and FC-SSTV+TNN stop the iterations before the stripe noise components satisfy the FC, leading to slightly dropping their MPSNRs and MSSIMs. On the other hand, FC did obtain a 2 [dB] better MPSNR and 0.05 better MSSIM than LR for the TNN case. Compared with TV, the performances of FC were similar for HTV, TN, SSTV+TNN, and $l_0$-HTV and better for SSTV and ASSTV. Moreover, FC stably performed better than the other characterizations for ATV, ITV, and ATV+NN. These reveal that our framework achieves good performance on average, whatever image regularizations are used.

D. Experiments in Real Noise Cases

In the real noise-case experiments, the parameter $\lambda$ (Table III) for each method was determined manually to balance the tradeoff between the visual quality (e.g., over-smoothed or not) and destriping performance (e.g., stripe noise is sufficiently removed or not). For the data fidelity parameter $\varepsilon$, we adjusted it to an appropriate value after empirically estimating the intensity of the noise in the real data. Specifically, it was set to 200 for Suwannee and 0 for Bats2. The stopping criterion of Algorithm 1 was set as $\|U^{(n+1)} - U^{(n)}\|_F / \|U^{(n)}\|_F < 1 \times 10^{-4}$.

We show the Suwannee destripping results for a real noise case in Fig. 14. The destripping result by S-HTV [see Fig. 14(a1)] includes residual stripe noise. The results by S-SSTV [see Fig. 14(a2)], S-ASSTV [see Fig. 14(a3)], S-TNN [see Fig. 14(a4)], and S-ASSTV [see Fig. 14(a6)] have brighter areas than the original image [see Fig. 13(a)], and some of the land shapes in the magnified areas were removed as the stripe noise components. These suggest that S and GS are less capable of capturing the vertical continuity of stripe noise. LR-ASSTV [see Fig. 14(c3)] recovered the narrow river that lies along with the vertical direction in the magnified areas. On the other hand, LR-SSTV [see Fig. 14(c2)] and LR-TNN [see Fig. 14(c4)] removed part of the global structure in the image as stripe noise. This may be due to the fact that LR allows for changes in the overall luminance level so that it does not prevent spectral oversmoothing caused by the image regularizations. In the results by TV-SSTV [see Fig. 14(d2)], TV-ASSTV [see Fig. 14(d3)], TV-SSTV+TNN [see Fig. 14(d5)], and TV-$l_0$-HTV [see Fig. 14(d6)], land shape was also partially removed as the stripe noise. For example, TV-ASSTV [see Fig. 14(d3)] completely removed the narrow river in the magnified area. This is because there is a conflict between SSTV, ASSTV, SSTV+TNN, and $l_0$-HTV, used as image regularizations, and TV, used as a stripe noise characterization. Compared with these existing stripe noise characterizations, for FC-HTV, FC-SSTV, FC-TNN, and FC-$l_0$-HTV, its strong ability of stripe noise characterization allows us to achieve desirable destripping. However, our results do not satisfy the FC and slightly include land shapes in the stripe.
noise components only for FC-ASSTV and FC-SSTV+TNN [see Fig. 14(e3) and (e5)]. This indicates that FC-ASSTV and FC-SSTV+TNN need more iterations to preclude the land shapes from their stripe noise components.

Fig. 15 shows the destriping results of the IR video Bats2. S and TV removed bats (moving objects) as the stripe noise. This is because the stripe noise components [see Fig. 15(a1)–(a3) and (d1)–(d3)] has sparse or vertical smoothness properties. GS and LR performed better than S and TV, but some of bats were regarded as stripe noise components [see Fig. 15(b1)–(b3) and (c1)–(c3)]. In contrast to these stripe noise characterizations, our FC, when combined with any of the image regularizations, removed only the stripe noise while maintaining bats [see Fig. 15(e1)–(e3)].

**E. Comparison With a Deep Learning-Based Method**

We compare our framework with a deep learning-based method [20], where we adjust the parameter, so as to achieve the best MPSNR. As observed images, the Moffett Field and Salinas degraded by stripe noise with [−0.3, 0.3] and Gaussian noise with \( \sigma = 0.05 \) are used. Fig. 16 show the destriping results, which validates the effectiveness of our framework compared to a deep learning-based method. The method in [20] did not recover edges and objects [see Fig. 16(b) and (e)], leading to worse MPSNRs and MSSIMs. This is due to the limitation of deep learning-based methods in capturing textures and singular features, as also mentioned in [21], [22].

**F. Discussion**

From the above experiments, we summarize the advantages and limitations of our framework as follows.

1) FC accurately captures various intensities of stripe noise for any target images without image components.
2) In particular, FC eliminates high intensities of stripe noise.
3) Our framework consistently removes stripe noise, whatever image regularizations are combined.
4) When using some image regularization, such as ASSTV and SSTV+TNN, our framework requires many iterations to converge.

**V. Conclusion**

In this article, we have proposed a general destriping framework for remote sensing images. Specifically, we formulated the destriping as a convex optimization problem equipped with the FC. Due to the strong characterization of stripe noise, our framework is compatible with various regularization functions and achieves effective destriping. Then, we develop
a solver for the problem based on DP-PDS, which allows us to avoid step size adjustment. Through destriping experiments using HSI and IR video data, we found that our framework is advantageous on average compared to existing methods, whatever image regularizations are used. For future work, our framework needs the extension to consider the various degradation, such as the spectral variability and the effectiveness demonstration in remote sensing image applications, such as classification, unmixing, compressed sensing reconstruction, and target recognition.

Appendix A

Convergence of DP-PDS

Consider a convex optimization problem of the following form:

$$\min_{Z,Y} f_1(Z) + f_2(Y) \quad \text{s.t.} \quad Y = \mathcal{R}(Z),$$

(30)

where $Z = (Z_1, \ldots, Z_{N_G}) \in \prod_{i=1}^{N_G} \mathbb{R}^{N_i \times \cdots \times N_i}$ and $Y = (Y_1, \ldots, Y_{M_G}) \in \prod_{i=1}^{M_G} \mathbb{R}^{m_i \times \cdots \times m_i}$ are variables that include $N_0$ tensors and $M_0$ tensors, respectively, $f_1$ and $f_2$ are proper lower semicontinuous convex functions, and $\mathcal{R}$ is a linear operator.

We consider the following iterative procedures:

$$Z^{(n+1)} \leftarrow \text{prox}_{G_1^{-1}, f_1}(Z^{(n)} - G_1 \odot \mathcal{R}(\gamma^{(n)})), \quad \gamma^{(n+1)} \leftarrow \text{prox}_{G_2^{-1}, f_2}(\gamma^{(n)} + G_2 \odot (2Z^{(n+1)} - Z^{(n)})),$$

(31)

where $f_2^{-1}$ is the Fenchel–Rockafellar conjugate function of $f_2$, and $G_1 = (G_{1,1}, \ldots, G_{1,N_G}) \in \prod_{i=1}^{N_G} \mathbb{R}^{N_i \times \cdots \times N_i}$ and $G_2 = (G_{2,1}, \ldots, G_{2,M_G}) \in \prod_{i=1}^{M_G} \mathbb{R}^{m_i \times \cdots \times m_i}$ are preconditioners. For any $Z^{(0)} = (Z_1^{(0)}, \ldots, Z_{N_G}^{(0)})$ and $\gamma^{(0)} = (\gamma_1^{(0)}, \ldots, \gamma_{M_G}^{(0)})$ that converge to the optimal solution of Problem (30) if the linear operator $\mathcal{R}$ and preconditioners $G_1, G_2$ satisfy the following condition [59, Lemma 1]: for any $X \neq 0 \in \prod_{i=1}^{N_G} \mathbb{R}^{N_i \times \cdots \times N_i}$

$$\|G_2 \odot \mathcal{R}(G_1 \odot X)\|_F < \|X\|_F.$$

(32)

Note that matrix–vector multiplication between a diagonal matrix and a vector is equivalent to the tensor–tensor Hadamard product. Therefore, (31) is identical to the algorithm described in [59].

DP-PDS sets $G_1$ and $G_2$ as follows. Since $\mathcal{R}^*$ is a linear operator, the $(i_1, \ldots, i_N)$th entry of $Z_i$ is yielded by linear combinations of $Y$ as follows:

$$Z_i(i_1, \ldots, i_N) = \sum_{j} \sum_{j_1, j_2, \ldots, j_M} k_{j_1, j_2, \ldots, j_M}^j \gamma_j(j_1, \ldots, j_M).$$

(33)

Then, the $(i_1, \ldots, i_N)$th entry of $G_{1,i}$ is given as

$$G_{1,i}(i_1, \ldots, i_N) = \frac{1}{\sum_j \sum_{j_1, j_2, \ldots, j_M} |k_{j_1, j_2, \ldots, j_M}^j|}.$$  

(34)

Similarly, the $(i_1, \ldots, i_M)$th entry of $Y_i$ is given as

$$Y_i(i_1, \ldots, i_M) = \sum_j \sum_{j_1, j_2, \ldots, j_N} k_{j_1, j_2, \ldots, j_N}^j \gamma_j(j_1, \ldots, j_N).$$

(35)

Then, the $(i_1, \ldots, i_M)$th entry of $G_{2,i}$ is given as

$$G_{2,i}(i_1, \ldots, i_M) = \frac{1}{\sum_j \sum_{j_1, j_2, \ldots, j_N} |k_{j_1, j_2, \ldots, j_N}^j|}.$$ 

(36)
These preconditioners $G_1$ and $G_2$ satisfy the condition in [59, Lemma 2], i.e., (31) computes the solution of Problem (30).

**APPENDIX B**

**CONVERGENCE OF OUR ALGORITHM**

Let $Z = (U, S)$ and $Y = (Y_1, \ldots, Y_{1,K}, Y_2, \ldots, Y_{2,L})$. Then, by defining

$$ f_1(Z) := \|S\|_1, $$

$$ f_2(Y) := \sum_{k=1}^K R_k (Y_{1,k}) + i_1 (Y_2) + i_2 (\xi_1, \xi_2), $$

$$ r(Z) := (\xi_1 (U), \ldots, \xi_K (U), \xi_2 (S), \xi_3 (S), U + S). $$

(37)

Problem (6) is reduced to Problem (30), i.e., Problem (6) is a special case of Problem (30). Therefore, our algorithm satisfies the convergence property of the original DP-PDS.

**REFERENCES**

[1] J. M. Bioucas-Dias et al., “Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches,” *IEEE J. Sel. Topics Geoip. Remote Sens.*, vol. 5, no. 2, pp. 354–379, Apr. 2012.

[2] J. W. Beletic et al., “Teledyne imaging sensors: Infrared imaging technologies for astronomy and civil space,” *Proc. SPIE*, vol. 7021, Jul. 2008, Art. no. 70210H.

[3] Z. Wu, N. Fuller, D. Theriault, and M. Betke, “A thermal infrared video benchmark for visual analysis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2014, pp. 201–208.

[4] W. He, H. Zhang, H. Shen, and L. Zhang, “Hyperspectral image denoising using local low-rank matrix recovery and global spatial–spectral total variation,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 3, pp. 713–729, Mar. 2018.

[5] N. Liu, W. Li, R. Tao, and J. E. Fowler, “Wavelet-domain low-rank/group-sparse destriping for hyperspectral imagery,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 10310–10321, Dec. 2019.

[6] L. Liu, L. Xu, and H. Fang, “Simultaneous intensity bias estimation and stripe noise removal in infrared images using the global and local sparsity constraints,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 1777–1879, Mar. 2020.

[7] W.-K. Ma et al., “A signal processing perspective on hyperspectral unmixing: Insights from remote sensing,” *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 67–81, Jan. 2014.

[8] A. Romero, C. Gatta, and G. Camps-Valls, “Unsupervised deep feature extraction for remote sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1349–1362, Mar. 2016.

[9] E. Maggio, V. Turchi, G. C. Fanchi, and P. Alliez, “Convolutional neural networks for large-scale remote-sensing image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[10] L. Gao, B. Zhao, X. Jia, W. Liao, and B. Zhang, “Optimized kernel minimum noise fraction transformation for hyperspectral image classification,” *Remote Sens.*, vol. 9, no. 6, p. 548, Jun. 2017.

[11] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, “Graph convolutional networks for hyperspectral image classification,” *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 7, pp. 5966–5978, Jul. 2021.

[12] X. Li, V. Monga, and A. Mahalanobis, “Multiview automatic target recognition for infrared imagery using collaborative sparse priors,” *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 10, pp. 6776–6790, Oct. 2020.

[13] B. Münch, P. Tiritik, F. Marone, and M. Stapanoni, “Stripe and ring artifact removal with combined wavelet-Fourier filtering,” *Opt. Exp.*, vol. 17, no. 10, pp. 8567–8591, 2009.

[14] R. Sheng-Hui, Z. Hui-Xin, Q. Han-Lin, L. Rui, and Q. Kun, “Guided filter and adaptive learning rate based non-uniformity correction algorithm for infrared focal plane array,” *Inf. Phys. Technol.*, vol. 76, pp. 691–697, May 2016.

[15] Q. Zeng, H. Qin, X. Yan, and H. Zhou, “Fourier spectrum guidance for stripe noise removal in thermal infrared imagery,” *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 1072–1076, Jun. 2020.

[16] X. Kuang, X. Sui, Y. Liu, Q. Chen, and G. Gu, “Singe infrared image optical noise removal using a deep convolutional neural network,” *IEEE Photon. J.*, vol. 10, no. 2, pp. 1–15, Apr. 2018.
(39) Y. Chen, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and M. Wang, “Group sparsity based regularization model for remote sensing image stripe noise removal,” Neurocomputing, vol. 267, pp. 95–106, Dec. 2017.

(40) Y. Chang, L. Yan, T. Wu, and S. Zhong, “Remote sensing image stripe noise removal: From image decomposition perspective,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 12, pp. 7018–7031, Dec. 2016.

(41) Hu W, Li N, Liu R, Tao F, Zhang, and P. Scheunders, “Hyper-spectral image restoration using adaptive anisotropy total variation and nuclear norms,” IEEE Trans. Geosci. Remote Sens., vol. 59, no. 2, pp. 1516–1533, Feb. 2021.

(42) X. Liu, X. Lu, H. Shen, Q. Yuan, Y. Jiao, and 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., vol. 54, no. 5, pp. 3049–3060, May 2016.

(43) X. Liu, H. Shen, Q. Yuan, X. Lu, and C. Zhou, “A universal destriping framework combining 1-D and 2-D variational optimization methods,” IEEE Trans. Geosci. Remote Sens., vol. 56, no. 2, pp. 808–822, Feb. 2018.

(44) H.-X. Dou, T.-Z. Huang, L.-J. Deng, X.-L. Zhao, and J. Huang, “Directional ℓ0 sparse modeling for image stripe noise removal,” Remote Sens., vol. 10, no. 3, p. 361, Feb. 2018.

(45) R. Chartrand, “Nonconvex splitting for regularized low-rank + sparse decomposition,” IEEE Trans. Signal Process., vol. 60, no. 11, pp. 5810–5819, Nov. 2012.

(46) B. Bao, C. Gan, and R. R. Nadakuditi, “Panoramic robust PCA for foreground-background separation on noisy, free-motion camera video,” IEEE Trans. Comput. Imag., vol. 5, no. 2, pp. 195–211, Jun. 2019.

(47) A. J. Tom and S. N. George, “Video completion and simultaneous moving object detection for extreme surveillance environments,” IEEE Signal Process. Lett., vol. 26, no. 4, pp. 577–581, Apr. 2019.

(48) Q. Lu, Z. Lu, X. Tao, and H. Li, “A new non-local video denoising scheme using low-rank representation and total variation regularization,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Jun. 2014, pp. 2724–2727.

(49) O. Riou, S. Berberi, and P. Bremond, “Nonuniformity correction and thermal drift compensation of thermal infrared camera,” Proc. SPIE, vol. 905, pp. 281–302, Apr. 2004.

(50) Y. Can, M. Y. Yang, and C.-L. Tisse, “Effective strip noise removal for low-textured infrared images based on 1-D guided filtering,” IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 12, pp. 2176–2188, Dec. 2016.

(51) N. D. Sidiroopoulos, L. De Lathauwer, E. Papalexakis, and C. Faloutsos, “Tensor decomposition for signal processing and machine learning,” IEEE Trans. Signal Process., vol. 65, no. 13, pp. 3551–3582, Jan. 2017.

(52) Y. Ji, Q. Wang, X. Li, and J. Liu, “A survey on tensor techniques and applications in machine learning,” IEEE Access, vol. 7, pp. 162950–162990, 2019.

(53) K. Naganuma, S. Takeyama, and S. Ono, “Zero-gradient constraints for destriping of remote-sensing data,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Jun. 2021, pp. 1480–1484.

(54) M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems,” IEEE Trans. Image Process., vol. 20, no. 3, pp. 681–695, Mar. 2011.

(55) M. Carivan and L. Blanc-Feraud, “Sparse Poisson noisy image deblurring,” IEEE Trans. Image Process., vol. 21, no. 4, pp. 1834–1846, Apr. 2012.

(56) G. Chierchia, N. Pustelnik, J.-C. Pesquet, and B. Pesquet-Popescu, “Epigrapical projection and proximal tools for solving constrained convex optimization problems,” Signal, Image Video Process., vol. 9, no. 8, pp. 1737–1749, 2015.

(57) S. Ono and I. Yamada, “Signal recovery with certain involved convex data-fidelity constraints,” IEEE Trans. Signal Process., vol. 63, no. 22, pp. 6149–6163, Nov. 2015.

(58) S. Ono, “Efficient constrained signal reconstruction by randomized epigraphical projection,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 4903–4907.

(59) T. Pock and A. Chambolle, “Diagonal preconditioning for first order primal-dual algorithms in convex optimization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Barcelona, Spain, Nov. 2011, pp. 1762–1769.

(60) A. Chambolle and T. Pock, “A first-order primal-dual algorithm for convex problems with applications to imaging,” J. Math. Imag. Vis., vol. 40, no. 1, pp. 120–145, May 2011.

(61) L. Condat, “A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms,” J. Optim. Theory Appl., vol. 158, no. 2, pp. 460–479, Aug. 2013.

(62) S. Becker and J. Fadili, “A quasi-Newton proximal splitting method,” in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 2618–2626.

(63) AVIRIS. Accessed: Mar. 2, 2022. [Online]. Available: https://aviris.jpl.nasa.gov/data/free_data.html

(64) GIC. Accessed: Mar. 2, 2022. [Online]. Available: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

(65) SpectIR. Accessed: Mar. 2, 2022. [Online]. Available: http://www.specir.com/free-data-samples/

(66) BU-TIV Dataset. Accessed: Mar. 2, 2022. [Online]. Available: http://crs.bsu.edu/BU-TIV/BU-TIV.html

(67) Z. Wu, N. I. Hristov, T. H. Kunz, and M. Betke, “Tracking-reconstruction or reconstruction-tracking? Comparison of two multiple hypothesis tracking approaches to interpret 3D object motion from several camera views,” in Proc. Workshop Motion Video Comput. (WMVC), Dec. 2009, pp. 1–8.

(68) Z. Wu, A. Thangali, S. Sclaroff, and M. Betke, “Coupling detection and data association for multiple object tracking,” in Proc. CVPR, Jun. 2012, pp. 1948–1955.

(69) Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004.

Kazuki Naganuma (Student Member, IEEE) received the B.E. degree in information and computer sciences from the Kanagawa Institute of Technology, Atsugi, Japan, in 2020. He is pursuing the M.E. degree with the Department of Computer Science, Tokyo Institute of Technology, Yokohama, Japan.

His research interests are in signal and image processing, and optimization theory.

Shunsuke Ono (Member, IEEE) received the B.E. degree in computer science and the M.E. and Ph.D. degrees in communications and computer engineering from the Tokyo Institute of Technology, Yokohama, Japan, in 2010, 2012, and 2014, respectively.

From April 2012 to September 2014, he was a Research Fellow (DC1) of the Japan Society for the Promotion of Science (JSPS). He is an Associate Professor with the Department of Computer Science, School of Computing, Tokyo Institute of Technology. From October 2016 to March 2020, he was a Researcher of Precursory Research for Embryonic Science and Technology (PRESTO), Japan Science and Technology Corporation (JST), Tokyo. His research interests include signal processing, computational imaging, hyperspectral imaging and fusion, mathematical optimization, and data science.

Dr. Ono received the Young Researchers’ Award and the Excellent Paper Award from the Institute of Electronics, Information and Communication Engineers (IEICE) in 2013 and 2014, respectively, the Outstanding Student Journal Paper Award and the Young Author Best Paper Award from the IEEE Signal Processing Society (SPS) Japan Chapter in 2014 and 2020, respectively, and the Funai Research Award from the Funai Foundation in 2017. He has been an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS since 2019.